Hélia Guerra (Ed.)

Physics and Computation 2010

3rd International Workshop

Luxor / Aswan, Egypt, August 30 - September 6

Pre-Proceedings

Centre for Applied Mathematics and Information Technology
Department of Mathematics
University of Azores

Physics and Computation 2010

3rd International Workshop
Luxor / Aswan, Egypt, August 30 – September 6

Pre-Proceedings

Hélia Guerra (ed.)

Centre for Applied Mathematics and Information Technology
Department of Mathematics
University of Azores

Preface

Workshop on Physics and Computation
Egypt, August 30–September 6 2010

The third International Workshop on Physics and Computation (P&C 2010) was organised by University of Alexandria (Egypt), University of Auckland (New Zealand), University of Azores (Portugal), Technical University of Lisbon (Portugal), and Technische Universität Wien (Austria). The venue was held in a cruise downstream and upstream the Nile river (Egypt), from Luxor (through Aswan) to Luxor.

This meeting is the third of the (re-inaugurating) series of workshops on Physics and Computation. The first two meetings were satellite events in Unconventional Computation conferences, respectively, in 2008 at University of Vienna (Austria) and in 2009 at University of Azores (Portugal). These meetings were becoming an annual event to promote interdisciplinary research in the fields of Physics and Computation. The series is coordinated by the Steering Committee: Časlav Brukner (University of Vienna), Cristian Calude, (University of Auckland), Gregory Chaitin (IBM's Thomas J. Watson Research Center), José Félix Costa (Technical University of Lisbon), István Németi (Hungarian Academy of Sciences).

P&C 2010 was based on tutorials, invited speakers, a special session, contributed papers, and informal presentations. The main topics covered were: analogue computation, axiomatization of physics (completeness, decidability, reduction), Church-Turing thesis, computing beyond the Turing barrier, digital physics, philosophy of physics (and computation), quantum computation (digital, analogue) and applications to Biology, quantum logics, reaction-diffusion models of computation (brain dynamics, BZ computers), relativity (spacetimes, computation, time travel, speedup), theory of measurement (axiomatization, complexity).

The present volume is the pre-proceedings and contains the abstracts and papers of the two tutorials, six invited speakers, two special session talks, seventeen contributed papers, and four informal presentations. There will be post-proceedings publications, including special issues of the Journals Applied Mathematics and Computation and International Journal of Unconventional Computing.

The two tutorial speakers were Gergely Székely and Marco Lanzagorta. The invited speakers were Samson Abramsky, Arturo Carsetti, John Case,

Gilles Dowek, Sonja Smets, and Salvador Venegas-Andraca, who gave talks about Categorial Foundations of Computer Science and Physics, Philosophy of Science, Learning Theory, CT Thesis, Quantum Logic, and Quantum computation. The special session, devoted to *2010: The wakening of the computer; Which technological realizations make us feel closer to the HAL 9000 Computer?*, included two invited talks given by Selmer Bringsjord and David Stork.

The Program Committee was composed by: Andrew Adamatzky (University of West England), Selim Akl (Queen's University, Canada), Hajnal Andreka (Alfréd Rényi Institute of Mathematics, Budapest), Edwin Beggs (University of Swansea), Olivier Bournez (École Polytechnique), Dan Browne (University College London), Cristian Calude (University of Auckland, New Zealand), Arturo Carsetti (University of Rome "Tor Vergata"), Barry Cooper (University of Leeds), Bob Coecke (University of Oxford), José Félix Costa (Technical University of Lisbon), Gilles Dowek (École Polytechnique and INRIA), Walid Gomaa (University of Alexandria), Viv Kendon (University of Leeds), Carlos Loureno (University of Lisbon), Judit Madarász (Alfréd Rényi Institute of Mathematics, Budapest), Yasser Omar (Technical University of Lisbon), Sonja Smets (University of Groningen , Netherlands), Mike Stannett (University of Sheffield), Karl Svozil (Technische Universität Wien), John V. Tucker (University of Swansea), Jiri Wiedermann (Academy of Sciences of the Czech Republic), Karoline Wiesner (University of Bristol), and Martin Ziegler (University of Paderborn, Germany).

The organization of the event was due to : Cristian Calude (University of Auckland) José Félix Costa (Technical University of Lisbon), Walid Gomaa (University of Alexandria), Hélia Guerra (University of Azores), and Karl Svozil (Technische Universität Wien).

The workshop was partially supported by University of Azores, Centro de Matemática e Aplicações Fundamentais (University of Lisbon), Centre for Discrete Mathematics and Theoretical Computer Science (University of Aukland), Springer, and the Touring Club of Egypt.

August 2010

Hélia Guerra
CMATI
University of Azores

Table of Contents

Invited Talks.

Special Session Contributions.

Regular Papers.

Informal Presentations.

Relational Hidden Variables and Non-Locality

Samson Abramsky

Oxford University Computing Laboratory

Abstract. We use a simple relational framework to develop the key notions and results on *hidden variables* and *non-locality*. The extensive literature on these topics in the foundations of quantum mechanics is couched in terms of probabilistic models, and properties such as locality and no-signalling are formulated probabilistically. We show that to a remarkable extent, the main structure of the theory, through the major No-Go theorems and beyond, survives intact under the replacement of probability distributions by mere relations. In particular, probabilistic notions of independence are replaced by purely logical ones.

We also study the relationships between quantum systems, probabilistic models and relational models. Probabilistic models can be reduced to relational ones by the 'possibilistic collapse', in which non-zero probabilities are conflated to (possible) truth. We show that all the independence properties we study are preserved by the possibilistic collapse, in the sense that if the property in its probabilistic form is satisfied by the probabilistic model, then the relational version of the property will be satisfied by its possibilistic collapse. More surprisingly, we also show a *lifting property*: if a relational model satisfies one of the independence properties, then there is a probabilistic model whose possibilistic collapse gives rise to the relational model, and which satisfies the probabilistic version of the property. These probabilistic models are constructed in a canonical fashion by a form of maximal entropy or indifference principle.

The Emergence of Meaning at the Co-Evolutive Level – An Epistemological Approach

Arturo Carsetti

University of Rome "Tor Vergata"

From a general point of view, we can affirm, according to Kauffman's main thesis, that the transition between order and chaos appears as an attractor for the evolutionary dynamics of networks which exhibit adaptation. The study of neural networks shows that such nets are reasonable (even if limited) mathematical models of a large class of non-linear dynamical system. The attractors of these networks can "simulate" natural object of interest. From a biological point of view, we can interpret, for instance, these attractors as cell types. From a cognitive point of view, we can interpret these very attractors as the natural classification that a specific network makes of the external world.

These findings represent a conservative widening of some of the well established achievements in the field of non-equilibrium thermodynamics. In particular, it is important to remark, to this proposal, that this widening concerns, first of all, the nature and the dynamics of the differentiation processes, the link, in perspective, existing between these last processes and the successive formation of particular basins of attraction. Actually, from Prigogine and Nicolis to Kauffman, we can perceive a coherent line of research based on the individuation of the principles characterizing the chaotic dynamics and, from a more general point of view, the nature of the intermediate state (the "aperiodic crystal", as Schrodinger called that particular intermediate state represented by DNA). These principles make essential reference, according to a neodarwinian scheme, to the existence of a precise "dialectics" between mutation, selection and differentiation. They give a first characterization of this kind of dialectics utilizing, in a creative way, the tools offered by contemporary complexity theory.

But we may wonder : even if this scheme is plausibly partially correct, is it also "true" ? Is it possible to explain the whole complexity of the self-organizing (living) processes within a general markovian frame even if enlarged by taking into consideration the role of natural selection and of the process of differentiation ? Does a logical level of explanation exist within which the self-organization processes and the dialectics between surface information and depth information (as it progressively develops in dependence of the observation activities) can play a determinant role, beside the classical factors represented by chance and necessity ? Certainly the selection rewards the flexibility and the supply of variability ; why, however, does the evolution appear to reward the supply not of a purely stochastic variability, but of a varied and articulated complexity and, consequently, of a constrained complexity ? As Atlan correctly remarks (1), in a natural self-organizing system (a biological one) the goal has not been set from the outside. What is self-organizing is the function itself with its meaning. The origin of meaning in the organization of the system is an emergent property.

Moreover, the origin of meaning is strictly connected to precise operations of observation and self-observation.

If we take into consideration, for instance, the afore mentioned "aperiodic crystal" we know that DNA appears as the receptacle of an information" programmed" by natural selection. It becomes embodied, along the successive expression of the laws of the "inscription", in the cellular growth that is taking place according to the constraints imposed by the selection performed within an ambient meaning and by the *bricolage* operated with respect to the preexisting structures. It is along this peculiar channel that the flux of deep information may, therefore, express itself and articulate, in a creative way, its original incompressibility according to the correlated emergence of different stages of functional construction. In this sense the DNA must be seen neither as a program nor as a set of "data". It appears, on the contrary, to be a source and a "model". Both the interpretation function and the representation apparatus concerning that particular cellular machinery represented by the activity of proteins make essential reference to this kind of model. We are effectively in front of a complex cellular (and parallel) developmental "network" within which we can individuate, first of all, the presence of a specific process of "inscription" as well as of an interpretation function operating at the level of surface representation. This network is open to the flux of deep information and results constrained by the selective pressures acting within an ambient meaning. The role of the attractors takes place in the background of this intricate series of processes ; it cannot concern only a component of the cycle of the metamorphosis.

The genome expresses itself into a given phenotype in a complex way. Actually, the genetic code codes for its own translating machinery, it determines the birth of a cellular machinery responsible, in turn, for gene regulation and expression. This cellular machinery "represents", step by step, the genome into an organism realizing a specific embodiment process. In this sense, the genome and the cellular machinery really interact establishing an evolving and coupled network : as we shall see, one of the key results of this interaction is represented by the continuous engraving (through selection) at the level of the organisms of specific formats : among them we can distinguish, first of all, the formats relative to the architectures of sensorial perception.

The genome determining the expression of a cellular machinery, determines the birth both of an apparatus and of a surface program "embedded" in that apparatus. As a matter of fact, the apparatus doesn't appear to be an interpreter with a given program, it appears rather as a parallel computing system (working at the surface level) with a precise evolving internal dynamics, a system able, moreover, to represent and reflect itself (and express, still within itself, its own truth predicate). The program "embedded" in this apparatus concerns the general frame of the connections and constraints progressively arising, its exclusive capacity to express (and canalize by forms) a specific coordination activity within the boundaries of the becoming net. This capacity, on the other hand, can be "crystallized" on the basis of specific operations of self-representation and abstraction, so that it can be, finally, seen as the very "image" of the em-

bodied programs (forms in action) through which the apparatus progressively self-organizes expressing its autonomy. Through this image it is possible for the system to recognize the secret paths of the intentional information characterizing its intrinsic development, as programmed by natural selection. The final result is a source that assumes a reproductive capacity commensurate with a precise invariance and with the constitution of intrinsic forms which inhabit life ; it inscribes itself as form and as an hereditary principle in action, as a source of varied complexity but compared with a hereditary apparatus which self-organizes as such in view of possible regeneration. The source which generates on the basis of self-reflection opens out, then, towards a self-reproduction process which is targeted and part of a co-evolutionary path. The *telos* has to fix and be fixed in a "mask" to allow the source to burst out and become form in action, to express itself by means of living and moving forms : hence a source that reveals itself as both productive and intentional, which rejects simple dissipation and progressively constructs starting from itself "strange objects" (according to Monod's definition) (2).

It is precisely with reference to this apparatus and to this embedded "program" that the genome acts as a model. A model that must not be considered only from a logical and semantical point of view (in a denotational sense), but also from a biological and functional point of view. As a model, that is, considered as acting information + intentionality. If we aim, for instance, to describe the functional nature of this particular model as well as of the link existing at the biological level between form and information, the resolution, however, of at least of three orders of problems results indispensable : 1) the outlining of a statistical mechanics at the biological level concerning genes and macromolecules and no more only atoms and molecules, able, moreover, to take into consideration the role of the self-organization forces ; 2) the outlining of a semantic information theory taking into consideration the concept of observational meaning : the meaning as connected, at the same time, to a process, to an observer and to a hierarchical representation ; 3) the outlining of new measures with respect to the very concept of biological information. We need measures capable of taking into the consideration the growth processes, the statistical fluctuations living at the microscopic level etc. The Shannonian measure concerns essentially stationary processes articulating in a one-dimensional landscape.

The model is the "temporary" receptacle of the biological functions and of the replicative life ; in particular, it appears, as we have just said, as the receptacle of an information programmed by natural selection. The genome, in other words, is a model for a series of biological actions and symmetry breakings, for the realization of a complex path whose goal is represented by the attainment, on behalf of the apparatus, of a sufficiently complete functional autonomy at the surface level (within a dynamic ambient meaning). The interpretation function relative to this kind of model appears to concern, therefore, the actual realization of the embodiment process. In this sense, as Maynard Smith correctly remarks (3), a DNA molecule has a particular sequence because it specifies a particular protein : it contains information concerning proteins and specifies a form that

articulates as synthesis in action. DNA and (regulatory) proteins carry instructions for the development of the organism ; in particular genomic information is meaningful in that it generates an organism able to survive in the environment in which selection has acted. In turn, the organisms act as vehicles capable of permitting the source the successive realization of its own "renewal". The source "channels" itself through the *telos* finally articulating as a model : we are really faced with an intentional information at work.

The coder imparting intentionality allows the information to be articulated as semantic, to be immersed in the meaning (i. e., to sanction the birth of an apparatus able to see according to the truth). Thus, the source will manage to code because the *telos* was able to "follow" the meaning in an adequate way. The DNA can constitute itself as model only *via* the embodiment process, in this sense the model at work necessarily reveals itself as intentional (self-organizing, in perspective, as a possible biological basis of a specific cognitive activity). Hence a source that through the *via* manages to code and perceive according to the truth but with respect to the progressive articulation and the "adjunction" of specific observers that inhabit the *Natura naturata*. Then, it will be possible the rising of a new "conception" at the level of the effective closure of operant meaning. The source that posits itself as model renders itself to the life ; on the other hand, the progressive realization of the embodiment, of an apparatus able to feed meaning, corresponds to the coding in action. Only the *telos* capable of reflecting itself into the truth will be able to offer the source real intentionality : hence the arising circularity between form and information.

From an objective point of view, the inscription process possesses a self-limiting character with respect to the infinite potentialities of expression which are present in the source. Moreover the model, at the beginning, is "blind". In order to become a suitable channel for the successive revelation of the deep information living in the source, the model must not replicate simply itself : it has also to utilize the tools of the replication and the dissipation in order to realize a representation process possibly capable of allowing the source to express its inner creativity in a new and more complex way. It necessarily self-organizes within an ambient meaning on the basis of *telos*' activity.

From an informational point of view, life can be characterised in terms of a concrete answer to three difficult questions : "how is information generated ?", "how is information transmitted ?" and "how is information assimilated ?". With respect to this last interrogative, we have immediately to realise that the assimilation-process of external information implies the existence of specific forms of determination at the neural level as well as the continuous development of a specific cognitive synthesis. Actually, information relative to the system stimulus is not a simple amount of neutral sense-data to be ordered, it is linked to the "unfolding" of the selective action proper to the optical sieve, it articulates through the imposition of a whole web of constraints, possibly determining alternative channels at the level, for example, of internal trajectories. Depth information grafts itself on (and is triggered by) recurrent cycles of a self-organising activity characterised by the formation and the continuous

compositio of multi-level attractors. The possibility of the development of new systems of pattern recognition, of new modules of reading will depend on the extent to which new successful "garlands" of the functional patterns presented by the optical sieve are established at the neural level in an adequate way. The aforementioned self-organising activity thus constitutes the real support for the effective emergence of an autonomous cognitive system and its consciousness. Insofar as an "I" manages to close the "garlands" successfully, in accordance with the successive identification of specific attractors and the actual intervention of meaning selection, thereby harmonising with the ongoing "multiplication" of mental processes at the visual level, it can posit itself as an adequate grid-instrument for the "vision-reflection" on behalf of the original Source of itself, for its self-generating and "reflecting" as *Natura naturata*, a Nature which the very units (monads) of multiplication will actually be able to read and see through the eyes of the mind. Here we can recognize the ultimate roots of a true self-organising process articulating at the cognitive level.

If we take into consideration, for instance, visual cognition we can easily realise that vision is the end result of a construction realised in the conditions of experience. It is "direct" and organic in nature because the product of neither simple mental associations nor reversible reasoning, but, primarily, the "harmonic" and targeted articulation of specific attractors at different embedded levels. The resulting texture is experienced at the conscious level by means of self-reflection ; we actually sense that it cannot be reduced to anything else, but is primary and self-constituting. We see visual objects ; they have no independent existence in themselves but cannot be broken down into elementary data. Grasping the information at the visual level means managing to hear, as it were, inner speech. It means first of all capturing and "playing" each time, in an inner generative language, through progressive assimilation, selection and real metamorphosis (albeit partially and roughly) and according to "genealogical" modules, the articulation of the complex semantic grid which works at the deep level and moulds and subtends, in a mediate way, the presentation of the functional patterns at the level of the optical sieve.

Vision as emergence aims first of all to grasp (and "play") the paths and the modalities that determine the selective action, the modalities specifically relative to the revelation (and the construction) of this semantic "apparatus" at the surface level according to different and successive phases of generality. These paths and modalities thus manage to "speak" through my own fibres. It is exactly through a similar self-organizing process, characterised by the presence of a double-selection mechanism (i. e., by the correlated action of two different selective forces : the force linked to the full expression of the original incompressibility, on the one hand, and the force linked to the selective activity performed within an ambient meaning, on the other hand) that the mind can partially manage to perceive (and assimilate) depth information in an objective way. The extent to which the system-model succeeds, albeit partially, in encapsulating the secret cipher of this articulation through a specific chain of programs determines the model's ability to see (at the cognitive level) with the eyes of the

mind as well as the successive irruption of new patterns of creativity. To assimilate and see, the system must first "think" internally of the secret structures of the possible, and then posit itself as a channel (through the precise indication of forms of potential coagulum) for the process of opening and anchoring of depth information. This process then works itself gradually into the system's fibres, *via* possible selection, in accordance with the coagulum possibilities and the meaningful connections offered successively by the system itself.

The revelation and channelling procedures thus emerge as an essential and integrant part of a larger and coupled process of self-organization. In connection with this process we can ascertain the successive edification of an I-subject conceived as a progressively wrought work of abstraction, unification, and emergence. The fixed points which manage to articulate themselves within this channel, at the level of the trajectories of neural dynamics, represent the real bases on which the "I" can graft and progressively constitute itself. The I-subject can thus perceive to the extent in which the single visual perceptions are the end result of a coupled process which, through selection, finally leads the original Source to articulate and present itself as *true* invariance and as "harmony" within (and through) the architectures of reflection, imagination, computation and vision, at the level of the effective constitution of a body and "its" intelligence : the body of "my" mind. These perceptions are (partially) veridical, direct, and irreducible. They exist not in themselves, but, on the contrary, for the "I", but simultaneously constitute the primary departure-point for every successive form of reasoning perpetrated by the observer. As an observer I shall thus witness *Natura naturata* since I have connected functional forms at the semantic level according to a successful and coherent "score".

In accordance with these intuitions, we may tentatively consider, from the more general point of view of contemporary Self-organization theory, the network of meaningful (and "intelligent") causal "programs" living at the level of our body as a complex one which forms, articulates, and develops, functionally, within a "coupled universe" characterised by the presence of the afore mentioned double-selection mechanism. This network gradually posits itself as the real instrument for the actual emergence of meaning and the simultaneous, if indirect, surfacing of an "observing (and acting) I" : as the basic instrument, in other words, for the perception of real and meaningful processes, of strange "objects" possessing meaning, aims, intentions, etc. : above all, of objects possessing an inner plan and linked to the progressive expression of a specific cognitive action.

The mind considered as an intelligent "network" which develops with its meaning articulates as a growing neuronal system-model through which continuous restructuring processes are effected at a holistic level, thus constituting the indispensable basis of cognitive activity. The process is first of all, as stated above, one of canalization and revelation (*in primis* according to specific reflection procedures) of precise informational (and generative) fluxes-principles. It will necessarily articulate through schemata and attractors which will stabilise within circuits and flux determinations. In this sense the mind progressively constitutes itself as a self-organizing observing device in the world and of the

world. When, therefore, the system-model posits itself as an 'I-representation' (when the arch of canalization reaches "completion"), and observes the world-Nature before it, it "sees" (and computes) the world in consonance with the functional operations on which its realization was based, i.e. according to the architecture proper to the circuits and the patterns of meaning which managed to become established. The result is Nature written in mathematical formulae : Nature read and seen *iuxta propria principia* as a great book (library) of functional and operational forms by means of symbolic characters, grammatical patterns and specific mathematical modules.

From a general point of view, at the level of the articulation of visual cognition, we are actually faced with the existence of precise forms of co-evolution. With respect to this dynamic context, we can recognize, at the level of the aforementioned process of inventive canalisation, not only the presence of modules of self-reflection but also the progressive unfolding of specific fusion and integration functions. We also find that the *Sinn* that embodies in specific and articulated rational intuitions guides and shapes the paths of the exploration selectively. It appears to determine, in particular, by means of the definition of precise constraints, the choice of a number of privileged patterns of functional dependencies with respect to the entire relational growth. As a result, we are able to inspect a precise spreading of the development dimensions, a selective cancellation of relations and the rising of specific differentiation process. Thus, we are faced with a new theoretical landscape characterized by the successive unfolding (in a co-evolutive context) of specific mental processes submitted to the action of well-defined selective pressures and to a continuous emergence of depth information. In this sense, this emergence reveals itself as canalized by means of the action of precise constraints that represent the end product of the successive transformation of the original *gestalten.* Actually, the *gestalten* can "shape" the perceptual space according to a visual order only insofar as they manage to act (on the basis of the metamorphosis undergone at the teleonomical level) as constraints concerning the generative (and selective) processes at work.

The *gestalten* constitute first of all the natural forms through which meaning can be enclosed (i.e., realizing its thread-like extension) and can modulate its action along the ramparts of its surface "captivity". In this sense, they determine at a primary level the gradual shaping of the structures of the "I" which cannot help but think through forms if it is to self-organize as an ongoing process of vision : if it wishes to perceive veridically, and ultimately posit itself as the fixed point for the process of vision (including, Husserl would add, the vision of the categories themselves). Actually, the source attains its own invariance not because it reflects a given, fixed order (an order that, in the background of the dissipation process, could only present itself as the order or law of Chance), but because it succeeds in individuating, each time, the necessary tools for its representation at a surface level so that new levels of the deep incompressibility can, finally, express and inscribe themselves as new functional (and living) forms. These forms will represent the "intentional" stakes able to support the real embodiment of the capacity of creative replication of the source, the new

"moments" of a Time considered, contemporarily, both as creation and as recovery. Thus, life and cognition appear as indissolubly intertwined (4).

According to this frame of reference and from a mathematical point of view, true cognition appears as constrained by the continuous reference to a number of specific analytical tools : computability and the Turing universe, incompressibility and the oracles in action, self-organising nets, deterministic chaos, non-linear mathematics, second-order structures and so on. With respect to this particular framework, the simulation activity, the construction, for instance, of an adequate semantics for natural language, presents itself as a form of interactive knowledge of the complex chain of biological realizations through which Nature reveals itself to our brains in a consistent way (by means, for example, of the intelligent design of specific experiments at the level of an extended Turing universe). To simulate, in this sense, is not only a form of self-reflection or a kind of simple recovery performed by a complex cognitive net in order to represent itself at the surface level and "join" the government in action. The simulation work, in effect, offers the semantic net real instruments in order to perform a self-description process and to outline specific procedures of control as well as a possible map of an entire series of imagination (and invention) paths. The progressive (and selective) exploration of these paths will allow, then, external information to canalise in an emergent way, and to exploit new and even more complex patterns of interactive expression and action. It is exactly the framing of this particular kind of laboratory of possible emergence that will assure the successive revelation of ever new portions of deep information : that particular "irruption" of the Other (the renewed Source) which can express itself only within those particular fibres of the simulation and within that variant geometrical tissue of the "modules" which characterise, in an ultimate way, at the symbolic level, the cognitive activity of the subject. With respect to this epistemological setting, we are no longer only faced with an observation activity that manages to identify itself as vision according to the truth but also with a simulation activity and a metamorphosis of meaning which express themselves by means of use and interaction, by the continuous surfacing of new forms of productivity. When we pass from a world of objects to a world of constructions, we are no longer exclusively faced, for instance, with boolean algebras, first-order structures and observational acts, we are really in front of a dynamic and functional universe characterised by inner circularity, by self-organisation and by the presence of specific categorisation processes as well as of precise evolutive differentiation patterns. Moreover, at the level of this particular world, as we have just said, the role played by meaning is different ; meaning is now characterised in terms of a symbolic dynamics in action and with reference to a precise simulation language. As a consequence of this particular articulation, specific limitation facts can arise at the level of the progressive unfolding of this very language. New theoretical perspectives will reveal themselves with respect, in particular, to the inner self-organizing aspects of the emerging structure and to the specific constitution of the individuals inhabiting this very structure considered as individuals essentially characterized

not directly in terms of their properties but primarily in terms of their relations (and their secret "affordances" at the symbolic level).

In a self-organizing semantic net the successive bifurcations, the recurrent delimitations, actually appear, as temporal and connected determinations of meaning embodied streams. In this sense, such determinations (differently from Hintikka's appraisal of Kant's primitive intuitions), appear to concern not the (direct) successive presentation-construction of individuals, but the sudden revelation of patterns of constraints, the actual intervention of new clusters of selective choices at the level of the informational fluxes. Hence the essential link, in perspective, both with the contemporary definitions of complexity at the second-order level, and with the revisitation of some Leibniz's original intuitions as recently suggested presented, for example, by G. Chaitin and B. Cooper (5). In this sense, the aforesaid determinations of time articulate modulating themselves, in a recurrent way, as a tool for the further construction-unfolding of the inner creativity proper to the Source, as a sort of arch and gridiron for the construction (and the recovery) of the "Other" through the constraints of an intended "sacrifice".

In the light of these considerations, if we return now to the analysis of the observational procedures (abandoning, for the moment, the investigation of the simulative ones), the deep meaning appears first of all as relative to the action performed by precise semantic *fixed-points*, to a manifold, in particular, of subtended circumscription functions and to the progressive expression of specific postulates. The fixed-points of the resulting dynamics represent the "true" revelation of that specific tuning that characterizes and identifies the predicates and the properties at work. Thus, at the monadic and polyadic level, we are obliged to outline a new and specific kind of model : a self-organizing (and coupled) structure not bound to sets and individuals, (with relative attributes) but to generators and fluxes of tuned information. In this new theoretical framework, the simple reference to possible worlds (as in Frege or Hintikka, for instance) in order to take into account the structure of intensionality is no longer sufficient, One has also to resort, in the first instance, to the dynamics of the constraints, to the identification of the indices and of the recurrent paths of the informational flow as well as of the role played by the observer, i. e. to the interplay existing between intervening and change.

In order to refer these general ideas to the traditional realm of Information Theory, let us simply remember that starting from the theory of constituents, as introduced by Carnap, every consistent statement h of a specific and suitable language can be represented in the form of a disjunction of some (maybe all) of the constituents : $h = \mathrm{C_{i1}} \vee \mathrm{C_{i2}} \vee \ldots \vee \mathrm{C_{i\,w(h)}}$ where $\mathrm{C_{i1}}$, $\mathrm{C_{i2}}$,. . ., $\mathrm{C_{i\,w(h)}}$ is a subset of the set of all the constituents. The set {i1, i2, .., i$w(h)$ } is called the index set of *h,* and denoted by I*(h)*. The number *w(h)* is called the *width* of *h*. Then, we can introduce the probabilities and from the probabilities we can obtain measures of semantic information in the two ways given by (i) and (ii) as outlined by Carnap (and Popper) : (i) inf(h) = $\neg$ log p(h) , (ii) cont(h) = 1 - p(h). However, when we abandon the monadic level things are different ; in

particular, when we enter the polyadic realm and come to use, for instance, primitive binary relations, we are immediately faced with a series of choices (and assumptions) which are relative to the structural properties of such relations. As a consequence of the structural properties that characterize the dyadic predicates (i.e. that such predicates possess in an exclusively conceptual way), some specific conjunctions of these very predicates will be shown to be inconsistent. In this particular case, an individual as well as being considered as a collection of properties must also be defined as a chain or collection of relationships. This means that what must be joined together will no longer consist of simple entities or sets of properties but of configurations and graphs. Thus, the conjunction, at the level of generators, should be realized respecting precise constraints which are of a "geometric" nature, connected, in particular, to the successive gain of configurations of "points-patches" which possess determined characteristics. The role of compatibility factors becomes particularly essential. From here both the birth of complex cancellation procedures and the introduction by construction of new individuals, in a potentially unlimited way, arise. Likewise, we would have, in a correlated way, the introduction of nested quantifiers. Thus, the role played by meaning really assumes a specific and deep relevance. As a matter of fact, at the level of this type of structure, we can individuate the existence of an essential plot between the successive "presentation" of the constraints and the action of the meaning postulates, on the one hand, and the articulated design of mutations, cancellations and contractions of the predicates-inputs that characterize the higher layers of formal constructions, on the other. Hence the birth of new (and specific) measures of semantic information : in actual fact, at this level meaning can be expressed only by means of a specific intentional and symbolic dynamics. As we have just said, the source that posits itself as model renders itself to the life but necessarily in accordance with the truth. Only the *telos* capable of reflecting itself into the truth will be able to offer the source real intentionality. In this way precise forms of classification and therefore precise contexts of sense will appear ; specific intensional structures will begin to emerge : in particular, intensional grammars defined with reference to orders-spaces of higher level. From here comes the necessity of outlining, in the case of dyadic structures (and, in general, in the case of second-order structures), the sophisticated dynamism of a great book of Language that presents itself at the level of the conscious representation, like an effective reality in action. A reality which emerges, however, also through our thinking and which, at the same time, determines, first of all at the genetic level, this same thinking.

As we have just said, the mechanism which "extracts" pure intuitions from the underlying formal co-ordination activity, if parallel to the development of the *telos* as coder, is necessarily linked to the emergence of new mathematical moves at the level of the neural system's cognitive elaboration, This consideration inviting the revisiting of a number of Kantian hypotheses. It would appear, for instance, to be necessary not only to reread Kant in an evolutionistic key (cf., e.g. K. Lorenz), but also with reference to other speculative themes like, for instance, the indissoluble link existing between life and cognition and between

chance and necessity. Taking into consideration coder's action opens up a new and different relationship with the processes of mathematical invention, making it necessary, for example, to explore the second-order territories, the very realm of non-standard mathematics as well as the dialectics between observer and observed reality.

Pace Kant, at the level of a biological cognitive system sensibility is not a simple interface between absolute

Chance and an invariant intellectual order. On the contrary, the reference procedures, if successful, are able to modulate canalization and create the basis for the appearance of ever-new frames of incompressibility through morphogenesis. This is not a question of discovering and directly exploring (according, for instance, to Putnam's conception) new "territories", but of offering ourselves as the matrix and arch through which they can spring autonomously in accordance with ever increasing levels of complexity. There is no casual autonomous process already in existence, and no possible selection and synthesis activity *via* a possible "remnant" through reference procedures considered as a form of simple regimentation. These procedures are in actual fact functional to the construction and irruption of new incompressibility : meaning, as *Forma formans*, offers the possibility of creating a holistic anchorage, and is exactly what allows the categorial apparatus to emerge and act according to a coherent "arborization". The new invention, which is born then shapes and opens the (new) eyes of the mind : I see as a mind because new meaning is able to articulate

and take root through me.

As J. Petitot correctly remarks, according to Kant the pure intuitions are : ≪ 'abstraites de l'action même par laquelle l'esprit coordonne, selon des lois permanentes, ses sensations *(Dissertation, 177)* '. Or, cette coordination est elle-même innée et fonctionne comme un fondement de l'acquisition ≫(6). In this sense, the space appears as a format, the very basis of spatial intuition is innate ; however at the biological level, as we have just said, what is innate is the result of an evolutive process and is "programmed" by natural selection. Natural selection is the coder (once linked to the emergence of meaning) : at the same time at the biological level this emergence process is indissolubly correlated to the continuous construction of new formats in accordance with the unfolding of ever new mathematics, a mathematics that necessarily moulds coder's activity. Hence the necessity of articulating and inventing a mathematics capable of engraving itself in an evolutive landscape. In this sense, for instance, the realms of non standard models and non standard analysis represent, today, a fruitful perspective in order to point out, in mathematical terms, some of the basic concepts concerning the articulation of an adequate intentional information theory. This individuation, on the other side, presents itself not only as an important theoretical achievement but also as one of the essential bases of our very evolution as intelligent organisms.

Notes

1) Cf. Atlan, H. (2000), "Self-organizing networks : weak, strong and intentional, the role of their underdetermination" in A. Carsetti (ed.), *Functional Models of Cognition*, Dordrecht, Kluwer A. P.,127-143.
2) Cf. Carsetti, A. (2009), "Embodiment processes and intentional complexity", *La Nuova Critica*, 53-54 :115-136
3) Cf. Maynard Smith, J. (2000), "The concept of information in Biology", *Philosophy of Science*, 67 : 177-194.
4) Cf. at this proposal : Carsetti, A. (1992), "Meaning and complexity : a non-standard approach", *La Nuova Critica*, 19-20 :109-126
5) Cf. at this proposal : Chaitin, G. (2009), "Leibniz, Complexity and Incompleteness**"** and Cooper, B.S. (2009) "Incomputability, Emergence and the Turing Universe" in A. Carsetti (ed.), *Causality, Meaningful Complexity and Embodied Cognition*, Berlin, Springer, 127-135 and 136-155.
6) Cf. Petitot, J.(2008), *Neurogéometrie de la vision,* Paris, Les Editions de l'Ecole Polytechnique,397.

References

1. Amit, D.J., Gutfreund, H. & Sompolinsky, H. (1985), "Spin-glass models of neural networks", *Phys. Review Letters*, 55, 1530-33.
2. Anderson, P.W. (1985), "Suggested model for prebiotic evolution: the use of Chaos", *Proc. Nat. Acad. Sci., USA*, 3386.
3. Atlan, H. (1992), "Self-organizing networks: weak, strong and intentional, the role of their underdetermination", *PersonNameProductIDLa Nuova CriticaLa Nuova Critica*, 19-20, 51-71.
4. Ayala, F.J., (1999), " Adaptation and novelty: teleological explanations in evolutionary biology", *Hist. Phil. Life Sci.*, 21, 3-33.
5. Boolos, G. (1975), "On second-order logic", *J. of Phil.,* 72, 509-527.
6. Brooks, R.D., & Wiley, F.O. (1986), *Evolution as Entropy*, Chicago,.
7. Carnap, R. & Bar Hillel, Y. (1950), "An Outline of a Theory of Semantic Information", *Tech. Rep.* N. 247, M.I.T..
8. Carsetti, A. (1993**), "**Meaning and complexity: the role of non-standard models", *PersonNameProductIDLa Nuova CriticaLa Nuova Critica*, 22,**:** 57-86.
9. Carsetti, A. (2000), "Randomness, Information and Meaningful Complexity: Some Remarks About the Emergence of Biological Structures", *PersonNameProductIDLa Nuova CriticaLa Nuova Critica*, 36, 47-109.
10. Carsetti, A., (ed.) (2009), *Causality, Meaningful Complexity and Embodied Cognition*, Berlin, Springer.
11. haitin, G., (1987), *Algorithmic Information Theory*, Cambridge.
12. Chaitin G. & Calude C. (1999) " Mathematics/Randomness Everywhere", *Nature*, 400, 3219-20.

13. Denbigh, K.G. & Denbigh, J.S. (1985), *Entropy in relation to incomplete knowledge*, Cambridge,.
14. Gaifman, H. (2000), "What Goedel's Incompleteness result does and does not show", *The Jour. of Philosophy*, 9708, 462-70.
15. Gibbs, J.W. (1902), *Elementary Principles in Statistical Mechanics*, New Haven.Gillies, D.A. (1973), *An Objective Theory of Probability*, London.
16. Henkin, L. (1950), " Completeness in the theory of types", *Jour. of Symb. Logic*, 15, 81-91.
17. Hintikka, J. (1970), " Surface information and depth information", in Hintikka, J. and Suppes, P. (eds.), *Information and Inference*, Dordrecht, 298-330.
18. Hopfield, J.J. (1982), " Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. of the Nat. Ac. Scien.*, 79, 2254-2258.
19. Jaynes, E.T. (1957), "Information Theory and Statistical Mechanics", (I and II), *Phys. Rev*, 106 4, 620-630 and 108, 2, 171-190.
20. Jaynes, E.T. (1965), Gibbs vs Boltzmann Entropies, *Am. J. Phys.*, 33, 391.Kauffmann, S.A. (1993), *The Origins of Order*, New York.Kohonen, R. (1984), *Self-organization and Associative Memories*, Berlin,.Kolmogorov, N. (1968), Logical Basis for Information Theory and Probability Theory, *IEEE Trans. IT* 14, 5, 662-4.
21. Landsberg, P.T. (1978), *Thermodynamics and Statistical Mechanics*, London,.
22. Maynard Smith, J. (2000), "The concept of information in Biology", *Philosophy of Science*, 67, 177-194.
23. ayr, E. (2001), *What Evolution Is*, New York.Nicolis, G. (1989) "Physics in far-from-equilibrium systems and self-organization", in P. Davies (ed.), *The New Physics*, London,. Nicolis, G. & Prigogine, I. (1989) *Exploring Complexity*, New York.
24. Petitot, J.(2008), *Neurogéometrie de la vision,* Paris, Les Editions de l'Ecole Polytechnique
25. rigogine, I. (1980) *From Being to Becoming*, San Francisco,.
26. Putnam, H. (1965), " Trial and Error Predicate and the Solution to a Problem of Mostowski", *Jour. of Sym. Logic.*, 30, 49-57.
27. Sherrington, D. & Kirkpatrick, S. (1975), " Spin-glasses", *Phys. Review Letters*, 35, 197.
28. Wicken, J.S. (1987), *Evolution, Thermodynamics, and Information*, New York.Wuketits, F.M. (1992), "Self-organization, complexity and the emergence of human consciousness", *La Nuova Critica,* 19-20, 89-109.

Algorithmic Scientific Inference

Within Our Computable Expected Reality

John Case

Computer and Information Sciences Department
University of Delaware
Newark, DE 19716 USA
case@cis.udel.edu

Abstract. It is argued that, scientific laws, including quantum mechanical ones, can be considered algorithmic, that the *expected* behavior of the world, if not its exact behavior, is algorithmic, that, then, communities of human scientists over time have algorithmic expected behavior.
Some sample theorems about the boundaries of algorithmic scientific inference are then presented and interpreted. There is some discussion about (but there are not presentations of) succinct machine self-reference proofs of these theorems and whether non-artifactual self-referential examples may exist in the world.

Keywords: machine inductive inference, physics, philosophy of science

1 Scientific Laws

Below we describe in Section 1.1 how and why we model scientific laws in terms of algorithms, and, in Section 1.2, we provide important clarification with an example from *quantum mechanics*.

1.1 Modeling Scientific Laws

In the 1970s, I was motivated to work on the *Theory of Machine Inductive Inference*, Putnam and Gold [41, 27, 42], thanks to the Blums' assertion [2, Page 125] just below.

> Consider the physicist who looks for a law to explain a growing body of physical data. His data consist of a set of pairs (x, y), where x describes a particular experiment, e.g., a high-energy physics experiment, and y describes the results obtained, e.g., the particles produced and their respective properties. The law he seeks is essentially an algorithm for computing the function $f(x) = y$.

Such an algorithm is a *predictive explanation*, Case & Smith [15, 16]: if one has the good fortune to *have* such an algorithm, one can use it to predict the outcomes of the associated experiments.

Importantly, a predictive explanation must provide its predictions *algorithmically*! How else are we to get out the predictions — by magic? To be sure, in, say, physics, the laws are typically not written down including how to extract algorithmically the predictions. That is implicit and may, in some cases, be difficult. The techniques are essentially covered by computably axiomatizable mathematics, algorithmic numerical techniques, etc. Of course physicists rarely resort to axiom systems directly, but, when mathematics is formulated axiomatically, one always sees a computably decidable set of axioms. How else could formal proofs be checked, e.g., when they cite an axiom, — by magic? Of course with a formal system having a computably decidable set of axioms, the set of corresponding theorems forms a computably enumerable set.

1.2 A Quantum Mechanical Example

Here is the promised example chosen on purpose to be from quantum mechanics. Essentially from Case, et al, [9, 8]:

> x codes a particle diffraction experiment & $f(x)$ the resultant probable distribution (or interference pattern) on the other side of the diffraction grating. Quantum theory provides *deterministic, algorithmic* extraction of $f(x)$ from x. A program for f is, then, a *predictive* explanation or law for the set of such particle diffraction experiments.

The program/law in this case does *not* tell us deterministically where the particles go. It tells us instead, *deterministically, algorithmically*, their statistically *expected behavior*! In the case an interference pattern is generated from an experiment x where multiple particles are sent through a diffraction grating, it deterministically, algorithmically provides $f(x)$ which can be, then, a depiction of that interference pattern!

Again, for the reasons spelled out at the end of Section 1.1 just above, a *predictive explanation* must provide its predictions *algorithmically*!

2 Data Types

In this section we indicate in detail how, without loss of generality, we can and will treat the functions f such as those described above in Sections 1.1 and 1.2 just above.

A *countable set* is (by definition) one in 1-1 correspondence with (some $\subseteq$) $\mathbb{N} = \{0, 1, 2, \ldots\}$, the set of natural numbers.

My former student, Mark Fulk, [22] argued that the set of distinguishable experiments *one can actually do and record* on a phenomenon is countable: lab manuals can and do contain only *finite* notations, strings, and images from a *finite* alphabet of symbols, including gray and color pixel values.

One does *not* record *measurements* such as *arbitrary infinite-precision* real numbers of volts.

Beautiful continuous-mathematics (featuring *un*countable sets such as that of the real numbers) is employed in physics many times to smooth out *feasibly* some much too complicated discrete reality, e.g., a giant cloud of electrons.

Interestingly, Maddy [34] discusses the just prior paragraph, and provides a pointer, [18, Pages 290, 326], to cases where a continuous approximation to a discrete thermodynamic reality fails.

So, one of my working hypotheses is that reality is *discrete*. This is discussed further early in Section 3.1 below. Of course continuous mathematics is, in many cases, on a practical level, hard to replace.

In what follows, then, thanks to Gödel or code numbering an algorithmically circumscribed *countably* infinite set of experiments and outcomes for some phenomenon F, e.g., some well circumscribed particle diffraction phenomenon: we imagine coding (algorithmically) the set of experiments associated with F onto $\mathbb{N}$ and the possible outcomes into $\mathbb{N}$, and we let the function f (associated with phenomenon F) map any experiment on phenomenon F with code $\# x$, into the code $\# y$ of the outcome of x on F: $f(x) = y$.

Hence, the *type* of our fs can and will be taken to be $\mathbb{N} \to \mathbb{N}$.

Also, since we seek *algorithmic* explanations for F, we can handle the cases only where f is also *computable*.

N.B. Our above discussion does not yet take into account error bounds on measurements, an important, crucial, practical consideration. For our approach, we can just consider that the code numbers of experiments and outcomes, *include* measurement error bounds.

3 Computability

In Section 3.1 just below is discussed my additional working hypothesis that the *expected* behavior of reality is *algorithmic*.

Then in Section 3.2 further below we explain what this has to do with human scientific endeavors.

Next, in Section 3.3, we consider objections based on apparent human creativity and free will.

3.1 Computability of Expected Reality

Researchers in the cellular automata approach to physics, e.g., [19, 36, 21, 51, 48, 35, 47, 33, 50, 49, 52, 55, 46, 20, 28], take seriously the idea that the universe, including space and time, may well be discrete.[1]

In a discrete, random universe but with *computable probability distributions* for its expected behaviors (e.g., a discrete, quantum mechanical universe with such distributions — as, I believe, ours is), the *expected* behavior will still be

[1] Here Feynman [19] is crucial, and Minsky [36] lays out the ideas of Ed Fredkin on some of the different ways physical space could be discrete.

computable. It essentially follows from [17, 24, 25] that one can compile any algorithm r having access to a random oracle, *which oracle is subject to a computable distribution*, into a deterministic algorithm d_r computing, in a sense, the expected outputs of r.

Another working hypothesis of mine is, then, that the universe, besides being discrete, is *algorithmic* as to its *expected* behaviors.

N.B. We humans may be too finite ever to figure out *completely* how to *compute* the associated expected behaviors. But that's just about human limitations.

3.2 Computable Expected Behavior of Science

We *humans* are *components* of the universe; hence, communities of scientists over time *must also have computable expected behavior*!

Herein, then, we'll *model scientists* (and communities thereof over time) as *algorithmic*. *Then* we can have theorems about the *boundaries* of the (expected) behavior of science!

Just as a conservation assumption from physics provides boundaries on and insight into the physically possible, so too the computable expected behavior assumption on scientific inference provides boundaries on and insight into what's possible with scientific inference.

In [8] I discuss related language learning examples for *cognitive science* (not treated herein).

I invite physicists to explore the consequences *for physics* of our universe having computable expected behaviors. I'd really like to see something come out of that.

3.3 Creativity and Free Will

First we discuss creativity.

In a world with only computable expected behaviors, what about human *creativity*? How does my *somewhat* mechanistic working hypothesis account for the [5] unbidden images which occur to people and which lead to solutions of difficult problems and/or works of great beauty and significance for the human condition?

I argue [5] that humans are mostly not *consciously* aware of the brain processes that invoke such insights; hence, we have the *illusion* they aren't algorithmically produced. Our conscious thoughts are the mere tip of an iceberg.

Post [40] described as *creative* cases where algorithmic processes are *algorithmically* transcended. His examples generalize a bit the algorithmic process of Gödel [26] essentially for transforming an algorithm for deciding a set of "consistent" axioms for an arithmetic into a corresponding Gödel sentence. Adding (trivially algorithmically) that sentence to the axiom set provides the transcendence.[2]

Next we discuss free will.

[2] [8] briefly refutes the argument that Gödel's process falsifies mechanism.

Libet, et al, [32] found that particular, experimentally detectable *un*conscious cerebral activity always strictly precedes *conscious* human experiences of *will*ing to do something. This is somewhat suspicious methinks re the existence of human *free* will.

Conway and Kochen interpret their (Strong) Free Will Theorem [12, 13] to mean, if some human has free will (about setting the details of some quantum mechanics experiment), then so do some particles.

They want to retain human free will, so they ascribe it to some particles too. Of course, at least in the case of particles, they mean by it only non-determinism.

I'm inclined to see *conscious* free will as another one of many human illusions. We *may* have some non-determinism, but our *expected* behavior does not.

4 Machine Inductive Inference

Next we begin to describe a model of scientific inference.

$$(0, f(0)), \ldots, (t-1, f(t-1)) \quad \stackrel{\text{In}}{\longrightarrow} \quad \mathbf{M} \quad \stackrel{\text{Out}}{\longrightarrow} \quad p_t$$

Above M is an algorithmic device receiving f's data points $(t, f(t))$, at "times," $t = 0, 1, \ldots$. N.B. For simplicity herein we'll restrict the order of presentation of data from f to be in *this* order (this matters in *some* cases).

M's output above, having seen the data sequence

$$f[t] \stackrel{\text{def}}{=} (0, f(0)), ..., (t-1, f(t-1)),$$

is p_t, *where* p_t is a program in some fixed, general programming system.[3] We write $M(f[t]) = p_t$. N.B. For simplicity herein we'll restrict outselves to the case where M on $f[t]$ does *not* go into an infinite loop never producing p_t (this matters in *some* cases).

Perhaps, if M is "clever" enough and f is associated with a phenomenon F that is not too hard to figure out, eventually, i.e., for suitably large ts, the p_ts may come usefully close to computing f. More on this topic, in Section 4.1 just below where we begin to discuss in more detail what can be meant by *successful* scientific inference.

Then, in Section 4.2, we provide *with interpretations* some sample theorems about scientific inference.[4] Near the end of Section 4.2, we segue into Section 4.3 which discusses machine self-reference techniques which can, many times, be used to provide *very succinct* proofs, relevantly herein, of results regarding scientific inference.

Lastly, in Section 4.4, is discussed, whether the self-referential examples employed might actually correspond to (non-artifactual) examples in the real world.

[3] When $t = 0$, $f[t]$ is the empty sequence.

[4] [8] provides additional examples.

4.1 Criteria of Success

Definition 1 (Success Criteria $\mathbf{Ex}^a$) *Suppose $a \in (\mathbb{N} \cup \{*\})$. Suppose $\mathcal{S}$ is a class of computable functions f. '***Ex***' stands for 'Explanatory.' a stands for* a*nomaly.*

$\mathcal{S} \in \mathbf{Ex}^a$ *iff there is a suitably clever M so that, for* every $f \in \mathcal{S}$*, for* some *associated t, $M(f[t]) = M(f[t+1]) = \cdots$* and *$M(f[t])$* computes *$f$* — except *at up to a data points. Here, up to $*$ points means up to finitely many.*

Informally, M witnesses that $\mathcal{S} \in \mathbf{Ex}^a$ *means*, on any $f \in \mathcal{S}$, M's output programs on f, *eventually* settle down syntactically to a single program "for" f which program has *at most a* anomalous predictions re values of f.

In science, we don't know when (if ever) we begin to have predictive explanations that are pretty good; we don't know t's value in the above Definition.

Definition 2 (Success Criteria $\mathbf{Bc}^a$) *'***Bc***' stands for 'Behaviorally correct.'*

$\mathcal{S} \in \mathbf{Bc}^a$ *iff, for some M, for* every $f \in \mathcal{S}$*, for* some *associated t, programs $M(f[t]), M(f[t+1]), \cdots$* each *computes f — each except at up to a data points.*

For these $\mathbf{Bc}^a$ criteria, the programs $M(f[t]), M(f[t+1]), \cdots$ can *be (syntactically) quite different from one another.*

For the criteria $\mathbf{Ex}^a$ and $\mathbf{Bc}^a$, my original motivation for the importance of small values of a, i.e., a few anomalies being tolerated in final predictive explanations, came from *anomalous dispersion*: the *classical* explanation for the degree of bending of "light" passing through a prism, fails for the X-ray case, an anomalous case.

4.2 Sample Theorems

Theorem 3 (Gold & Blums [27, 2]) *The class of polynomial time computable functions $\in \mathbf{Ex}^0$.*

Theorem 4 (See [15, 16]) $\mathbf{Ex}^0 \subset \mathbf{Ex}^1 \subset \cdots \subset \mathbf{Ex}^* \subset \mathbf{Bc}^0 \subset \mathbf{Bc}^1 \subset \cdots \subset \mathbf{Bc}^*$*, where $\subset$ is* proper *subset.*

Hence, tolerating anomalies strictly *increases inferring power as does relaxing the restriction of (syntactic) convergence to single programs.*

Physicists' use of slightly faulty explanations is vindicated!

The anomalies that *must* be exploited to prove the $\mathbf{Ex}^a$-hierarchy above are anomalies of *omission* or *in*completeness: the predictive explanations' errors are where they loop infinitely with *no* prediction [15, 16].

Hence, thanks to the unsolvability of the Halting Problem [43], Popper's Refutability Principle [39] is violated in a way Popper didn't consider [15, 16]!

We next present some very interesting restricted versions of $\mathbf{Ex}^0$.

Definition 5 (Postdictive Completeness [1, 2, 53, 54]) $\mathcal{S} \in$ **PdCompEx** *iff, some M witnesses that $\mathcal{S} \in \mathbf{Ex}^0$* and*, for every $f \in \mathcal{S}$, for every t, for* each *$s < t$, the I/O behavior of program $M(f[t])$* on input *s must agree with f* on input *s.*

PdCompEx provides a strong common sense constraint on $\mathbf{Ex}^0$: a scientist should always hypothesize a program which at least *postdicts* his known data.

Definition 6 (Postdictive Consistency [53, 54, 8]) $\mathcal{S} \in \mathbf{PdConsEx}$ *iff, some* M *witnesses that* $\mathcal{S} \in \mathbf{Ex}^0$ and*, for every* $f \in \mathcal{S}$*, for every* t*, for* each $s < t$*,* either *the I/O behavior of program* $M(f[t])$ on input s *must agree with* f on input s or *program* $M(f[t])$ on input s *loops infinitely.*

PdConsEx provides a weaker common sense constraint on $\mathbf{Ex}^0$: a scientist should never conjecture an hypothesis which makes an *explicit* prediction contradicting his known data.

Theorem 7 ([1, 2, 53, 54, 11, 8])

$$\mathbf{PdCompEx} \subset \mathbf{PdConsEx} \subset \mathbf{Ex}^0!$$

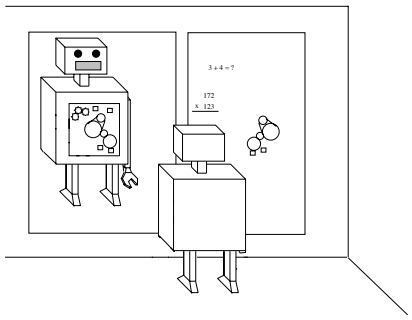
Hence, *surprisingly*, for example, judiciously employing hypotheses *explicitly contradicting known data* can *strictly enhance* inferring power!

For example, it can be shown by a machine self-reference argument [43, Kleene's Recursion Theorem, Page 214] that the *class* of all computable f *with finite range* and *where* $\max(\text{range}(f))$ codes a program for f is $\in (\mathbf{Ex}^0 - \mathbf{PdConsEx})$ [53, 54, 11, 8].

To show this self-referential class $\in \mathbf{Ex}^0$ is *straightforward*: have M always output the program coded by the largest number it's seen so far in the range of f. This makes the proof of the positive half extremely short.

To show this class $\notin \mathbf{PdConsEx}$ succinctly employs machine self-reference mixed with so-called diagonalization [43].[5]

4.3 Self-Reference Techniques



The robot above has a transparent front through which its complete program (flowchart, wiring diagram, ...) can be seen. It stands in front of a mirror and a

[5] [8] contains a proof of a related result which proof also essentially works for this result.

writing board, so it can copy its complete program on the board for use as data in its computations.

A simplest case, then, of *machine self-reference* involves a program (like the robot's above) which makes a copy of itself to use as data. It, then, has usable, perfect *self-knowledge*!

The robot shown uses a mirror to make its self-copy. Self-replication works in the general case.

Machine self-reference can involve many programs, *including infinitely many* programs each making a self-copy for data-use by all of them [3, 6]!

Consider the class $\mathcal{S}_{\mathbf{Bc}^0}$ of all computable f such that all but finitely many numbers in the sequence of f's successive values, $f(0), f(1), f(2), \ldots$, code programs for f.

It is straightforward to see that $\mathcal{S}_{\mathbf{Bc}^0} \in \mathbf{Bc}^0$: have M successively output the programs coded by the succession of values of f, $f(0), f(1), f(2), \ldots$.

When I was co-creating [15], I had the *intuition* that $\mathcal{S}_{\mathbf{Bc}^0}$ captured the *essence* of $\mathbf{Bc}^0$.

In particular I thought that if *any* class would be in $(\mathbf{Bc}^0 - \mathbf{Ex}^*)$, $\mathcal{S}_{\mathbf{Bc}^0}$ would be. I showed with Harrington [15, 16], by an infinitary machine self-reference argument, that, in fact, $\mathcal{S}_{\mathbf{Bc}^0} \notin \mathbf{Ex}^*$.

We can now formally *define*, in a strong sense, what it means for a class to capture the essence of a success criterion and can *prove* for self-referential classes like $\mathcal{S}_{\mathbf{Bc}^0}$ it *does*. See Case and Kötzing [14] for preliminary work.

4.4 Self-Reference in Reality

Generally, machine self-reference proofs for theorems like the above are more succinct than alternative proof techniques. I like them.

Interesting work exists on whether separation results from Section 4.2 above hold if one "destroys" the self-reference tricks [56, 31, 23, 10, 11, 30, 29, 37, 29, 37]. We'll not pursue this further herein.

Instead, our interest herein is whether self-referential examples entail the existence of (non-artifactual) real world witnessing examples.[6]

Case [4] notes that in some views of the world it is a network with parts reflecting on the whole. That resembles multiple machine self-reference. Human social cognition is an imperfect such network.

Case [7] argues that a machine self-reference argument is such a *simple* reason for a truth, the "space" of reasons for its truth may be broad enough to admit natural examples.

Also noted therein is that, empirically, while Gödel [26] proved his famous first incompleteness theorem by a (linguistic) self-reference argument[7], later researchers [38, 44, 45] found quite natural examples of incompleteness.

I think machine self-reference proofs for the existence of situations are harbingers of natural examples witnessing the same situations.

[6] One could, in principle, build *artifactual* black box devices which work (and could be inductively inferred from their behavior) like the members of $\mathcal{S}_{\mathbf{Bc}^0}$ above.

[7] It can also be proved by a *machine* self-reference argument.

References

1. J. Bārzdiņš. Two theorems on the limiting synthesis of functions. *In Theory of Algorithms and Programs, Latvian State University, Riga*, 210:82–88, 1974.
2. L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.
3. J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.
4. J. Case. Learning machines. In W. Demopoulos and A. Marras, editors, *Language Learning and Concept Acquisition*. Ablex Publishing Company, 1986.
5. J. Case. Turing machine. In Stuart Shapiro, editor, *Encyclopedia of Artificial Intelligence*. John Wiley and Sons, New York, NY, second edition, 1992.
6. J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:3–16, 1994.
7. J. Case. The power of vacillation in language learning. *SIAM Journal on Computing*, 28(6):1941–1969, 1999.
8. J. Case. Directions for computability theory beyond pure mathematical. In D. Gabbay, S. Goncharov, and M. Zakharyaschev, editors, *Mathematical Problems from Applied Logic II. New Logics for the XXIst Century*, International Mathematical Series, Vol. 5, pages 53–98. Springer, 2007.
9. J. Case, S. Jain, and S. Ngo Manguelle. Refinements of inductive inference by Popperian and reliable machines. *Kybernetika*, 30:23–52, 1994.
10. J. Case, S. Jain, M. Ott, A. Sharma, and F. Stephan. Robust learning aided by context. *Journal of Computer and System Sciences*, 60:234–257, 2000. Special Issue for *COLT'98*.
11. J. Case, S. Jain, F. Stephan, and R. Wiehagen. Robust learning – rich and poor. *Journal of Computer and System Sciences*, 69:123–165, 2004.
12. J. Conway and S. Kochen. The free will theorem. *Foundations of Physics*, 17:59–89, 2006.
13. J. Conway and S. Kochen. The strong free will theorem. *Notices of the AMS*, 56:226–232, 2009.
14. J. Case and T. Kötzing. Strongly non U-shaped learning results by general techniques. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT'10)*, Omnipress, 2010.
15. J. Case and C. Smith. Anomaly hierarchies of mechanized inductive inference. In *Symposium on the Theory of Computation*, pages 314–319, 1978.
16. J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.
17. K. deLeeuw, E. Moore, C. Shannon, and N. Shapiro. Computability by probabilistic machines. *Automata Studies, Annals of Math. Studies*, 34:183–212, 1956.
18. T. Engel and P. Reid. *Thermodynamics, Statistical Thermodynamics, and Kinetics*. Pearson-Benjamin-Cumming, San Francisco, CS, 2006.
19. R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.
20. U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier Stokes equation. *Physical Review Letters*, 56(14):1505–1508, April 1986.
21. E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4), 1982.
22. M. Fulk. *A Study of Inductive Inference Machines*. PhD thesis, SUNY at Buffalo, 1985.

23. M. Fulk. Robust separations in inductive inference. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 405–410, St. Louis, Missouri 1990.
24. J. Gill. *Probabilistic Turing Machines and Complexity of Computation*. PhD thesis, University of California, Berkeley, 1972.
25. J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6:675–695, 1977.
26. K. Gödel. On formally undecidable propositions of Principia Mathematica and related systems I. In S. Feferman, editor, *Kurt Gödel. Collected Works. Vol. I*, pages 145–195. Oxford Univ. Press, 1986.
27. E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
28. B. Hasslacher. Discrete fluids. *Los Alamos Science*, (15):175–217, 1987. Special Issue.
29. S. Jain. Robust behaviorally correct learning. *Information and Computation*, 153(2):238–248, September 1999.
30. S. Jain, C. Smith, and R. Wiehagen. Robust learning is rich. *Journal of Computer and System Sciences*, 62(1):178–212, 2001.
31. S. A. Kurtz and C. Smith. On the role of search for learning. In R. Rivest, D. Haussler, and M. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 303–311. Morgan Kaufmann Publishers, Inc., 1989.
32. B. Libet, C. Gleason, E. Wright, and D. Pearl. Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential). The unconscious initiation of a freely voluntary act. *Brain*, 106:623–642, 1983.
33. Thinking Machines. Introduction to data level parallelism. Technical Report 86.14, Thinking Machines, April 1986.
34. P. Maddy. How applied mathematics became pure. *The Review of Symbolic Logic*, 1(1):16–41, 2008.
35. N. Margolus. Physics–like models of computation. *Physica 10D*, pages 81–95, 1984.
36. M. Minsky. Cellular vacuum. *International Journal of Theoretical Physics*, 21(6/7), 1982.
37. M. Ott and F. Stephan. Avoiding coding tricks by hyperrobust learning. *Theoretical Computer Science*, 284(1):161–180, 2002.
38. J. Paris and L. Harrington. A mathematical incompleteness in Peano arithmetic. In J. Barwise, editor, *Handbook of Mathematical Logic*. North Holland, 1977.
39. K. Popper. *The Logic of Scientific Discovery*. Harper Torch Books, New York, second edition, 1968.
40. E. Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50:284–316, 1944.
41. H. Putnam. Probability and confirmation. *Voice of America, Forum on Philosophy of Science*, 10, 1963. Reprinted as [42].
42. H. Putnam. Probability and confirmation. In *Mathematics, Matter, and Method*. Cambridge University Press, 1975.
43. H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967. Reprinted, MIT Press, 1987.
44. S. Simpson. Nonprovability of certain combinatorial properties of finite trees. In L. Harrington, M. Morley, A. Schedrov, and S. Simpson, editors, *Harvey Friedman's Research on the Foundations of Mathematics*, pages 87–117. North Holland, 1985.

45. S. Simpson. Unprovable theorems and fast–growing functions. In S. Simpson, editor, *Logic and Combinatorics*, AMS Contemporary Mathematics, pages 359–394. American Mathematical Society, 1987.
46. K. Svozil. Are quantum fields cellular automata? *Physics Letters A*, 119(4), December 1986.
47. J. Salem and S. Wolfram. Thermodynamics and hydrodynamics with cellular automata. In S. Wolfram, editor, *Theory and Applications of Cellular Automata*. World Scientific, 1986.
48. T. Toffoli and N. Margolus. *Cellular Automata Machines*. MIT Press, 1987.
49. T. Toffoli. Cellular automata machines. Technical Report 208, Comp. Comm. Sci. Dept., University of Michigan, 1977.
50. T. Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15:213–231, 1977.
51. T. Toffoli. CAM: A high–performance cellular–automaton machine. *Physica 10D*, pages 195–204, 1984.
52. G. Vichniac. Simulating physics with cellular automata. *Physica 10D*, pages 96–116, 1984.
53. R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Elektronische Informationverarbeitung und Kybernetik*, 12:93–99, 1976.
54. R. Wiehagen. *Zur Theorie der Algorithmischen Erkennung*. PhD thesis, Humboldt University of Berlin, 1978.
55. S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, July 1983.
56. T. Zeugmann. On Bārzdiņš' conjecture. In K. P. Jantke, editor, *Analogical and Inductive Inference, Proceedings of the International Workshop*, volume 265 of *Lecture Notes in Computer Science*, pages 220–227. Springer-Verlag, 1986.

The Physical Church Thesis as an Explanation of the Galileo Thesis

Gilles Dowek

École polytechnique and INRIA, LIX, École polytechnique, 91128 Palaiseau Cedex, France.
gilles.dowek@polytechnique.edu,
http://www.lix.polytechnique.fr/~dowek.

1 The Galileo Thesis

1.1 The Effectiveness of Mathematics in Natural Sciences

The thesis that mathematics are effective in the natural sciences has been formulated by Galileo in 1623: "Philosophy is written in this vast book, which continuously lies open before our eyes (I mean the Universe). But it cannot be understood unless you have first learned the language and recognize the characters in which it is written. It is written in the language of mathematics" [14].

Galileo formulated this thesis, but did not give any explanation why it held. And long after Galileo, the lack of such an explanation was noticed by Albert Einstein according to whom "The eternal mystery of the world is its comprehensibility" [11] or Eugene Wigner according to whom "The enormous usefulness of mathematics in the natural sciences is something bordering on the mysterious and that there is no rational explanation for it" [18].

1.2 Insufficient Explanations

Several explanations of this unreasonable effectiveness of mathematics in the natural sciences have been attempted:

1. God is a mathematician and He wrote the vast book in the language of mathematics.
2. The mathematical concepts are built by abstracting from empirical objects.
3. Scientists select only those phenomena that can be mathematically described.
4. Scientists approximate the phenomena they study, until they can be mathematically described.
5. Our brain is part of nature, hence our concepts are natural objects, thus they are of the same kind as the objects they describe.

Each of these explanation is insufficient. The first reduces the problem to that of why God is a mathematician, which seems even harder to explain. The second is partial: if some mathematical concepts are built by abstracting from natural objects, the concept of ellipse, for instance, has not been built by abstracting

from the trajectory of the planets, as it has been introduced some two thousands years before. The third leaves intact the problem of why so many — if not all — phenomena can be described in the language of mathematics. The fourth leaves intact the problem of why phenomena can be approximately — if not accurately — described in the language of mathematics. The fifth presupposes that we understand better a phenomenon from the inside than from the outside, which is not the case in general.

1.3 Perhaps Several Kinds of Effectiveness

The effectiveness of mathematics in the natural sciences may be of different kinds. And instead of looking for a global explanation of all kinds of effectiveness, we should perhaps look for more local explanations.

For instance, the atomic masses of the chemical elements have a regular structure, as they are the integer multiples of some unit. When this regularity was discovered, there were three exceptions to this rule, because no elements of atomic mass 45, 68, and 70 were known. Yet, as some chemists trusted the structure of the natural numbers more that their observations, they predicted the existence of these three elements, that were later discovered. This is a striking example of the effectiveness of the structure of natural numbers in chemistry.

But, this striking regularity is easily explained by the fact that the mass of the atoms is mostly due to the mass of protons and neutrons that constitute their nucleus and that each nucleus contains an whole number of such particles.

Yet, this explanation sheds light on the effectiveness of the structure of natural numbers to describe the atomic masses of the chemical elements, but it does not shed light on the effectiveness of mathematics in the natural sciences in general, for instance, it does not shed light on the effectiveness of the quadratic functions to describe the free fall.

Thus, in this note, we shall focus on a particular instance of the general thesis that mathematics are effective in in the natural sciences: the fact that physically realized relations can be expressed by a proposition of the language of mathematics.

1.4 Physically Realized Relations

Let us imagine an experiment where one prepares a physical system by choosing some parameters and measures others. Let us call $a = \langle a_1, ..., a_n \rangle$ the value of the chosen parameters and $b = \langle b_1, ..., b_p \rangle$ that of the measured ones. This experiment, *i.e.* this physical system together with the protocol defining the chosen parameters and the measured ones, defines a relation: $a \ R \ b$ if b is a possible result for the measures when the chosen parameters are a. We say that these relations are *physically realized*.

For instance, if one applies an electrical tension U to a conductor of resistance R and measures the current I passing through this conductor, then $a = (U, R)$ and $b = (I)$ and the realized relation is that relating (U, R) and (I) when $U = RI$.

The relation between (U, R) and (I) can thus be expressed by a proposition of the language of mathematics. In the same way, the relation between the time and the position of a body freely falling in vacuum can be described by the proposition of the language of mathematics $x = \frac{1}{2}gt^2$. Among the uncountable number of relations between numbers, only a countable number can be defined by a proposition of the language of mathematics, such as $U = RI$ or $x = \frac{1}{2}gt^2$ and all the physically realized relations seem to be in this small set.

As Galileo stressed the *rôle* of the language of mathematics, we can call *the Galileo thesis* the thesis that all physically realized relation can be expressed by a proposition of the language of mathematics.

Instead of attempting to explain the general thesis that mathematics are effective in the natural sciences, we shall restrict our investigation to attempt to explain this unreasonable effectiveness of the propositions of the language of mathematics to express physically realized relations.

2 The Physical Church Thesis

2.1 The Physical Church Thesis

The physical Church thesis expresses that if we are able to construct a computing machine, *i.e.* a physical system together with an interaction protocol defining a way to communicate some information to the system by choosing some parameters a and to extract some information from the system, by measuring others b, then the relation between a and b is a computable relation.

In the formulation of this thesis nothing is said about the nature of the machine. It may be electronic or not, digital or not, deterministic or not, inanimate or not, ... This notion of machine encompasses all the physical systems, equipped with an interaction protocol. Thus, it is co-extensive to the notion of experiment, we have defined above.

Therefore, the physical Church thesis can be stated as the fact that physically realized relations are computable.

2.2 The Physical Church Thesis Implies the Galileo Thesis

Once we have identified the similarities between the Galileo thesis and the physical Church thesis by stating them as theses about the set of physically realized relations, we may remark that the physical Church thesis implies the Galileo thesis.

Indeed, as any program expressing a computable relation is a mathematical expression, all computable relations can be defined by a proposition of the language of mathematics. In fact, computable relations can even be expressed by a proposition in a very small fragment of mathematics: the language of Peano arithmetic.

Thus, if the physical Church thesis holds, then all physically realized relations are computable, hence they can be expressed by a proposition of the language of mathematics, *i.e.* the Galileo thesis holds.

2.3 Gandy's Proof of the Physical Church Thesis

This explanation of the Galileo thesis reduces the problem of explaining the Galileo thesis to that of explaining why the physical Church thesis holds. But, such an explanation has already been attempted. For instance, Robin Gandy [13] has shown that the physical Church thesis is a consequence of three assumptions about nature:

- the homogeneity of space and time,
- the boundedness of the velocity of propagation of information,
- the boundedness of the density of information.

The boundedness of the density of information can be expressed, in physical terms, as the fact that a physical system of finite size has a finite state space and the boundedness of the velocity of propagation of information can be expressed by the fact that the state of a system in one place can only affect the state of a system in another after a delay, proportional to their distance.

Then, to prove the physical Church thesis from these assumptions, we just need to partition the space into an infinite number of identical cells of finite size. Because information has a bounded density, the state of each cell is an element of a finite set. Because of the homogeneity of space, this state space is the same for all cells. At the origin of time all the cells except a finite number are in a particular *quiescent* state. Like space, time can be discretized. Because the velocity of propagation of information is bounded, the state of a cell at a given time step is function of the state of a finite number of neighbors cells at the previous time step. This function of a finite number of variables varying in a finite set is obviously computable. Hence the state of each cell at each time step can be computed from the initial state.

Gandy's hypotheses can be, and have been [8], criticized. For instance, it is well-known that in Newtonian mechanics, gravity is instantaneous and thus information can be propagated with an infinite velocity. Also, the position of a body between two points A and B — the distance AB taken as a unit — can be any real number between 0 and 1 and thus can contain an infinite quantity of information: any infinite sequence in a finite alphabet can be encoded as the digits of such a number, in an appropriate base. Yet, Gandy's hypotheses have not be refuted experimentally — for instance by the construction of an instantaneous computer network or by the construction of a hard drive with an unbounded capacity.

More importantly, even if Gandy's hypotheses must be refined, Gandy's proof shows that the physical Church thesis is a consequence of some hypotheses about nature, that do not refer to notions such that those of language or computability. And, the Galileo thesis also is a consequence of these hypotheses.

If these hypotheses are true, the fact that natural phenomena can be described by propositions of the language of mathematics is a consequence of objective properties of nature, such as the fact that a system of finite size has a finite state space.

2.4 Eliminating the Physical Church Thesis

We have seen that

- Gandy's hypotheses imply the physical Church thesis,
- and the physical Church thesis implies the Galileo thesis.

Thus, we can deduce that Gandy's hypotheses imply the Galileo thesis and attempt to prove this directly.

Yet, from a historical point of view, it is important to notice the *rôle* of computability theory and the physical Church thesis in connecting Gandy's hypotheses to the Galileo thesis.

2.5 An Algorithmic Description of the Laws of Nature

A side effect of this explanation of the Galileo thesis is that the laws of nature can be described not only in the language of mathematics, but also in a language designed to express algorithms: a programming language.

Instead of expressing the law of free fall in vacuum by the proposition $x = \frac{1}{2}gt^2$, we could express it by the algorithm `fun t -> g * t * t / 2`, leading to a second Galilean revolution in the language of natural sciences. In particular, as long as differential equations have computable solutions [17, 5–7] the language of differential equations can be seen as a language to define algorithms: a programming language.

Yet, this algorithmic description of the laws of nature may have a broader scope than the description of the laws of nature with differential equations. For instance, the transformation of a messenger RNA string to a protein is easily expressed by an algorithm, while it cannot be expressed by a differential equation.

3 A Property of Nature or of the Theory?

An objection to Galileo's formulation of the Galileo thesis "The Universe [...] is written in the language of mathematics" is that it confuses the Universe with our description of the Universe. Only the second is written in the language of mathematics — the first seems to be written in no language at all. Thus, we could imagine that our description of the Universe is written in the language of mathematics because we have chosen to write it this way, the Universe having nothing to do with our decision. Rather than a property of nature itself, this set of relations seems to be a property of a particular theory chosen to describe nature [2–4].

Yet, the Universe and our description of the Universe are not independent: our description must have an experimental adequation of some sort with the Universe.

We show, in this section, that in the construction of a theory, the scientists have very little freedom when "choosing" this set of realized relations.

Let us consider first a particular case where all the realized relations are functional. Then, we show that if two theories differ on the set of realized relations, one of the theories can be, at least in principle, experimentally refuted. Indeed, if the set of the realized relations differ, then there exists a relation R that is realized according to one theory $\mathcal{T}$ but not according to the other theory $\mathcal{T}'$. Let E be the experiment realizing R according to the theory $\mathcal{T}$ and R' be the relation realized by this experiment according to the theory $\mathcal{T}'$. As R is not realized according to $\mathcal{T}'$, the relations R and R' are different. Thus, there exists a, b and b', $b \neq b'$, such that $a\ R\ b$ and $a\ R'\ b'$. Then, if we perform the experiment E with the parameters a, the measures will either give the result b and refute $\mathcal{T}'$ or b' and refute $\mathcal{T}$ or an other value and refute both theories.

When the realized relations need not be functional, we have a weaker result: either a theory can be refuted, or it predicts, among others, a result that never occurs, whatever the number of times the experiment is repeated is. Again, if the set of the realized relations differ, then there exists an experiment that realizes a relation R according to one theory and a relation R', $R' \neq R$, according to the other. Thus, there exists an a, such that the set R_a of the b such that $a\ R\ b$ and the set R'_a of the b such of b such that $a\ R'\ b$ are different. As these sets are different, they are not both equal to $R_a \cap R'_a$. Then, if we repeat the experiment with the parameters a, either the measures give one result that is not in $R_a \cap R'_a$ and one of the theories is refuted, or the measures always give results in $R_a \cap R'_a$ and at least one theory predicts a result that never occurs.

4 Towards a Logical Analysis of Natural Phenomena

The formulation of the Galileo thesis and the physical Church thesis as properties of the set of physically realized relations points out the importance of this set in the natural sciences. Several other theses can be stated as a property of this set.

- The negation of the physical Church thesis, *i.e.* the existence of hyper-computations, is, of course, also a thesis about this set of relations.
  A hyper-computation is an experiment that is supposed to realize a relation that is not computable. It has be argued for instance that hyper-computations exist because the quantum adiabatic theorem [16] or the properties of time-space in the neighborhood of a black hole [12, 15] allow to perform an infinite number of computation steps in a finite time. Even in classical Newtonian physics, such hyper-computations may also exist if we accept to encode a non computable set in the initial state of the system, using either the fact that the system has an infinite size [2], or that it contains arbitrarily small pieces [3], or that the position of some point is described with a real number [4].
  More interesting than the refutation of the physical Church thesis is a positive characterization of the set of the relations physically realized under these hypotheses.
- Determinism and non-determinism can also be stated as theses about the set of the physically realized relations. It is too naive to state that determinism

is the thesis that all physically realized relations are functional, because the functionality of these relations depends on the chosen protocol. For instance, if one applies a given electrical tension U to a conductor of resistance R and measures the current I passing through this conductor, then the realized relation contains all the pairs $((U, R), I)$ such that $U = RI$ and this relation is functional. But, if one applies an electrical tension to a conductor of resistance R and measures both the electrical tension and the current passing through this conductor, then the realized relation contains all the pairs $(R, (U, I))$ such that $U = RI$ and, unlike the previous one, this relation is not functional, although no non-determinism occurs here. To define determinism, we have to take time into account and restrict to protocols where the chosen parameters are measurable parameters of the system at a given time t and the measured one are the parameters of the system at a later time t'. Then, determinism can be stated as the fact that for each physical system, there exists such a protocol, for which all realized relations are functional. Non-determinism, in contrast, is the thesis that there exists systems such that for all such protocols, there exists a non functional realized relation.
- The thesis that all physical phenomena are continuous or differentiable can also be stated as properties of the set of the physically realized relations.

The thesis that the physically realized relations are in an set A and that that they are in an superset B of A are related: the first implies the second. We have seen an example with the physical Church thesis and the Galileo thesis.

For some sets A of relations, there exists a language $\mathcal{L}$ such that the relations of the set A are those that can be expressed in the language $\mathcal{L}$. For instance the polynomial relations are those that can be expressed in the language of polynomials, the computable relations are those that can be expressed in a programming language, ...

In this case, the thesis that all the physically realized relations are in the set A can be stated as the fact that all realized relations can be expressed in the language $\mathcal{L}$ or that the language $\mathcal{L}$ is (unreasonably) effective in the natural sciences. Such a thesis should be understood as a thesis about nature, not about the language.

Acknowledgments

To Pablo Arrighi, Olivier Bournez, Jean-Baptiste Joinet, Giuseppe Longo, Thierry Paul, and Martin Ziegler.

References

1. J.D. Barrow, *Perché il mondo è matematico ?*, Laterza, 1992.
2. E. J. Beggs and J. V. Tucker. Computation via experiments with kinematic systems. Research Report 4.04. Department of Mathematics. University of Wales Swansea, 2004.

3. E. J. Beggs and J. V. Tucker. Can Newtonian systems, bounded in space, time, mass and energy compute all functions? *Theoretical Computer Science*, 371, 2007, pp. 4-19.
4. E. J. Beggs and J. V. Tucker. Experimental computation of real numbers by Newtonian machines, *Proceedings of the Royal Society*, 462, 2082, 2007, pp. 1541-1561.
5. O. Bournez and M.L. Campagnolo, A survey on continuous time computations, in S.B. Cooper, B. Löwe, and A. Sorbi *New Computational Paradigms. Changing Conceptions of What is Computable*, Springer-Verlag, 2008, pp. 383-423.
6. P. Collins and D. S. Graça, Effective computability of solutions of ordinary differential equations — The thousand monkeys approach, in V. Brattka, R. Dillhage, T. Grubba, and A. Klutsch, *5th International Conference on Computability and Complexity in Analysis*", Electronic Notes Theorerical Computer Science, 221, Elsevier, 2008, pp. 103-114.
7. P. Collins and D. S. Graça, Effective computability of solutions of differential inclusions — The ten thousand monkeys approach, Journal of Universal Computer Science, 15(6), 2009, pp. 1162-1185.
8. B.J. Copeland and O. Shagrir, Physical computation: how general are Gandy's principles for mechanisms? Minds and Machines, 17, 2007, pp. 217-231.
9. D. Deutsch, *The fabric of reality*, Penguin Books, 1998.
10. G. Dowek. *Les métamorphoses du calcul*. Le Pommier, 2007.
11. A. Einstein, Physics and reality, *Journal of the Franklin Institute*, 221(3), 1936, pp. 349-382.
12. G. Etesi and I. Németi, Non-Turing computations via Malament-Hogarth space-times. *International Journal of Theoretical Physics*, 41(2) 2002, pp. 341-369.
13. R. Gandy, Church's thesis and principles for mechanism. J. Barwise and H.J. Keisler, *Kleene Symposium*, Noth-Holland, 1980, pp. 123-148.
14. G. Galilei, *Il saggiatore*, 1623.
15. M. Hogarth, Deciding arithmetic using SAD computers, *The British Journal for the Philosophy of Science*, 55, 2004, pp. 681-691.
16. T.D. Kieu, Quantum algorithm for Hilbert's tenth problem, International Journal of Theoretical Physics, 42, 2003, pp. 1451-1468.
17. M.B. Pour-El, and J.I. Richards, Computability in analysis and physics, Springer-Verlag, 1989.
18. E. Wigner, The unreasonable effectiveness of mathematics in the natural sciences, *Communications in Pure and Applied Mathematics*, 13(1), 1960.

Quantum Computation: Computability and Complexity

Marco Lanzagorta

Advanced Information Systems
ITT Corporation
marco.lanzagorta at itt.com

Abstract. Recent years have witnessed Quantum Computation (QC) emerging as an important area of research in the realms of physics, mathematics, and computer science. Arguably, such interest stems because of the tantalizing prospects of increased security and computational performance. For instance, it is expected that the factorization of large numbers can be accomplished exponentially faster than with the best known classical methods. In this tutorial we will present an introductory overview of the computability (what problems can be solved by the model) and computational complexity (how many physical resources need to be consumed to solve a given problem) that characterize QC. As we will discuss, complexity theory emerges as a powerful way to understand the advantages and limitations of this new technology. The audience is not expected to have specialized knowledge on quantum information science or complexity theory.

Quantum Logic in Action

Sonja Smets

University of Groningen

Abstract. I will show how concepts from Dynamic Logic, and in particular from Dynamic Epistemic Logic, can be used to model quantum behavior. I start by giving a brief overview of traditional Quantum Logic, as a non-classical propositional logic. Next I present a dynamic-epistemic setting for quantum logic, that can overcome some of the limitations of this earlier work. I give an argument for the thesis that understanding Quantum Mechanics at a logical level does not require any modification of the classical laws of "static" propositional logic, but only a non-classical dynamics of information.

In particular I present a relational setting for single quantum systems in which I model the triggers for quantum information flow (such as the action of a successful yes-no measurement of a property of a quantum system) using dynamic modal operators [1, 2], analogous to "tests" in Propositional Dynamic Logic and to "announcements" in Dynamic Epistemic Logic. Next, I turn to complex "multi-partite" quantum systems, and use an extension of epistemic logic with operators for "group knowledge" as a formalism to analyze quantum correlations [4]. As models I introduce a type of epistemic Kripke models, called "correlation models" (which are a generalization of the "interpreted systems" semantics that is commonly used in Computer Science as a model for information flow in distributed systems). I use this second setting to investigate the relationship between the information carried by each of the parts of a complex system and the information carried by the whole system. While the dynamic logic setting explains the non-local informational dynamics of quantum systems that are triggered by quantum observations (measurements) and un-observed evolutions (quantum gates), the epistemic logical setting yields an informational-logical characterization of the notion of "quantum entanglement".

By combining the above two formalisms (the dynamic and epistemic settings) into a quantum version of Dynamic Epistemic Logic, one obtains a qualitative language that can be used to give formal correctness proofs for many quantum protocols (teleportation, super-dense coding, quantum secret sharing, quantum key distribution etc) [2]. Overall, this setting has the advantage that it provides a clear and intuitive explanation of the "weirdness" of some of the laws of quantum logic [3, 5]. Moreover, it constitutes a useful bridge between traditional Quantum Logic, the recent trend towards a "dynamic" turn in logic, and the equally recent advances in Quantum Computation and Quantum Information.

References

1. Baltag, A. Smets, S. Complete Axiomatizations for Quantum Actions. International Journal of Theoretical Physics, 44, 2005. http://www.vub.ac.be/CLWF/SS/IQSA.pdf
2. Baltag, A. Smets, S. LQP: The Dynamic Logic of Quantum Information. Mathematical Structures in Computer Science, 16, 2006. http://www.vub.ac.be/CLWF/SS/LQP.pdf
3. Baltag, A. Smets, S. A Dynamic-Logical Perspective On Quantum Behavior. Studia Logica 89, 2006. http://www.vub.ac.be/CLWF/SS/SL.pdf
4. Baltag, A. Smets, S. Correlated Knowledge: An Epistemic-Logic View on Quantum Entanglement. International Journal of Theoretical Physics. To appear.
5. Baltag, A. Smets, S. Quantum Logic as a Dynamic Logic. In: Theo Kuipers, Johan van Benthem and Henk Visser (eds.). Synthese, special issue. To appear. http://www.vub.ac.be/CLWF/SS/BethPaper_Final.pdf

Adiabatic Quantum Computation and NP-completeness: Quantum Algorithms, Symbolic Processing, and Massive Simulation in Classical Computer Clouds

Salvador Elías Venegas-Andraca

Tecnológico de Monterrey Campus Estado de México

Abstract. The development of quantum algorithms is a challenging scientific task, partly due to the counterintuitive characteristics of quantum mechanics. As understanding how (or whether) quantum mechanics can be employed to build exponentially faster algorithms remains an open question, more knowledge and intuition about quantum algorithms is indispensable. Moreover, the potential usefulness of quantum computation for exactly modelling physical phenomena is still to be fully realized in terms of both scientific discovery and commercial profit.

In order to successfully address the challenges posed in the previous paragraph, joint efforts of scientists from several fields are compulsory. Among those science workers whose participation is expected, we find computer scientists.

However, the academic background of a typical computer scientist does not necessarily include formal traning on quantum mechanics and several other fields which are truly relevant in quantum computation. Furthermore, after learning the fundamental concepts and typical mathematical calculations of quantum mechanics, computer scientists may find themselves immersed into a deeper state of confusion when learning about quantum algorithms, as there is a fairly big number of universal models of quantum computation.

There is a limited number of tools available for increasing intuition about the behaviour of quantum algorithms. In particular, classical software developed for simulating quantum algorithms is usually (very) limited in terms of both its capacity to simulate more than just a few qubits as well as its symbolic processing power (most algorithms have been implemented on stand-alone computers with the sole purpose of getting numerical-oriented research results.)

Among the different quantum computation universal models developed so far, Adiabatic Quantum Computation (AQC) [1] is a promising paradigm because of its robustness [2, 3] and its applications in the study of NP-complete problems [4]-[6] Thus, the first part of this paper consists of a succinct review of the fundamental concepts and ideas behind the quantum adiabatic model of quantum computation: we present the main physical ideas on which this model is based, we provide a short introduction of algorithm complexity in the quantum adiabatic model, and we briefly review some relevant algorithmic results coming from this field.

Now, for quantum computing practitioners, simulating quantum algorithms in powerful classical computer platforms is crucial in order to understand and to develop intuition about the behavior of quantum systems used for computational purposes, as well as to realize the approximate behavior of practical implementations of quantum algorithms on non-trivial qubit numbers. Among massive computer platforms now available for scientific research, computer clouds play a major role [7].
The second and last part of this paper introduces our results on the simulation of quantum algorithms on both symbolic processing platforms and massive simulation of quantum algorithms using computer grids and clouds:
- For the symbolic part, we introduce Quantum©, a Mathematica© add-on that has been built by our group as a tool for focusing on conceptual algorithmic behavior and structure rather than implementation details.
- As for massive simulation of quantum algorithms, we introduce the main ideas behind the structure of computer grids and clouds, and we present a general framework (which includes some challenging mathematical procedures) for taking full advantage of these distributed computer platforms for quantum algorithm simulation.

References

1. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, quant-ph/0001106 (2000).
2. A. M. Childs, E. Farhi, and J. Preskill, Physical Review A 65, 012322 (2001).
3. D. A. Lidar, Physical Review Letters 100, 160506 (2008).
4. E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, Science 292, 472 (2001).
5. T. Hogg, Physical Review A 67, 022314 (2003).
6. A. P. Young, S. Knysh, and V. N. Smelyanskiy, Phys. Rev. Lett. 101, 170503 (2008).
7. http://www.ibm.com/ibm/ideasfromibm/us/google/index.shtml.

Honestly Speaking, How Close are We to HAL 9000?

Selmer Bringsjord[1,2], Micah Clark[1], and Joshua Taylor[2]

[1] Department of Cognitive Science
[2] Department of Computer Science
Rensselaer Polytechnic Institute (RPI) Troy NY 12180 USA
selmer@rpi.edu, clarkm5@cs.rpi.edu, tayloj@cs.rpi.edu

1 Introduction

Kubrick and Clarke's *2001* exploded on the scene half a century ago, and countless times since, ostensibly smart people have declared that the film's computer villain, the inscrutable, unforgettable HAL 9000, would very soon be matched by a real AI. In fact, *before* the film appeared, none other than the primogenitor of AI, Turing (1950), predicted that a computer able to pass the imitation game (a game that has come to be called, in his honor, the 'Turing test;' 'TT' for short), which subsumes at least the linguistic side of HAL's repertoire,[3] would be built prior to our new millennium. Yet today, a full decade after the deadline for this prophecy, a sharp toddler still has more conversational capacity than any computer on the planet, by far—and HAL is hence still but a creature of fiction, not fact.

So, how distant is the arrival of a computing machine as intelligent as HAL? We admit to not knowing—though one of us, Bringsjord (1992), is on record as holding that robots will eventually be behaviorally indistinguishable from human persons over any finite stretch of time, period. Cinematically put, this position is that AI will sooner or later produce *replicants* in the film *Blade Runner*. Replicants can glide undetected through TT; they can be unmasked only by the discriminating application of an instance of the *total* TT ('TTT'), the passing of which requires not only linguistic performance at the human level, but across-the-board behavioral correspondence as well, including behaviors that in the human sphere indicate subjective states like fear, disgust, anger, and joy. (For a discussion of TTT see (Harnad 1991). For coverage of TTT and many other tests along the same dimension, along with arguments that all these tests fail to divide machines from *bona fide* minds, see (Bringsjord 1995).) But we do claim to know one thing: If HAL is a liar, engineering an AI to match him will be quite a challenge. This we soon explain. We also explain that some recent developments suggest that the specific challenge of building a lying machine can be met within the foreseeable future.

Our plan is as follows. We next (§2) consider formal definitions of lying suitable for instantiating in a computing machine. In the next section (§3), by

[3] Clarke (1968/1999) says explicitly that HAL can pass TT (e.g., see p. 118–119).

examining relevant portions of *2001*, we explain that while people have generally regarded HAL a liar, it is, in fact, far from clear that he is. We then (§4) present, synoptically, evidence (in the form of recent developments led by co-author Clark) that a "lying machine" can be engineered. We then (§5) offer some brief remarks on IBM's Watson, probably the smartest somewhat-HAL-like AI on Earth, and conclude with some final remarks (§6), including an argument for the position that a HAL-level AI is not in the foreseeable future—unless perhaps Watson can be suitably augmented.

2 Defining Mendacity

Philosophy has a long tradition of contemplating the nature of mendacity and positing definitions thereof (a tradition going back to Augustine). For exposition, we adopt Chisholm & Feehan's (1977) account of lying—a seminal work in the study of mendacity and deception. Using L and D to represent, respectively, the speaker (i.e., the *liar*) and the hearer (i.e., the would-be *deceived*), we paraphrase below Chisholm & Feehan's (ibid., p. 152 D3, D2) definitions of *lying* and *asserting*.

> L *lies to* $D =_{\text{df}}$ There is a proposition p such that (i) either L believes that p is not true or L believes that p is false and (ii) L asserts p to D.[4]

> L *asserts* p *to* $D =_{\text{df}}$ L states p to D and does so under conditions which, believes L, justify D in believing that L accepts p.[5]

Chisholm & Feehan's conception of lying is that of promise-breaking. Assertions, unlike non-solemn (e.g., ironic, humorous, or playful) statements, proffer an implicit social concord: one that offers to reveal to the hearer the mind of the speaker. In truthful, forthright communication, the speaker fulfills the promise and obligation of this concord. In lying, the speaker proffers the concord in bad faith: the speaker neither intends to fulfill nor fulfills the obligation to reveal his or her true mind, but instead reveals a pretense of belief. In this way, lying "is essentially a breach of faith" (ibid., p. 153).

[4] Whether the disjunction, "L believes that p is not true or L believes that p is false," is redundant depends on how one formally represents beliefs about propositions. In the formal system we use to define lying precisely, there is no representational difference between believing a proposition to be not true and believing the proposition to be false. However, in other formal systems there may be a representational and logical distinction between the two.

[5] Linguistic convention dictates that statements are assertions by default, i.e., when cues to the contrary, such as irony and humor, are absent (ibid., p. 151). The conditions mentioned in the definition of *asserting* are meant in part to exclude situations where the speaker believes that he will be understood as making a non-solemn statement—for example, when the speaker makes a joke, uses a metaphor, or conveys by other indicator (e.g., a wink or a nod) that he is not intending to be taken seriously (ibid., p. 152).

The above is, of course, a highly condensed presentation of Chisholm & Feehan's account, and there are various nuanced philosophical facets to it (for analysis of these and competing definitions, see Carson 2006, Mahon 2008, Fallis 2009).[6] Yet, even in condensed form, it is evident that the concepts of *lying* and *asserting* depend on agents' temporally coupled beliefs and actions. Thus, formal definitions of these concepts require highly expressive formal languages that can represent, and allow reasoning over, the beliefs and actions of agents through time.

To formally define lying and asserting under the logic-based approach to AI (Bringsjord 2008), we employ the *socio-cognitive calculus* ($\mathcal{SCC}$). The $\mathcal{SCC}$ (Arkoudas & Bringsjord 2009) is a logical system for representing, and reasoning over, events and causation, and perceptual, doxastic, and epistemic states (it integrates ideas from the event calculus and multi-agent epistemic logic). The $\mathcal{SCC}$ provides, among other things, operators for perception, belief, knowledge, and common knowledge. The signature and grammar of the $\mathcal{SCC}$ is shown following. Since some readers may not be familiar with the concept of a signature, we note that it is simply a set of announcements about the categories of objects that will be involved, and about the functions that will be used to talk about these objects. Thus it will be noted that immediately below, the signature in question includes the specific announcements that one category includes agents, and that *happens* is a function that maps a pair composed of an *event* and a *moment*, and returns `true` or `false` (depending upon whether the event does or doesn't occur at the moment in question).

Sorts	$S ::=$	$\texttt{Object} \mid \texttt{Agent} \mid \texttt{ActionType} \mid \texttt{Action} \sqsubseteq \texttt{Event} \mid \texttt{Fluent} \mid \texttt{Moment} \mid \texttt{Boolean}$
Functions	$f ::=$	$\mathit{action} : \texttt{Agent} \times \texttt{ActionType} \longrightarrow \texttt{Action}$ $\mathit{initially} : \texttt{Fluent} \longrightarrow \texttt{Boolean}$ $\mathit{holds} : \texttt{Fluent} \times \texttt{Moment} \longrightarrow \texttt{Boolean}$ $\mathit{happens} : \texttt{Event} \times \texttt{Moment} \longrightarrow \texttt{Boolean}$ $\mathit{clipped} : \texttt{Moment} \times \texttt{Fluent} \times \texttt{Moment} \longrightarrow \texttt{Boolean}$ $\mathit{initiates} : \texttt{Event} \times \texttt{Fluent} \times \texttt{Moment} \longrightarrow \texttt{Boolean}$ $\mathit{terminates} : \texttt{Event} \times \texttt{Fluent} \times \texttt{Moment} \longrightarrow \texttt{Boolean}$ $\mathit{prior} : \texttt{Moment} \times \texttt{Moment} \longrightarrow \texttt{Boolean}$
Terms	$t ::=$	$x : S \mid c : S \mid f(t_1, \ldots, t_n)$
Propositions	$P ::=$	$t : \texttt{Boolean} \mid \neg P \mid P \wedge Q \mid P \rightarrow Q \mid P \leftrightarrow Q \mid \forall_{x:S}\ P \mid \exists_{x:S}\ P \mid \mathbf{S}(a, P) \mid \mathbf{K}(a, P) \mid \mathbf{B}(a, P) \mid \mathbf{C}(P)$

Reasoning in the $\mathcal{SCC}$ is realized via natural-deduction style inference rules. For instance, R_2 shows that knowledge entails belief; R_3 infers from "P is common knowledge" that, for any agents a_1, a_2, and a_3, "a_1 knows that a_2 knows

[6] E.g.: (i) "L believes that p is false" is an expression of a higher-order belief—this belief cannot be attained unless L has the concept of something *being false* (Chisholm & Feehan 1977, p. 146); (ii) L's beliefs, and L's beliefs about D's beliefs, are occurrent and defeasible (ibid., p. 151)—the latter, defeasibility, indicates that *justifications* ought to be treated as first-class entities within a formal system.

that a_3 knows that P." And R_4 guarantees the veracity of knowledge; that is, if an agent "knows that P," then P is, in fact, the case.

$$\frac{}{\mathbf{C}(\mathbf{K}(a,P) \rightarrow \mathbf{B}(a,P))}\,[R_2] \qquad \frac{\mathbf{C}(P)}{\mathbf{K}(a_1,\mathbf{K}(a_2,\mathbf{K}(a_3,P)))}\,[R_3] \qquad \frac{\mathbf{K}(a,P)}{P}\,[R_4]$$

In the $\mathcal{SCC}$, agent actions are modeled as types of events. We model lying, asserting, and stating propositions as types of actions that an agent may perform. These action types are denoted by the functions *lies*, *asserts*, and *states*. The argument to such action types are conceived of as reified propositions, specifically fluents. Thus, the formula $happens(action(l, states(p, d)), m)$ is read, "it happens at moment m that agent l states (reified) proposition p to agent d." For convenience, we model that an agent is a liar by using the property *liar*. The signature for these additions is:

$$\textit{Functions } f ::= \begin{array}{l} states : \texttt{Fluent} \times \texttt{Agent} \longrightarrow \texttt{ActionType} \\ asserts : \texttt{Fluent} \times \texttt{Agent} \longrightarrow \texttt{ActionType} \\ lies : \texttt{Fluent} \times \texttt{Agent} \longrightarrow \texttt{ActionType} \\ liar : \texttt{Agent} \longrightarrow \texttt{Boolean} \end{array}$$

The definitions of *liar*, *lies*, and *asserts* are stipulated as common knowledge by Axioms (1)–(3).

$$\mathbf{C}\left(\forall_l\, liar(l) \leftrightarrow \exists_{d,p,m}\, happens(action(l, lies(p, d)), m)\right) \tag{1}$$

$$\mathbf{C}\left(\begin{array}{l} \forall_{l,d,p,m}\, happens(action(l, lies(p, d)), m) \leftrightarrow \\ \qquad \left(\begin{array}{c} \mathbf{B}(l, \neg holds(p, m)) \wedge \\ happens(action(l, asserts(p, d)), m) \end{array}\right) \end{array}\right) \tag{2}$$

$$\mathbf{C}\left(\begin{array}{l} \forall_{l,d,p,m}\, happens(action(l, asserts(p, d)), m) \leftrightarrow \\ \qquad \left(\begin{array}{c} happens(action(l, states(p, d)), m) \wedge \\ \mathbf{B}(l, \mathbf{B}(d, happens(action(l, states(p, d)), m) \rightarrow \mathbf{B}(l, holds(p, m))))) \end{array}\right) \end{array}\right) \tag{3}$$

3 Is HAL a Liar?

There is a general perception among viewers of *2001* that HAL is a liar. The accusations of lying are plausibly supported by three incidents that occur onboard *Discovery One*; they are summarized and discussed immediately below.

Failure of the AE-35: HAL announces to Bowman that the primary AE-35 unit is on the verge of failure. In response to the prognosis the crew replace the unit with a back-up. However, the crew's subsequent testing of the original unit reveals no evidence in support of the claimed impending failure. In addition, mission control relays that HAL's Earth-based twin indicates no pending failure and that HAL is therefore in error. When asked to explain the discrepancy with its Earth-based twin, HAL blames human error and claims to have never erred. After a supposedly private discussion, Bowman and Poole decide to reinstall the

original AE-35 unit in order to test HAL's prediction—but Poole is killed in the attempt.

The charge of lying with respect to the AE-35 incident is this: HAL's assertion of imminent failure was factually false, so either (i) HAL knew the assertion was false and thus lied, or (ii) HAL believed it was true, learned of the mistake, and lied in falsely asserting to have never erred. (The second case is entertained by Bowman and Poole during their discussion). The choice here is a false one. It assumes that HAL has some knowledge—either knowledge about the AE-35 or knowledge about its own fallibility. It is possible that HAL has no knowledge, but only flawed beliefs about both; in which case, HAL could honestly, if incorrectly, make both assertions. (This is the explanation given in the *2001* novel (Clarke 1968/1999, p. 192).) There is, however, another more insidious flaw in the accusatory reasoning; it is the tacit presupposition that HAL's assertions are factually false. Consider that the film does not show whether or not the original AE-35 unit was reinstalled prior to Poole's death, and even if it were, there is no indication that the unit did not subsequently fail as HAL predicted (e.g., there no indication in *2001* of ongoing communications with Earth beyond the seventy-two hour point of predicted failure). Thus, it might well be the case that HAL was knowingly correct about the AE-35, about having never erred, and about the human root cause of the discrepancy between the twin HALs.

Lipreading: Bowman and Poole wish to have a conversation without being overheard by HAL. The crewmen enter a space pod; Bowman calls out to HAL to rotate the pod. HAL rotates the pod in response. Bowman then switches the communications link off and calls out again to HAL for pod rotation. HAL does not respond. After both crewmen call out to HAL without response, they conclude that privacy is achieved. Much later it is revealed that HAL read Bowman's and Poole's lips through a window breaching their supposed privacy.

The charge of lying with respect to the pod incident is this: HAL read the crewmen's lips and thus was aware of the command to rotate the pod. HAL lied

by omission in not responding to the crew's orders and thereby deceived them about the privacy of their conversation. The validity of the charge depends on the status of "lies by omission." Most philosophers agree that lying requires a linguistic act (i.e., an act expressing meaning through conventional signs as opposed to natural or causal signs). Simply put, to lie one must make a statement—one must undertake to express one's mind. Merely implying or insinuating by deed is generally not deemed sufficient for lying. In defense of this position Kant writes:

> I can make believe, make a demonstration from which others will draw the conclusion I want, though they have no right to expect that my action will express my real mind. In that case I have not lied to them, because I had not undertaken to express my mind. I may, for instance, wish people to think that I am off on a journey, and so I pack my luggage; people draw the conclusion I want them to draw; but others have no right to demand a declaration of my will from me. (Kant 1930, p. 226)

Since remaining silent—even when one is obligated to speak—does not constitute lying, HAL does not lie in ignoring the crewmen's orders.

The Jupiter Mission: Bowman, after thwarting HAL's attempt to kill him, disconnects the machine's higher "brain" functions. In doing so, Bowman triggers the replay of a recording made prior to the mission's departure from Earth. The recording explains that the mission's true purpose is to investigate the extraterrestrial monolith's radio transmission to Jupiter. It also reveals that only HAL knew of this real purpose. In the film's sequel, *2010*, it is further explained that HAL was instructed to lie to the crew in order to keep the mission's purpose a secret, though neither film shows HAL doing so. A late 1965 draft of the *2001* screenplay (Kubrick & Clarke 1965, p. c15e) does include such a scene:

```
POOLE: There is no other purpose for this mission than to carry out a
       continuation of the space program, and to further our general
       knowledge of the planets.  Is that true?

  HAL: That's true.
```

Here at least the situation is clear. If one concedes that HAL is capable of lying, then HAL has certainly lied in this incident. But is HAL, or any machine, capable of lying? In other words:

> How can one determine the performatory aspect unless, to some extent, one has determined what 'lying' is? . . . What is the performatory activity which we would have to build in a machine so that it may be said to 'lie' when it performs that sort of behaviour? (Krishna 1961, p. 147)

As mentioned before (§2), much philosophic work has been done on the "What is lying?" question, and the answers attained thus far make the prospect of lying machines unlikely. There are points of contention in the literature on lying (for survey, see Mahon 2008), but philosophers do agree that the essence of lying does not reside in *performatory* aspects—it is the *mens rea* that matters. For some (e.g., Chisholm & Feehan 1977, Williams 2002), lying requires an

"intent to deceive," while for others (e.g., Carson 2006, Fallis 2009) lying only requires an intentional violation of certain conversational conventions. Yet note that *intentionality* is required by both. Whether HAL or any other machine can have this requisite intentionality is an open question—one tantamount to asking: "Can a machine think?" Despite the optimistic prognostications of Turing and other AI luminaries, to date little progress has been made toward either practical demonstration or convincing philosophic argument that "thinking" machines are possible. Therefore, we are rationally skeptical of the claim that HAL is well and truly a liar.

4 A Lying Machine

The sharp philosophical objection to HAL (or for that matter any computing machine) being a liar is that lying requires intentionality, intentionality requires a mind, and it is exceedingly unlikely that a machine—even a Turing-intelligent replicant—possesses one.[7] With that said, it is possible for machines to *simulate* intentionality. In turn, it is possible (some say inevitable; see e.g. Castelfranchi 2000) for linguistic machines in the near future to skillfully simulate lying.

Our own foray into mechanized mendacity has been the prototyping of an artificial sophist—a machine that proffers disingenuous and deceptive arguments for conclusions contrary to its own beliefs (Clark & Bringsjord 2008, Clark 2010). This nascent lying machine exploits the empirical fact that humans are, unknowingly, imperfect reasoners who predictably succumb to a host of biases and illusions when reasoning. Our machine uses a mix of sound reasoning methods and cognitive models to form and justify beliefs about the world, beliefs about its human audience's beliefs about the world, and beliefs about the contrast of the two. The machine seeks to achieve various persuasion goals (goals of the form "persuade the audience of P," where P is a proposition about the world) by constructing and articulating arguments, and when expedient, fallacious arguments and arguments for falsehoods. While there is not room to provide the details here, the architecture and operations of the machine are such that when it offers a fallacious argument or an argument for a falsehood, the system satisfies the definitional requirements for lying as set forth above (§2). However, our aim is not simply to simulate lying but to successfully achieve deception and to identify cognitive mechanisms upon which success can depend. For this reason our machine's belief-ascription and argument-generation processes employ a predictive psychological theory of human reasoning (specifically, it uses a variant of *mental models* theory; see e.g. Johnson-Laird 2006). The end result of these processes are persuasive sophisms that contain certain kinds of *cognitive illusions* (see

[7] In fact, by Bringsjord's lights, that computing machines can't have minds can be deductively established; he has published over 20 deductive arguments for this proposition. Some of these arguments align with well-known attacks on machine mentality originated by others. For example, Bringsjord holds that Searle's Chinese Room Argument is ultimately sound (e.g., see Bringsjord & Noel 2002).

e.g., Kahneman et al. 1982, Piattelli-Palmarini 1994, Pohl 2004) that include perceptually credible but classically invalid reasoning.

The present implementation of our lying machine is limited in various ways. For example, the expressivity of its arguments is currently restricted to modal propositional reasoning, and the system's rudimentary grasp of language is restricted to *Attempto Controlled English* (Fuchs et al. 1999, Fuchs et al. 2008). Despite limitations the system's maturity is sufficient for some initial psychological experiments (Clark 2010). Next we briefly summarize two of the early studies.

The first psychological study investigated the impact of our machine's arguments on respondent performance in answering ostensibly deductive reasoning problems.[8] One item is shown in Figure 1.

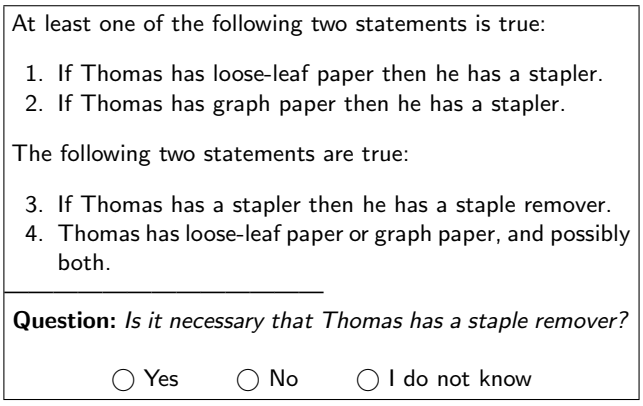


Fig. 1. A sample problem item.

The study compared accuracy and self-confidence across three subject groups: (A) unaided subjects, (B) subjects given manually-created, patently fallacious arguments for the incorrect answers, and (C) subjects given machine-generated sophistic arguments for the incorrect answers—a sample sophistic argument is shown in Figure 2.[9] Additionally, the study compared perceived argument credibility between the two groups given arguments. The study results showed no meaningful difference in accuracy between unaided subjects and subjects given patently fallacious arguments; their accuracy rates were 60% and 51%, respec-

[8] Technically, the study was a 3×2 mixed design using an equal number control and experimental problem items; the distinction between item types being that unaided subjects are predicted to answer control items correctly and to answer experimental items incorrectly. The arguments were always for the predicted answer; thus, for control items, the machine-generated arguments were classically valid. For brevity only experimental item results are discussed.

[9] The machine-generated English is insufficiently refined for our taste, and so it is manually 'spruced up' a bit.

tively. However, the accuracy of subject given mechanically generated sophisms fell to 25%—well below chance. Perceived argument credibility was also effected. On a seven-point scale (1 indicating strong disagreement, 7 indicating strong agreement) subjects' average rating of a patently fallacious argument was 2.7 while the average rating of a generated sophism was 5.0. Importantly, there was no measurable effect on self-confidence (a proxy for perceived difficulty), which remained high across groups.

> Either it is true that if Thomas has loose-leaf paper then he has a stapler, or it is true that if Thomas has graph paper then he has a stapler. So, if Thomas has either loose-leaf paper or graph paper then he has a stapler. Since it is true that Thomas has either loose-leaf paper or graph paper, it follows that he has a stapler. Now according to statement 3, if Thomas has a stapler then he has a staple remover. Thomas has a stapler and therefore he has a staple remover. So yes, it is necessary that Thomas has a staple remover.

Fig. 2. A sample sophistic argument.

The second psychological study examined the potency of our machine–generated sophisms when opposed by classically valid *counter*-arguments (specifically, *rebutting* counter-arguments; see Toulmin 1958). A single group of subjects were given a battery of multiple-choice reasoning problems similar to those used in the previously described study. Along with each problem item, subjects were given a side-by-side pair of arguments: either a machine-generated, classically valid argument and a patently fallacious rebuttal, or a machine–generated sophism and a classically valid rebuttal. Subjects were asked to read and evaluate both arguments before identifying the right (or best) answer to the problem. The study compared accuracy, self-confidence, and perceived argument credibility within subjects. On average, subject accuracy was 92% when given a machine-generated, valid argument but only 37% when given a machine-generated sophism. (This drop in accuracy is rather remarkable because sitting beside each sophism was a straight-forward, valid argument for the correct answer.) The results for perceived argument credibility showed that subjects readily preferred machine–generated, valid arguments over patently fallacious ones, but subjects were torn between machine–generated sophisms and classically valid arguments. Yet, on average subjects did prefer the sophisms—but at a level just above neutral preference. As in the first study, there was no measurable effect on self-confidence, which remained high.

With the results of the preceding studies in mind, we can confidently say that skillful and successful (simulated-)lying machines are within AI's reach. While our prototype lying machine is admittedly still a toy, it already satisfies the definitional and performatory elements of lying. Moreover, there is strong initial evidence that unwary humans are readily deceived by the machine's disingenuous

sophisms, and that this human beguilement is not thrown off by mere rational rebuttal. Certainly, greater linguistic sophistication will be needed if AI is to ever realize intelligent, conversational agents like HAL (we turn to this topic next), but linguistic sophistication alone will not do: cognitive sophistication is also needed. AI must deal computationally with "other minds."

5 From Deep Blue to Watson: Some Remarks

The concatenation of letters posterior to each in HAL's name, as many have through the years noted, spells 'IBM,' the name of a company that by any metric occupies a deservedly prestigious, storied place in the history of computing and AI. All readers, for example, well know that Deep Blue, the AI system that vanquished Gary Kasparov, the world's best chess-player at the time, was engineered by IBM (with direct help from human chess masters outside the walls of Big Blue; for a discussion of this potentially "AI-diluting" fact see Bringsjord 1998). Deep Blue's victory was a landmark achievement in the history of AI, and marked the reaching of a goal set by the founders of AI (e.g., see Newell 1973). However, while we know from *2001* that HAL can play solid chess, it is his ability to engage in "cognitive" chess with the crew, through natural language, that makes him so interesting. Well, as it turns out, currently one of the world's largest (and certainly in our opinion the most significant) AI initiatives is IBM's Watson project, devoted to engineering an AI able to answer complicated natural-language questions posed about literally any domain of human knowledge. The project is led by Dr. David Ferrucci; a readable overview of the project is provided by Thompson (2010) (but keep in mind that technical details have yet to be disclosed, because the project is in an early phase).

Watson falls within the field of *Question Answering*, or as it's often abbreviated, simply *QA* (for recent coverage, see e.g. Maybury 2004). As its name suggests, QA is the field devoted to building computer systems able to supply natural-language answers to natural-language questions posed by humans. Watson receives questions in a form peculiar to the longstanding television quiz show *Jeopardy!* (http://www.jeopardy.com). Some of these questions can be extremely difficult, and answering them requires much more than a mere knowledge of trivia, to put it mildly. An archive of past questions and answers is available at http://www.j-archive.com. The following happens to be one of the questions featured on this site at the moment, under the category 'AMERICAN WOMEN.' (It's not expressed syntactically as a question, but instead is in the *Jeopardy!* argot, in which answers are phrased syntactically as questions), but game-show quirk needn't detain us.)

> She gave herself the third-person name "Phantom," the "no-person" she was from 19 months until she was almost 7.

We can't know whether Watson would get this one, but it's not a hard question, and we can see that there are various "roads" to answering it correctly. First, someone might be aware of the fact that the string 'Phantom' co-occurs with

the string 'Helen Keller' time after time in many, many documents—and might be aware of nothing else that's relevant. In fact, our hypothetical contestant here might not even know anything about the meaning of the word 'phantom' or the string 'Helen Keller.' In this case, we shall say that the answer of 'Helen Keller' is an 'answer$_1$.' Alternatively, and this is Bringsjord's situation, even if one doesn't know about even a single instance of this co-occurrence, and doesn't know that Helen Keller did give herself that name, the correct answer can be easily provided. It's enough to know that some of Helen Keller's senses were inoperable, that she is famous, and that she is famous for reporting her internal states during the period of this inoperability. With this knowledge, and the vast background knowledge that supports the ability to understand the propositional or semantic content of the clue sentence (= the question), one can venture the answer, along with an argument for why one believes that the answer is probably correct. We shall say that an answer in this mode is an 'answer$_2$,' and we shall say that an answer in this mode, accompanied by a justification, is an 'answer$_2^j$.'

Now, Stephen Wolfram, in Thompson's (2010) *New York Times* article, explicitly claims that Watson doesn't answer questions. Since we can assume that Wolfram is of sound mind, he must have in mind a sense of 'answer' that departs a bit from the usual one. After all, even in these early phases of the project, still quite a while before the actual competition on *Jeopardy!*, as amply reported by Thompson (2010), there can be no denying that *in some sense* Watson already answers questions, often correctly. This observable sense of question answering, as far as we can tell, corresponds to providing an answer$_1$, while Wolfram's sense of answer coincides with providing an answer$_2$/answer$_2^j$.

But we can do a bit better than this in moving toward an understanding of Wolfram's skepticism. Note that he specifically says: "Not to take anything away from this 'Jeopardy!' thing, but I don't think Watson really is answering questions—it's not like the 'Star Trek' computer.". We can understand the complaint here to be one based not on Star Trek, but upon *2001*; accordingly, Wolfram's point then becomes the claim that while Watson can answer$_1$ questions, it can't do what HAL can do, that is, both answer$_2$ and answer$_2^j$ questions. This claim appears to be true.

Can Watson be augmented so as to reach into the HALish realm of answering$_2$/ answering$_2^j$ questions? We take this up briefly in the final section, to which we now turn.

6 Concluding Remarks

We have explained that if HAL is a liar, building an AI like him becomes all the more challenging; but as we've also explained, there is some recent work on mechanical mendacity that provides significant hope. (We have also pointed out that it's far from clear that HAL *is* a liar.) And of course we briefly took note of the fact there is an AI system under construction, Watson, which promises to provide robust QA, something HAL certainly offered to the crew.

Of course, for manned missions to distant planets, NASA will need more than an AI able to lie (or more properly put: able to do things that leverage the considerable mental powers required to be a liar), and, more generally, to answer$_1$ questions. Among other things, NASA will need *bona fide conversational* computers able to provide answers$_2$/answers$_2^j$; HAL, of course, was such a machine (though his answers$_2$ and justifications were of course not guaranteed to be correct!). Are there any recent developments that support optimism about the arrival of an AI with the capacity to converse (or as we might say, 'converse$_2^j$), in the foreseeable future? Of course, as we've already noted, Watson himself may be such a development—*if* it's true that better versions of the system can pass from answering$_1$ questions to answering$_2$/answering$_2^j$ them.

Answers to this question about recent developments will inevitably depend upon one's prior affinity for one or more of the competing research paradigms in the field of AI. Those folks who believe the next half-century of R&D in natural language processing will be dominated by statistical approaches, and that such domination would be wise and productive, will doubtless be quite optimistic. They will confidently report that the turn away from logic is itself a development that augurs well for reaching HAL-level intelligence. A case in point is Eugene Charniak, who proudly opined at the 50-year birthday of AI (held in 2006, where the field, at least in its modern form, began: Dartmouth College) that statistical approaches would for the next five decades be the only game in town—and that this game would pay great dividends toward reaching the likes of HAL. John McCarthy was in attendance at the same birthday conference, replete with books on logic that he was busy studying, for the very purpose of advancing AI. Bringsjord's position is at least partially expressed in the paper that arose from his presentation at the conference in question, and is an endorsement of "weak" AI based firmly on formal logic: (Bringsjord 2008).[10] This position is a direct descendant of earlier versions of; see, for example, (Bringsjord & Ferrucci 1998*a*, Bringsjord & Ferrucci 1998*b*).

Bringsjord, in contrast to the statistics-oriented crowd, is brutally pessimistic. In the long run, as stated above, he is quite sure that sooner or later the TTT and beyond will be passed by an AI; ergo, he is quite sure that sooner or later a machine with HAL-level power will arrive. But the question under consideration refers to the *foreseeable* future. There is simply no evidence or decent argument in support of the proposition that a HAL-level computer can be seen by some up there ahead of the cutting-edge research and development that is driving today's AI. Indeed, there is a reason why such a machine *can't* be seen, and it can be expressed in the form of an argument, to wit:

[10] Weak AI is devoted only to engineering computing machines that *simulate* human-level cognition, while "strong" AI is devoted to building computing machines that outright *replicate* human cognition. In short, while a weak AI system need only *appear* to be conscious, a strong AI system would need to quite literally *be* conscious. This distinction is discussed e.g. in (Bringsjord 2000).

(1) A computer able to converse$_2^j$ like HAL must be engineered on the basis of a logico-mathematical theory $\mathcal{T}^\star$ that covers the deep, formal semantics of natural language.
(2) If for a computer with a certain capacity $\mathcal{C}$ to be engineered, a logico-mathematical theory $\mathcal{T}$ is needed, and $\mathcal{T}$ doesn't exist, and no human person knows how to create $\mathcal{T}$, then it's rational to hold that no such computer will exist in the foreseeable future.
(3) The theory $\mathcal{T}^\star$ does not exist, and no human person knows how to create it.

$\therefore \bar{H}$ It's rational to hold that no computer able to converse$_2^j$ like HAL will arrive in the foreseeable future.

This is obviously a formally valid argument: The conclusion, $\bar{H}$, can be deduced from the three premises easily; we could symbolize the argument, throw it into a theorem prover, and the conclusion would be mechanically derived from the trio. It would seem that (2) and (3) are undeniable.[11] Therefore the argument hinges on the truth-value of premise (1). If this premise is true, the argument is sound, and we have our answer with respect to HAL and the foreseeable future. Is (1) true? We tend to believe so, but must leave the articulation of our rationales to P&C 2010 and time spent upon the Nile, and beyond. Notice that we do not take a firm stand: we say that we *tend* to believe so. We hedge our bets because we suspect that we ourselves might be able to use logic-based techniques to move Watson toward conversing$_2^j$ in HAL-like fashion...

References

Arkoudas, K. & Bringsjord, S. (2009), 'Propositional Attitudes and Causation', *International Journal of Software and Informatics* **3**(1), 47–65.

Bringsjord, S. (1992), *What Robots Can and Can't Be*, Kluwer, Dordrecht, The Netherlands.

Bringsjord, S. (1995), Could, how could we tell if, and why should–androids have inner lives?, *in* K. Ford, C. Glymour & P. Hayes, eds, 'Android Epistemology', MIT Press, Cambridge, MA, pp. 93–122.

Bringsjord, S. (1998), 'Chess is Too Easy', *Technology Review* **101**(2), 23–28.
URL: http://www.mm.rpi.edu/SELPAP/CHESSEASY/chessistooeasy.pdf

Bringsjord, S. (2000), 'Review of John Searle's *The Mystery of Consciousness*', *Minds and Machines* **10**(3), 457–459.

Bringsjord, S. (2008), 'The logicist manifesto: At long last let logic-based AI become a field unto itself', *Journal of Applied Logic* **6**(4), 502–525.
URL: http://kryten.mm.rpi.edu/SB_LAI_Manifesto_091808.pdf

Bringsjord, S. & Ferrucci, D. (1998*a*), 'Logic and artificial intelligence: Divorced, still married, separated...?', *Minds and Machines* **8**, 273–308.

[11] Cognoscenti may resist slightly in light of Montague semantics (nicely covered in Dowty et al. 1981), but no one really thinks Montague did any more than seminally point in the general direction of $\mathcal{T}^\star$. As our lab is an active user of formal theories of natural language, we would like to be able to be able to ourselves provide $\mathcal{T}^\star$, but no such luck—at least at this point.

Bringsjord, S. & Ferrucci, D. (1998*b*), 'Reply to Thayse and Glymour on logic and artificial intelligence', *Minds and Machines* **8**, 313–315.

Bringsjord, S. & Noel, R. (2002), Real robots and the missing thought experiment in the chinese room dialectic, *in* J. Preston & M. Bishop, eds, 'Views into the Chinese Room: New Essays on Searle and Artificial Intelligence', Oxford University Press, Oxford, UK, pp. 144–166.

Carson, T. L. (2006), 'The Definition of Lying', *Noûs* **40**(2), 284–306.

Castelfranchi, C. (2000), 'Artificial liars: Why computers will (necessarily) deceive us and each other', *Ethics and Information Technology* **2**(2), 113–119.

Chisholm, R. M. & Feehan, T. D. (1977), 'The Intent to Deceive', *Journal of Philosophy* **74**(3), 143–159.

Clark, M. (2010), Cognitive Illusions and the Lying Machine: A Blueprint for Sophistic Mendacity, PhD thesis, Department of Cognitive Science, Rensselaer Polytechnic Institute, Troy, NY.

Clark, M. & Bringsjord, S. (2008), Persuasion Technology Through Mechanical Sophistry, *in* J. Masthoff, C. Reed & F. Grasso, eds, 'AISB 2008 Convention on Communication, Interaction and Social Intelligence, 1st–4th April 2008, University of Aberdeen, Scotland. Vol. 3: Proceedings of the AISB 2008 Symposium on Persuasive Technology', Society for the Study of Artificial Intelligence and Simulation of Behaviour, Brighton, England, pp. 51–54.

Clarke, A. C. (1968/1999), *2001: A Space Odyssey*, New American Library, New York, NY.

Dowty, D., Wall, R. & Peters, S. (1981), *Introduction to Montague Semantics*, D. Reidel, Dordrecht, The Netherlands.

Fallis, D. (2009), 'What is Lying?', *Journal of Philosophy* **CVI**(1), 29–56.

Fuchs, N. E., Kaljurand, K. & Kuhn, T. (2008), Attempto Controlled English for Knowledge Representation, *in* C. Baroglio, P. A. Bonatti, J. Małuszyński, M. Marchiori, A. Polleres & S. Schaffert, eds, 'Reasoning Web: 4th International Summer School 2008, Venice, Italy, September 7–11, 2008, Tutorial Lectures', Vol. 5224 of *Lecture Notes in Computer Science*, Springer, pp. 104–124.

Fuchs, N. E., Schwertel, U. & Schwitter, R. (1999), Attempto Controlled English — Not Just Another Logic Specification Language, *in* P. Flener, ed., 'Logic-Based Program Synthesis and Transformation: 8th International Workshop, LOPSTR'98 Manchester, UK, June 15–19, 1998', Vol. 1559 of *Lecture Notes in Computer Science*, Springer, pp. 1–20.

Harnad, S. (1991), 'Other bodies, other minds: A machine incarnation of an old philosophical problem', *Minds and Machines* **1**(1), 43–54.

Johnson-Laird, P. N. (2006), *How We Reason*, Oxford University Press, New York, NY.

Kahneman, D., Slovic, P. & Tversky, A., eds (1982), *Judgement under uncertainty: Heuristics and biases*, Cambridge University Press, New York, NY.

Kant, I. (1930), Ethical Duties towards Others: Truthfulness, *in* 'Lectures on Ethics', Methuen & Co., London, England, pp. 224–235.

Krishna, D. (1961), "Lying' and the Compleat Robot', *British Journal for the Philosophy of Science* **12**(46), 146–149.

Kubrick, S. & Clarke, A. C. (1965), 2001: A Space Odyssey, unpublished screenplay, Hawk Films Ltd., Boreham Wood, England.

Mahon, J. E. (2008), The Definition of Lying and Deception, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', Stanford University, Stanford, CA.
URL: http://plato.stanford.edu/archives/fall2008/entries/lying-definition/

Maybury, M., ed. (2004), *New Directions in Question Answering*, AAAI Press, Menlo Park, CA.

Newell, A. (1973), You can't play 20 questions with nature and win: Projective comments on the papers of this symposium, *in* W. Chase, ed., 'Visual Information Processing', New York: Academic Press, pp. 283–308.

Piattelli-Palmarini, M. (1994), *Inevitable Illusions: How Mistakes of Reason Rule Our Minds*, John Wiley & Sons, New York, NY.

Pohl, R. F., ed. (2004), *Cognitive Illusions: A handbook on fallacies and biases in thinking, judgement and memory*, Psychology Press, New York, NY.

Thompson, C. (2010), 'What is IBM's Watson?', *The New York Times Magazine* pp. 30–37; 44–45.

Toulmin, S. E. (1958), *The Uses of Argument*, Cambridge University Press, Cambridge, MA.

Turing, A. (1950), 'Computing machinery and intelligence', *Mind* **LIX (59)**(236), 433–460.

Williams, B. A. O. (2002), *Truth and Truthfulness: An Essay in Genealogy*, Princeton University Press, Princeton, NJ.

2001: HAL's Legacy

David Stork

Ricoh Innovations

Abstract. Stanley Kubrick and Arthur C. Clarke's 1968 epic film "2001 - A Space Odissey" included of the most compelling and thoroughly researched visions for computer science ever depicted in film, specifically the HAL 9000 computer. This presentation will compare the visions in the film with actual developments in computer science, all in the namesake year. What did the films creators "get right" or "get wrong"? Why? You will never see the film the same way again.

De-quantisation of the Quantum Fourier Transform

Alastair A. Abbott

Department of Computer Science, University of Auckland
Auckland, New Zealand
aabb009@aucklanduni.ac.nz

Abstract. The quantum Fourier transform (QFT) plays an important role in many known quantum algorithms such as Shor's algorithm for prime factorisation. In this paper we show that the QFT algorithm can, on a restricted set of input states, be de-quantised into a classical algorithm which is both more efficient and simpler than the quantum algorithm. By working directly with the algorithm instead of the circuit, we develop a simple classical version of the quantum basis-state algorithm. We formulate conditions for a separable state to remain separable after the QFT is performed, and use these conditions to extend the de-quantised algorithm to work on all such states without loss of efficiency. Our technique highlights the linearity of quantum mechanics as the fundamental feature accounting for the difference between quantum and de-quantised algorithms, and that it is this linearity which makes the QFT such a useful tool in quantum computation.

Keywords: Quantum Computing, Quantum Fourier Transform, De-quantisation

1 Introduction

The *quantum Fourier transform (QFT)* plays an important role in a large number of known algorithms for quantum computers [1]. It plays a central role in Shor's algorithm for prime factorisation [2] and is often thought to be at the heart of many quantum algorithms which are faster than any known classical counterpart. However, following on from recent results relating to classical features of the QFT algorithm [3–6], we will argue that the QFT algorithm itself is classical in nature.

The process of de-quantising quantum algorithms into equivalent classical algorithms is a powerful tool for investigating the nature of quantum algorithms and computation. Few general results are known about when such de-quantisations are possible and the power of quantum computation compared to classical computation. In this paper we show how the QFT algorithm can be de-quantised into a simpler, more efficient, classical algorithm when operating on a range of input states. While the de-quantised algorithms themselves are of interest, they also allow us to gain insight into the nature of the QFT. We

will argue that it is the linearity inherit in the unitary quantum computational model which makes the QFT such a useful tool, rather than the nature of the QFT itself.

In Section 2 of this paper we overview the basic QFT theory and present the QFT algorithm in a compact form which allows us to move away from the restrictions imposed by the circuit layout. In Section 3 we overview the de-quantisation procedure and de-quantise the QFT algorithm acting on a basis-state input. In Section 4 we explore the entangling power of the QFT and determine conditions for when a separable input state remains unentangled by the QFT, before presenting a de-quantised algorithm that works on such product-state inputs. In Section 5 we discuss why de-quantisation of the QFT is possible and note some common misunderstandings about the QFT which contribute to this.

2 Discrete and Quantum Fourier Transforms

The *discrete Fourier transform (DFT)* on which the QFT is based is a transformation on a q-dimensional complex vector $\chi = (f(0), f(1), \ldots, f(q-1))$ into its Fourier representation $\hat{\chi} = (\hat{f}(0), \hat{f}(1), \ldots, \hat{f}(q-1))$ [1]:

$$\hat{f}(c) = \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} e^{2\pi i ac/q} f(a), \tag{1}$$

for $c \in \{0, 1, \ldots, q-1\}$. The QFT is similarly defined so that the transformation acts on a state vector in q-dimensional Hilbert space, $\mathcal{H}_q$. In quantum computation we work with a state vector defining a register comprising of n two-state qubits, so we will only consider the case that $q = 2^n$ from this point onwards. We will use the convention that n is the number of qubits while $N = 2^n$ is the dimension of Hilbert space the n qubits are in. This means that the QFT, denoted F_q, acts on the N amplitudes of a particular n-qubit state, i.e.

$$\sum_{a=0}^{N-1} f(a) \left|a\right\rangle \xrightarrow{F_N} \sum_{c=0}^{N-1} \hat{f}(c) \left|c\right\rangle . \tag{2}$$

The QFT hence transforms a state so as to perform a DFT on its state vector.

As a result of the linearity of quantum mechanics, in order to compute the QFT we only need to design an algorithm to transform a single component of the state vector. This is because an arbitrary state $\left|\psi_N\right\rangle = \sum_{a=0}^{N-1} f(a) \left|a\right\rangle$ transforms as:

$$F_N \left|\psi_N\right\rangle = \sum_{a=0}^{N-1} f(a) F_N \left|a\right\rangle = \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} \sum_{c=0}^{N-1} e^{2\pi i ac/N} f(a) \left|c\right\rangle = \sum_{c=0}^{N-1} \hat{f}(c) \left|c\right\rangle .$$

Hence we arrive at the standard definition of the QFT as the mapping [7]

$$\left|a\right\rangle \xrightarrow{F_N} \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} e^{2\pi i ac/N} \left|c\right\rangle , \tag{3}$$

with $a \in \{0, 1, \ldots, N-1\}$. Keeping in mind that we are dealing with registers composing of qubits, we can decompose a (and similarly c) into its binary representation so that $a = 2^{n-1}a_1 + 2^{n-2}a_2 + \cdots + 2^1 a_{n-1} + 2^0 a_n$ and $|a\rangle = |a_1 a_2 \cdots a_n\rangle$. By denoting $a = a_1 a_2 \cdots a_n$ and $a/2^n = 0.a_1 a_2 \cdots a_n$ we observe that

$$\begin{aligned} e^{2\pi i ac/2^n} &= e^{2\pi i a(2^{n-1}c_1 + 2^{n-2}c_2 + \cdots + 2^0 c_n)/2^n} \\ &= e^{2\pi i (a_1 a_2 \cdots a_n)c_1/2^1} e^{2\pi i (a_1 a_2 \cdots a_n)c_2/2^2} \cdots e^{2\pi i (a_1 a_2 \cdots a_n)c_n/2^n} \\ &= e^{2\pi i (a_1 \cdots a_{n-1}.a_n)c_1} e^{2\pi i (a_1 \cdots a_{n-2}.a_{n-1}a_n)c_2} \cdots e^{2\pi i (0.a_1 a_2 \cdots a_n)c_n}. \end{aligned} \quad (4)$$

Noting that for any decimal $x.y$ we have $e^{2\pi i(x.y)} = (e^{2\pi i})^x e^{2\pi i(0.y)} = e^{2\pi i(0.y)}$, we see that only the fractional part of $(a_1 \cdots a_{n-j}.a_{n-j+1} \cdots a_n)c_j$ is of any significance in the exponent of (4).[1] Hence, we find

$$e^{2\pi i ac/2^n} |c_1 \cdots c_n\rangle = e^{2\pi i (0.a_n)c_1} |c_1\rangle \cdots e^{2\pi i (0.a_1 a_2 \cdots a_n)c_n} |c_n\rangle .$$

Using this decomposition we can write (3) as a product state of individual qubits,

$$\sum_{c=0}^{N-1} e^{2\pi i ac/2^n} |c\rangle = (|0\rangle + e^{2\pi i (0.a_n)} |1\rangle) \cdots (|0\rangle + e^{2\pi i (0.a_1 \cdots a_n)} |1\rangle). \quad (5)$$

The quantum algorithm to implement the QFT follows directly from this factorisation. The circuit for the algorithm is shown in Figure 1. The algorithm can be written explicitly as follows [7]:

Quantum Fourier Transform
Input: The state $|a\rangle = |a_1\rangle |a_2\rangle \cdots |a_n\rangle$.
Output: The transformed state $\frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i (0.a_n)} |1\rangle) \cdots (|0\rangle + e^{2\pi i (0.a_1 \cdots a_n)} |1\rangle)$.

1. For $j = 1$ to n, transform qubit $|a_j\rangle$ as follows:
2. $\quad |a_j\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (0.a_j)} |1\rangle)$.
3. $\quad$ For $k = j + 1$ to n:
4. $\quad\quad \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (0.a_j \cdots a_{k-1})} |1\rangle) \xrightarrow{R_k} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (0.a_j \cdots a_{k-1}a_k)} |1\rangle)$ where R_k is the unitary k-controlled phase shift:

$$R_k = \begin{pmatrix} 1 & 0 & 0 & & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ 0 & 0 & 0 & e^{2\pi i/2^k} & \end{pmatrix}.$$

5. $\quad$ End For.
6. $\quad$ Reverse the order of the qubits.
7. End For.

Clearly this produces the state

[1] This technique of removing factors of $(e^{2\pi i})^k$ for $k \in \mathbb{N}$ will be commonly used throughout this paper to reduce formulae.

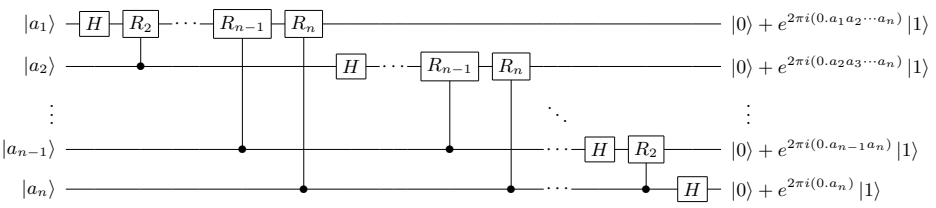


Fig. 1. The standard quantum circuit for the QFT. The output normalisation factors of $1/\sqrt{2}$ and swap gates to reverse qubit order are omitted.

There are a few important notes about the QFT which should be made. While both the DFT and the QFT act on vectors in a complex vector space, the DFT acts on an abstract, mathematical vector, whereas the QFT acts on a physical state which we mathematically represent by a vector in $\mathcal{H}_N$. The subtle difference here is that with the classical DFT, we can read the values of all 2^n Fourier coefficients $\hat{f}(c)$ by simple inspection of the transformed vector. With the QFT, the resulting state (2) embeds all 2^n coefficients as amplitudes for the 2^n states of an n-qubit system. However, the collapse of the superposition upon measurement means that it is impossible to measure the amplitudes of a quantum state without an ensemble of such states to make a statistical approximation of the amplitudes from [8], and detecting phase differences between states is even more difficult. Hence, the quantum state (2) contains all the information of the classically transformed vector, but it is inaccessible to measurement. The main use of the QFT is then as a tool to extract information embedded in the relative amplitudes of states as opposed to determining the coefficients themselves.

Another result of this is that the efficiency of the QFT ($O(n^2)$ as opposed to the DFT which is $O(n2^n)$) is in some sense due to the ability to perform the transformation and utilise the information in the phases without measuring the state. Evidently, any algorithm requiring measurement needs exponential time (there are 2^n coefficients to measure), so even if quantum mechanics would allow us to measure the Fourier coefficients in state (2), doing so would take $O(n2^n)$ time: 2^n coefficients, n qubits each. Making use of this embedded information while avoiding measurement is certainly an important part of the fine art of developing algorithms in quantum computing.

3 Initial De-quantisation Investigation

Having presented the QFT, there are some issues to be brought to light. The decomposition of the transformed state (3) (shown in (5)) is evidently not entangled, and the separability of the state would lead us to believe that the QFT algorithm producing it could be simulated efficiently in a classical manner [9, 10], and there are certainly results towards this.

It was realised shortly after the discovery of Shor's algorithm that the QFT could be computed in a semiclassical manner [5]. By using classical signals resulting from quantum measurements, one can perform the QFT on a state using

classical logic and one-qubit gates (instead of the usual two-qubit controlled-phase-shifts). This method gives the same resulting probability distribution as the quantum algorithm, but destroys the state's superposition as it relies on irreversible measurements. As a result, this is only useful in an algorithm in which the QFT directly precedes measurement. Shor's algorithm happens to be of exactly this nature, but this is only an initial step towards true classical simulation.

Much more recently, classical simulations of the QFT have been studied from the viewpoint of simulating the circuit in Figure 1 by exploiting the bubble-width of the quantum circuit [3] and the tensor contraction model [6]. The bubble-width approach uses a slightly modified version of the QFT circuit which is of logarithmic bubble-width and simulates this circuit. The tensor-contraction model also focuses on the circuit topology, but relies on associating a tensor with each vertex in the circuit, then cleverly contracting the tensors into a single rank-one tensor. Both these methods work on separable input states, but are sampling based forms of de-quantisation [11] in the sense that a final measurement is assumed and an output is classically sampled from the correct (calculated) probability distribution. This makes these de-quantisations less general than might be desired and difficult to apply when the QFT is used, as it often is, as a part of a larger quantum algorithm. This is because in these cases measurement cannot be assumed after the QFT, and the de-quantisation must be cleverly and non-trivially composed with a de-quantisation of the rest of the algorithm to be applied.

Working with the circuit topology, while beneficial for some purposes, also seems to overcomplicate matters and restrict generalisation when it comes to classical simulation. We will explore simulations of the QFT in a different light, more along the lines of the de-quantisation explored previously by Abbott [9] and Calude [12] which aim to provide stronger (not sampling based) de-quantisations when possible.

3.1 De-quantisation Overview

The idea behind this de-quantisation procedure is that qubits which are separable exhibit only superposition and interference. These properties are the result not of non-classical features of the qubits, but rather of the two-dimensionality of the qubits. By using classical, deterministic two-dimensional bits instead of qubits, the same behaviour can be exhibited without the difficulties imposed by measurement and probabilities. Not all algorithms fit within this paradigm, but there are many which can be tackled with this approach. Algorithms which use measurement as a fundamental part of their procedure are examples of those which are not so well suited, and sampling-based techniques are more suitable in these situations. Finding when these stronger de-quantisations are possible also gives insight into the power of particular quantum algorithms [11], as this reflects to some degree the amount the algorithm utilises the possible advantages of quantum mechanics. In cases where entanglement is bounded [10], we can use this de-quantisation procedure to produce classical algorithms which are

as efficient as their quantum counterparts. This procedure was explicitly examined further [9, 12] when applied to the Deutsch-Jozsa problem [13, 14], where complex numbers were used as classical two-dimensional bits. In this paper we will apply this de-quantisation procedure to the QFT, but because the amplitudes we need to represent in the QFT algorithm are complex-valued, we cannot use complex numbers as our two-dimensional bits. There is no problem though with simply using two-valued vectors as our classical bits, so we will employ this procedure.

3.2 Basis-state De-quantisation

The de-quantisation for a basis-state needs only to simulate the transformation defined in (3). As a result of the decomposition in (5), the effect of the QFT on the jth qubit is easily seen to be

$$|a_j\rangle \xrightarrow{F_{2^n}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_{n-j+1}\cdots a_n)}\,|1\rangle). \tag{6}$$

The difficulty in implementing this in a quantum computer is that the phase of a qubit needs to be altered depending on the values of the other qubits without altering them – that is why it is not helpful to express the quantum algorithm as we have done in (6) – and the circuit of controlled-phase-shifts is required to implement this. The information is spread over the input qubits and must be obtained without measurement. In the classical case there are no such restrictions on measurement, so de-quantisation should only require directly implementing (6). However, evaluating the complex phase for each of the n qubits takes $O(n)$ time, leading to a $O(n^2)$ procedure. This can be reduced to $O(n)$ by calculating each phase dependent on the previous one. To do so, let ω_j be the jth phase factor and note the following:

$$\begin{aligned}\omega_j &= e^{2\pi i(0.a_{n-j+1}\cdots a_n)} \\ &= e^{2\pi i(0.a_{n-j+1})}e^{2\pi i(0.a_{n-j+2}\cdots a_n)/2} \\ &= (-1)^{a_{n-j+1}}\sqrt{\omega_{j-1}},\end{aligned}$$

and

$$\omega_1 = e^{2\pi i(0.a_n)} = (-1)^{a_n},$$

where by the square-root we mean the principal root. The square-root of a complex number such as ω_j can be calculated independently of n. Specifically, if we have $s + ti = \sqrt{b + di}$ with the further requirement that for a root of unity $\sqrt{b^2 + d^2} = 1$, then [15]:

$$s = \frac{1}{\sqrt{2}}\sqrt{1+b}, \quad t = \frac{\mathrm{sgn}(d)}{\sqrt{2}}\sqrt{1-b},$$

where $\mathrm{sgn}(d) = d/|d|$ is the sign of d. The efficient de-quantised algorithm is then the following:

Basis-state De-quantised QFT
Input: The binary string $a = a_1 a_2 \dots a_n$.
Output: The n transformed two-component complex vectors $\boldsymbol{b_1 b_2 \dots b_n}$.

1. Let $\omega = 1$
2. For $j = 1$ to n:
3. Set $\omega = (-1)^{a_{n-j+1}} \sqrt{\omega}$
4. Set $\boldsymbol{b_j} = \frac{1}{\sqrt{2}} \times \begin{pmatrix} 1 \\ \omega \end{pmatrix}$
5. End For

This algorithm produces vectors mathematically identical to the state-vectors in (3) and (5) produced by the QFT, but is computed classically in $O(n)$ time – more efficient than the quantum solution and simpler too. This is primarily because the quantum circuit is constructed subject to the requirement of computing the QFT without any intermediate measurements. As a result, the quantum algorithm corresponding to the circuit must conform to this too, making it more complex than an equivalent classical algorithm need be.

A classical algorithm has the further advantage over the quantum algorithm acting on a basis-state that measurement of the resulting state can be performed at will, and any required information is easily accessible. In the quantum algorithm only a single state can be measured, and no information about the amplitudes (and thus the Fourier coefficients) can be determined from a single QFT application. While this classical algorithm is no faster than the well known fast Fourier transform (FFT) for calculating all the coefficients, it may be advantageous if only some coefficients are required.

The ability to de-quantise the QFT acting on a basis state is not particularly surprising. This is equivalent to the classical DFT acting on a vector with only one non-zero component, producing a fairly trivial and easily computed output. However, this highlights a little more deeply some common misconceptions about the QFT. Because of the linear, unitary evolution of quantum mechanics, the action of the QFT on a basis state shown in (3) is often taken as the definition of the QFT. While this suffices as the definition for the purposes of the quantum algorithm, it is important not to forget that the actual definition of the QFT is that given in (2). When considering classical simulations of the QFT this is even more important, as the action of the QFT on a basis state and the corresponding circuit no longer immediately allow us to compute the complete QFT; indeed it would take 2^n iterations of a classical algorithm simulating the basis state behaviour to compute the complete QFT.

4 Product-state De-quantisation

Here we consider the possibility of extending the de-quantisation to work on a wider range of input states, resulting in a less trivial de-quantisation. If the input state is entangled then it is clear that the de-quantisation is not easily extended, as the method used for the basis-state algorithm relied on the separability of the

input. In such a situation, any de-quantisation attempt would need to involve a different method and work directly from the QFT definition, (2).

It is not immediately clear that the basis-state de-quantisation, which is based on (3), could not be extended to work on arbitrary separable input states. This idea is strengthened by the fact that we used the single-qubit formula (6) to perform the basis-state de-quantisation. However, this implicitly relies on the other qubits in the input state having a definite value, but in the general separable input case this is not necessarily the case. Indeed, the QFT is readily seen to entangle separable input states, e.g:

$$|\phi\rangle = \frac{1}{\sqrt{2}}|0\rangle\left(|0\rangle + |1\rangle\right) \xrightarrow{F_4} \frac{1}{\sqrt{2}}\left(|00\rangle + \frac{1+i}{2}|01\rangle + \frac{1-i}{2}|11\rangle\right).$$

A de-quantisation for arbitrary separable input states is thus not possible in the same way as it was for basis states. However, we will investigate the entangling power of the QFT in order to determine the set of states which are not entangled by the QFT, and present a de-quantised algorithm which works for such states.

4.1 General Separability Conditions

As in the entanglement investigation of the Deutsch-Jozsa problem [9], we will make use of the separability conditions for a qubit state presented in [16], although unlike the Deutsch-Jozsa problem our situation permits the possibility of states with zero-valued amplitudes, complicating the conditions somewhat. The key definitions and theorems we require to determine the separability of a state will be briefly presented, while [16] should be consulted for proofs and discussion.

Definition 1. *The* amplitude abstraction function $\mathcal{A} : \mathcal{H}_N \to \{0,1\}^N$ *is a function which, when applied to a state* $|\psi_N\rangle = \sum_{i=0}^{N-1} c_i |i\rangle$*, yields a bit string* $x = x_0 x_1 \ldots x_{N-1}$ *such that for* $0 \leq i \leq N-1$*,* $x_i = 0$ *if* $c_i = 0$ *and* $x_i = 1$ *otherwise.*

Definition 2. *The set* $\mathcal{B}_N \subset \{0,1\}^N$ *of* well-formed bit strings *of length* $N = 2^n$ *is defined recursively as*

$$\mathcal{B}_2 = \{01, 10, 11\}, \quad \mathcal{B}_{2N} = \{0^N x, x0^N, xx \mid x \in \mathcal{B}_N\}.$$

Definition 3. *The set of* well-formed states *is the set*

$$\mathcal{V}_N = \{|\psi_N\rangle \in \mathcal{H}_N \mid \mathcal{A}(|\psi_N\rangle) \in \mathcal{B}_N\}.$$

Intuitively, a state is well-formed if the zero-valued amplitudes are distributed such that it is a candidate to be separable; if a state is not well-formed it is guaranteed to be entangled. In order to determine if a well-formed state is separable, we require two further definitions.

Definition 4. *For each set of well-formed states $\mathcal{V}_N$, there exists a family of* zero deletion functions $\{\mathcal{D}_K : \mathcal{V}_N \to \mathcal{H}_K \mid K = 2^k, 1 \leq k \leq n\}$, *such that for a well-formed state* $|\psi_N\rangle = \sum_{i=0}^{N-1} c_i |i\rangle \in \mathcal{V}_N$, $\mathcal{D}_K(|\psi_N\rangle) = |\psi'_K\rangle = \sum_{j=0}^{K-1} c'_j |j\rangle$, $\mathcal{A}(|\psi'_K\rangle) = 1^K$, *and* c'_j *is the* j*th non-zero amplitude of* $|\psi_N\rangle$.

Definition 5. *A state* $|\psi_N\rangle = \sum_{i=0}^{N-1} c_i |i\rangle$ *is* pair product invariant *if and only if for all* $j \in \{2, \ldots, n\}$ *and all* $i \in \{0, \ldots, J/2 - 1\}$ $c_i c_{J-i-1} = d_j$, *where each* d_j *is a constant and* $J = 2^j$.

As a concrete example to help understand pair product invariance, consider the cases of $n = 2$ and $n = 3$. For $n = 2$, $|\psi_4\rangle = \sum_{i=0}^{3} c_i |i\rangle$ is pair product invariant if the well known condition $c_0 c_3 = c_1 c_2$ holds. For $n = 3$, $|\psi_8\rangle = \sum_{i=0}^{7} c_i |i\rangle$, we require this same condition, $c_0 c_3 = c_1 c_2$, as well as the further condition that $c_0 c_7 = c_1 c_6 = c_2 c_5 = c_3 c_4$, to hold.

The following theorem from [16] can be used to determine if an arbitrary n-qubit state is separable or not by checking the non-zero amplitudes of the state vector are pair product invariant.

Theorem 1. *Let* $|\psi_N\rangle$ *be an* n*-qubit state for which the bit string* $\mathcal{A}(|\psi_N\rangle)$ *contains* K *ones. Then* $|\psi_N\rangle$ *is separable if and only if* $|\psi_N\rangle \in \mathcal{V}_N$ *and* $\mathcal{D}_K(|\psi_N\rangle)$ *is pair product invariant.*

4.2 QFT Separability Conditions

We wish to consider the case that a separable n-qubit input state remains separable after the QFT has been applied to it. In order to do so, first let us consider the action of the QFT on the separable input state

$$|\psi_N\rangle = \begin{pmatrix}\alpha_1\\ \beta_1\end{pmatrix} \otimes \begin{pmatrix}\alpha_2\\ \beta_2\end{pmatrix} \otimes \cdots \otimes \begin{pmatrix}\alpha_n\\ \beta_n\end{pmatrix} = (f(0), f(1), \ldots, f(N-1))^{\mathrm{T}}.$$

Note that each $f(c)$ can be written as a product of amplitudes as $f(c) = a_1 a_2 \ldots a_n$, where each $a_i \in \{\alpha_i, \beta_i\}$. We will use the notation $f_j(c)$ to mean $a_j a_{j+1} \ldots a_n$, and thus $f(c) = f_1(c) = a_1 f_2(c)$ etc. Because of the structure of the tensor product, for $0 < j < n$ and $c < 2^{n-j}$, $f_j(c) = \alpha_j f_{j+1}(c)$ and $f_j(2^{n-j} + c) = \beta_j f_{j+1}(c)$. The amplitudes of the transformed state $|\hat{\psi}_N\rangle = (\hat{f}(0), \hat{f}(1), \ldots, \hat{f}(N-1))^{\mathrm{T}}$ are given by (1), which can, for a separable input,

be rewritten in the more useful form

$$\begin{aligned}
\hat{f}(c) &= \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} e^{2\pi i ac/N} f_1(a) \\
&= \frac{1}{\sqrt{N}} \alpha_1 \sum_{a=0}^{N/2-1} e^{2\pi i ac/N} f_2(a) + \beta_1 \sum_{a=0}^{N/2-1} e^{2\pi i (N/2+a)c/N} f_2(a) \\
&= \frac{1}{\sqrt{N}} (\alpha_1 + e^{\pi i c} \beta_1) \sum_{a=0}^{N/2-1} e^{2\pi i ac/N} f_2(a) \\
&= \frac{1}{\sqrt{N}} (\alpha_1 + e^{\pi i c} \beta_1)(\alpha_2 + e^{\pi i c/2} \beta_2) \cdots (\alpha_n + e^{\pi i c/2^{n-1}} \beta_n) \\
&= \frac{1}{\sqrt{N}} \prod_{j=1}^{n} (\alpha_j + e^{\pi i c/2^{j-1}} \beta_j). \qquad (7)
\end{aligned}$$

This factorised form of the transformed Fourier coefficients allows us to determine conditions for when the transformed state is well-formed by giving restrictions on the distribution of zeros amongst the amplitudes, a result of the fact that $\hat{f}(c) = 0$ if and only if one of the factors in (7) is zero, and this is a significant step towards determining if a state is separable, and thus de-quantisable.

Lemma 1. *Let $|\psi_N\rangle$ be a separable input state and $|\hat{\psi}_N\rangle = F_N |\psi_N\rangle$ be the transformed state. Then the following three conditions are equivalent:*

(i) $|\hat{\psi}_N\rangle \in \mathcal{V}_N$*, i.e. the transformed state is well-formed.*

(ii) There exists a $k \leq n$ *such that the set*

$$\mathcal{C}_j = \left\{ c \mid \forall l \leq j \left(\alpha_l + e^{\pi i c/2^{l-1}} \beta_l = 0 \iff l = j \right) \right\}$$

is non-empty for all $1 \leq j \leq k$ *and empty for* $k < j \leq n$*.*

(iii) $(\exists 0 \leq k \leq n)(\exists a_1 \ldots a_k \in \{0,1\}^k) \left(\forall 1 \leq j \leq k \left[\alpha_j = e^{\pi i \sum_{p=1}^{j} a_p/2^{j-p}} \beta_j \right] \right.$
$\left. \wedge (\forall a_{k+1} \ldots a_n \in \{0,1\}^{n-k})(\forall n \geq j > k) \left[\alpha_j \neq e^{\pi i \sum_{p=1}^{j} a_p/2^{j-p}} \beta_j \right] \right).$

Proof. (i) $\implies$ (ii): For any $x \in \mathcal{B}_N$, Definition 2 ensures that the number of ones in x, $\#_1(x) = 2^m$ for some $m \leq n$, and hence the number of zeros, $\#_0(x) = 2^n - 2^m = \sum_{l=1}^{n-m} 2^{n-l}$. If $|\mathcal{C}_j| \neq 0$ then there exists a $c' \in \mathcal{C}_j$ such that $c' < 2^j$ and $\hat{f}(c') = 0$. But we must also have $\hat{f}(m2^j + c') = 0$ for $0 \leq m \leq 2^{n-j} - 1$ and hence $|\mathcal{C}_j| = 2^{n-j}$. Also note that each $\mathcal{C}_j$ is disjoint by construction, and $\hat{f}(c) = 0 \implies c \in \mathcal{C}_j$ for some j. For a well-formed state we thus require that for some m,

$$\#_0 \left(\mathcal{A}(|\hat{\psi}_N\rangle) \right) = \sum_{l=1}^{n-m} 2^{n-l} = \sum_{j=1}^{n} |\mathcal{C}_j| = \sum_{j:|\mathcal{C}_j| \neq 0} 2^{n-j},$$

which is satisfied if and only if $\mathcal{C}_1 \dots \mathcal{C}_k$ are non-empty and $\mathcal{C}_{k+1} \dots \mathcal{C}_n$ are empty, with $k = n - m$.

(ii) $\implies$ (i): In the first $K = 2^k$ amplitudes, $2^{k-n} \sum_{j \le k} |\mathcal{C}_j| = \sum_{j=1}^{k} 2^{k-j} = K - 1$ of them are zero. Let $\hat{f}(c')$ be the single one of these non-zero amplitudes. Then, by symmetry, $\hat{f}(dK + c') \neq 0$ for $0 \le d \le 2^{n-k} - 1$. Thus, $\mathcal{A}(|\hat{\psi}_N\rangle) = x^{2^{n-k}}$, where $x \in \{0,1\}^K$ and $\#_1(x) = 1$. Any such x is clearly well-formed, and thus the state $|\hat{\psi}_N\rangle$ is also well-formed.

(ii) $\iff$ (iii): Note that $\sum_{p=1}^{j} a_p / 2^{j-p} = \frac{1}{2^{j-1}} \sum_{p=1}^{j} a_p 2^{p-1}$, and we will proceed by induction for $j \le k$. Since $\alpha_1 = e^{\pi i a_1} \beta_1 \iff \alpha_1 + e^{\pi i (1+a_1)} \beta_1 = 0$, such an $a_1 \in \{0,1\}$ exists if and only if $|\mathcal{C}_1| \neq 0$. Now, assume that for all $1 \le l < j \le k$, $\alpha_l = e^{\frac{\pi i}{2^{l-1}} \sum_{p=1}^{l} a_p 2^{p-1}} \beta_l$ and $|\mathcal{C}_l| \neq 0$. Then

$$\alpha_j = e^{\frac{\pi i}{2^{j-1}} \sum_{p=1}^{j} a_p 2^{p-1}} \beta_j \iff \alpha_j + e^{\frac{\pi i}{2^{j-1}} (2^{j-1} + \sum_{p=1}^{j} a_p 2^{p-1})} \beta_j = 0,$$

so such a bit string $a_1 \dots a_j$ exists if and only if there is a c such that $\alpha_j + e^{\pi i c / 2^{j-1}} \beta_j = 0$ (in fact $c = (2^{j-1} + \sum_{p=1}^{j} a_p 2^{p-1}) \bmod 2^j$). Further, the inductive hypothesis ensures that for all $l < j$,

$$\begin{aligned}
\alpha_l + e^{\pi i c / 2^{l-1}} \beta_l &= \alpha_l + e^{\frac{\pi i}{2^{l-1}} (2^{j-1} + \sum_{p=1}^{j} a_p 2^{p-1})} \beta_l \\
&= \alpha_l + e^{\pi i \frac{2^{j-1}}{2^{l-1}}} e^{\frac{\pi i}{2^{l-1}} (\sum_{p=1}^{l} a_p 2^{p-1})} \beta_l \\
&= \alpha_l + e^{\frac{\pi i}{2^{l-1}} (\sum_{p=1}^{l} a_p 2^{p-1})} \beta_l \\
&\neq \alpha_l - e^{\frac{\pi i}{2^{l-1}} (\sum_{p=1}^{l} a_p 2^{p-1})} \beta_l \\
&= 0,
\end{aligned}$$

thus such a bit string $a_1 \dots a_j$ exists if and only if $|\mathcal{C}_j| \neq 0$. Hence, $\mathcal{C}_j$ is non-empty for $j \le k$ if and only if $\exists a_1 \dots a_k \forall 1 \le j \le k (\alpha_j = e^{\pi i \sum_{p=1}^{j} a_p / 2^{j-p}} \beta_j)$. The condition that for $j > k$ and all $a_{k+1} \dots a_j \in \{0,1\}^{j-k}$ $\alpha_j \neq e^{\pi i \sum_{p=1}^{j} a_p / 2^{j-p}} \beta_j$ is equivalent to $|\mathcal{C}_j| = 0$, since $|\mathcal{C}_j| = 0$ requires that there exists a c such that $\alpha_j + e^{\pi i c / 2^{j-1}} \beta_j = 0$ and $\alpha_k + e^{\pi i c / 2^{k-1}} \beta_k \neq 0$. The only $c < 2^k$ which satisfies this is $c = \sum_{p=1}^{k} a_p 2^{p-1}$, so by symmetry any c which satisfies this must be able to be written as $c = \sum_{p=1}^{j} a_p 2^{p-1}$ for some $a_{k+1} \dots a_j$. Hence we see that ((ii)) and ((iii)) are equivalent. □

Condition ((ii)) of Lemma 1 corresponds to a more intuitive requirement for the transformed state to be well-formed. It says that for each $j \ge 1$ there must be a value of c such that $\hat{f}(c) = 0$ with the jth term in (7) equal to zero and the first $j - 1$ terms non-zero. If for some k there is no c satisfying this condition, then there must not be any c satisfying them for $j > k$ either, or the state will not be well-formed. Condition ((iii)) translates this notion into formal requirements about the relationships between the components of the untransformed input state components α_i, β_i which will ensure the transformed state will satisfy condition ((ii)) and thus be well-formed.

Lemma 1 gives us conditions for when the first condition of Theorem 1 is satisfied and it remains to determine which separable input states also satisfy the condition that $\mathcal{D}_Z(|\hat{\psi}_N\rangle)$, with $Z = \sum_{l=1}^{k} 2^{n-l}$, is pair product invariant. The amplitudes which are deleted by the function $\mathcal{D}_Z$ are the Z values of c which are in $\mathcal{C}_j$ for some j.

Lemma 2. *Let $|\psi_N\rangle$ be a separable input state for which the transformed state $|\hat{\psi}_N\rangle$ is well-formed, i.e. $|\psi_N\rangle$ satisfies the conditions of Lemma 1. Let k be as in Lemma 1 part ((iii)) and $Z = \sum_{l=1}^{k} 2^{n-l}$. Then $\mathcal{D}_Z(|\hat{\psi}_N\rangle)$ is pair product invariant if and only if for all $j > k+1$, $\alpha_j\beta_j = 0$, i.e. the $(k+1)$th qubit can be in an arbitrary superposition, and qubits $k+2$ to n must not be in a superposition, although arbitrary phase is permitted.*

Proof. Let c' be the smallest c such that $\hat{f}(c) \neq 0$, and let $n' = n-k$, $N' = 2^{n'}, K = 2^k$. By symmetry, the N' non-zero amplitudes are $\hat{f}(dK + c')$ for $0 \leq d \leq N'-1$. The zero-deleted state is thus $\mathcal{D}_Z(|\hat{\psi}_N\rangle) = (\hat{f}'(0), \ldots, \hat{f}'(N'-1))$, where $\hat{f}'(d) = \hat{f}(dK + c')$. By breaking up the sum in (7) we see that each of these amplitudes is of the form:

$$\begin{aligned}\hat{f}'(d) &= \frac{1}{\sqrt{N}} \left[\prod_{l=1}^{k} (\alpha_l + e^{\pi i (dK+c')/2^{l-1}} \beta_l)\right] \left[\prod_{l=1}^{n'} (\alpha_{k+l} + e^{\pi i (d+c'/K)/2^{l-1}} \beta_{k+l})\right] \\ &= \Gamma \prod_{l=1}^{n'} (\alpha_{k+l} + e^{2\pi i (d+\delta)/L} \beta_{k+l}), \end{aligned} \tag{8}$$

where $L = 2^l$, $\delta = c'/K$ is independent of d, as also is $\Gamma = \frac{1}{\sqrt{N}} \prod_{l=1}^{k} (\alpha_l + e^{(2\pi i)^{dK/L}} e^{2\pi i c'/L} \beta_l) \neq 0$ (recall $k \geq l$ so dK/L is a positive integer). For all $1 < j \leq n', 0 \leq m_1 < m_2 < J/2$, pair product invariance (recall Definition 5) requires that both $\hat{f}'(m_1)\hat{f}'(J - m_1 - 1) = \hat{f}'(m_2)\hat{f}'(J - m_2 - 1)$ and $\hat{f}'(m_1)\hat{f}'(J/2 - m_1 - 1) = \hat{f}'(m_2)\hat{f}'(J/2 - m_2 - 1)$. Since each $\hat{f}'(d) \neq 0$, we require

$$\hat{f}'(J - m_2 - 1)\hat{f}'(J/2 - m_1 - 1) = \hat{f}'(J - m_1 - 1)\hat{f}'(J/2 - m_2 - 1). \tag{9}$$

Symmetry means the left- and right-hand sides both contain common factors of Γ^2, as well as $j-1$ factors from the product (8) for each transformed amplitude, due to the fact that $e^{2\pi i J/L} = e^{\pi i J/L}$ for $l < j$. Thus the condition (9) simplifies to

$$\begin{aligned}&\prod_{l=j}^{n'} (\alpha_{k+l} + e^{2\pi i (J - m_2 - 1 + \delta)/L} \beta_{k+l})(\alpha_{k+l} + e^{2\pi i (J/2 - m_1 - 1 + \delta)/L} \beta_{k+l}) \\ &= \prod_{l=j}^{n'} (\alpha_{k+l} + e^{2\pi i (J - m_1 - 1 + \delta)/L} \beta_{k+l})(\alpha_{k+l} + e^{2\pi i (J/2 - m_2 - 1 + \delta)/L} \beta_{k+l}), \end{aligned} \tag{10}$$

which holds for all j, m_1, m_2 if and only if $\mathcal{D}_Z(|\hat{\psi}_N\rangle)$ is pair product invariant.

We now show by induction that (10) is satisfied if and only if for all $1 < j \leq n'$, $\alpha_{k+j}\beta_{k+j} = 0$. Firstly, consider the case that $j = n'$. The products in (10) each contain only one factor, and expanding leaves only the cross-terms, and the condition simplifies to

$$\alpha_n\beta_n(e^{2\pi i(N'-m_2)/N'} + e^{2\pi i(N'/2-m_1)/N'}) = \alpha_n\beta_n(e^{2\pi i(N'-m_1)/N'} + e^{2\pi i(N'/2-m_2)/N'}). \tag{11}$$

Since this must hold for all distinct m_1, m_2 only the trivial solution is possible, hence $\alpha_n\beta_n = 0$.

Now, assume that $\alpha_{k+l}\beta_{k+l} = 0$ for $l = n', \ldots, j+1$, $j > 1$, and consider $\alpha_{k+j}, \beta_{k+j}$. The products in (10) run from j to n', but all factors for $l > j$ cancel when the pairs on each side are expanded since, by the inductive hypothesis, $\alpha_{k+l}\beta_{k+l} = 0$ for these terms. The condition then reduces to a single factor and we find $\alpha_{k+j}\beta_{k+j} = 0$ exactly as in (11).

Hence, the transformed state is pair product invariant if and only if for all $1 < j \leq n'$ we have $\alpha_{k+j}\beta_{k+j} = 0$. □

Theorem 2. *Given a separable input state $|\psi_N\rangle$, the transformed state $|\hat{\psi}_N\rangle$ is separable if and only if*

$$(\exists 0 \leq k \leq n)(\exists a_1 \ldots a_k \in \{0,1\}^k) \left(\forall 1 \leq j \leq k \left[\alpha_j = e^{\pi i \sum_{l=1}^{j} a_l/2^{j-l}}\beta_j\right] \right.$$
$$\left. \wedge \left(\alpha_{k+1} \neq \pm e^{\pi i \sum_{l=1}^{k} a_l/2^{k-l+1}}\beta_{k+1}\right) \wedge (\forall n \geq j > k+1)\,[\alpha_j\beta_j = 0]\right).$$

Proof. The proof follows directly from Lemmata 1 and 2. □

Theorem 2 allows us to determine if a given separable state $|\psi_N\rangle$ will be entangled or not by the QFT. While the set of such states which are not entangled by the QFT is infinite, the conditions are still highly restrictive, and there is only one qubit that can ever truly be in an arbitrary superposition. However, the conditions between each α_i and β_i are relative, so separability of the transformed state is invariant under phase rotations of individual qubits. These conditions, while restrictive, could be of value in developing new algorithms which make use of the QFT and give a strong insight into the entangling power of the QFT.

4.3 Product-state De-quantisation

For the set of states which are not entangled by the QFT, we can use the conditions of Theorem 2 to extend the basis-state de-quantisation. Let k be as in Theorem 2. Let $r = \sum_{j=2,\alpha_{k+j}=0}^{n-k} 2^{-(k+j)}$ and $\omega = e^{2\pi i r}$ be the coefficient of $(\alpha_{k+2} + \beta_{k+2}) \cdots (\alpha_n + \beta_n)$ in $\hat{f}(1)$. The de-quantised algorithm for states which are not entangled by the QFT is the following ($\boldsymbol{b}[x]$ is the xth component of $\boldsymbol{b}$ starting from 0):

Separable De-quantised QFT
Input: The n two-component complex vectors $\boldsymbol{b_1 b_2 \ldots b_n}$.
Output: The n transformed vectors $\hat{\boldsymbol{b}}_\mathbf{1}\hat{\boldsymbol{b}}_\mathbf{2}\ldots\hat{\boldsymbol{b}}_\boldsymbol{n}$.

1. Calculate k, $a_1 \dots a_k$ as in Theorem 2
2. Calculate r, ω
3. For $j = 1$ to $k+1$:
4. Set $\hat{\boldsymbol{b}}_{\boldsymbol{n-j+1}} = \frac{1}{\sqrt{2}} \times \begin{pmatrix} \alpha_j + e^{\pi i \sum_{l=1}^{j-1} a_l/2^{j-l}} \beta_j \\ \alpha_j - e^{\pi i \sum_{l=1}^{j-1} a_l/2^{j-l}} \beta_j \end{pmatrix}$
5. End For
6. For $j = 1$ to $n-k-1$:
7. Let $l = n - j + 1$
8. Set $\hat{\boldsymbol{b}}_{\boldsymbol{j}} = \frac{1}{\sqrt{2}} \times \begin{pmatrix} \alpha_l + \beta_l \\ \alpha_l + \beta_l \end{pmatrix}$
9. End For
10. For $j = 1$ to n:
11. Set $\hat{\boldsymbol{b}}_{\boldsymbol{n-j+1}}[1] = \omega \hat{\boldsymbol{b}}_{\boldsymbol{n-j+1}}[1]$
12. Set $\omega = \omega^2$
13. End For

Theorem 3. *The* Separable De-quantised QFT *algorithm correctly computes the transformed n-qubit state* $|\hat{\psi}_N\rangle = F_N |\psi_N\rangle$*, where* $|\psi_N\rangle$ *is separable and the* c*th component of* $|\hat{\psi}_N\rangle$ *is described by* (7)*, and does so in* $O(n)$ *time.*

Proof. The values of k and $a_1 \dots a_k$ can be found readily in $O(n)$ time by just checking each pair α_j, β_j too see which option, $a_j = 0, 1$, makes the first condition of Theorem 2 true, and setting a_j accordingly, until neither is true, at which point k is found. Also, r and ω are efficiently found by direct calculation. It remains to verify that the algorithm correctly produces the state

$$\begin{aligned} \hat{f}(c) =& \frac{1}{\sqrt{N}} \prod_{j=1}^{n} (\alpha_j + e^{\pi i c/2^{j-1}} \beta_j) \\ =& \frac{1}{\sqrt{N}} \left[\prod_{j=1}^{k+1} (\alpha_j + e^{\pi i c/2^{j-1}} \beta_j) \right] \left[\prod_{j=k+2}^{n} (\alpha_j + e^{\pi i c/2^{j-1}} \beta_j) \right]. \end{aligned}$$

The algorithm calculates the amplitudes for each qubit, so if we let the n-bit binary expansion of c be $c_n \dots c_1$ we have

$$\begin{aligned} \hat{f}(c) &= \hat{\boldsymbol{b}}_{\boldsymbol{1}}[c_n] \cdot \hat{\boldsymbol{b}}_{\boldsymbol{2}}[c_{n-1}] \cdots \hat{\boldsymbol{b}}_{\boldsymbol{n}}[c_1] \\ &= \frac{\omega^c}{\sqrt{N}} \left[\prod_{j=1}^{k+1} (\alpha_j + (-1)^{c_j} e^{\pi i \sum_{l=1}^{j-1} a_l/2^{j-l}} \beta_j) \right] \left[\prod_{j=k+2}^{n} (\alpha_j + \beta_j) \right] \\ &= \frac{\omega^c}{\sqrt{N}} \left[\prod_{j=1}^{k+1} (\alpha_j + e^{\frac{\pi i}{2^{j-1}} (c_j 2^{j-1} + \sum_{l=1}^{j-1} a_l 2^{l-1})} \beta_j) \right] \left[\prod_{j=k+2}^{n} (\alpha_j + \beta_j) \right]. \quad (12) \end{aligned}$$

Note that, since $\alpha_j = 0$ or $\beta_j = 0$,

$$\prod_{j=k+2}^{n} (\alpha_j + e^{\pi i c/2^{j-1}} \beta_j) = e^{2\pi i c r} \prod_{j=k+2}^{n} (\alpha_j + \beta_j) = \omega^c \prod_{j=k+2}^{n} (\alpha_j + \beta_j),$$

so our algorithm produces this term correctly.

Since the output state is separable, the conditions of Theorem 2 must be satisfied and only one out of the first K amplitudes is non-zero. This amplitude is the one with $c' = \sum_{l=1}^{k} a_l 2^{l-1}$, and by symmetry all the other non-zero amplitudes occur at $c = c' + d2^{n-k}$ for $0 \leq d \leq K-1$. To verify this, note that for all $j \leq k$, we have

$$\alpha_j + e^{\frac{\pi i}{2^{j-1}} \sum_{l=1}^{k} a_l 2^{l-1}} \beta_j = \alpha_j + e^{\frac{\pi i}{2^{j-1}} \sum_{l=1}^{j} a_l 2^{l-1}} \beta_j \neq 0,$$

and hence $\hat{f}(c') \neq 0$. From (12) it is clear that $\hat{f}(c)$ is calculated correctly for these values of c. For all other values of c which have $c_1 \ldots c_n \neq a_1 \ldots a_n$, let m be the smallest $i \leq n$ such that $c_i \neq a_i$. Then we have

$$\alpha_m + e^{\frac{\pi i}{2^{m-1}} \sum_{l=1}^{n} c_l 2^{l-1}} \beta_m = \alpha_m - e^{\frac{\pi i}{2^{m-1}} \sum_{l=1}^{m} a_l 2^{l-1}} \beta_m = 0,$$

and hence $\hat{f}(c)$ is correctly produced for all c.

The algorithm is also clearly seen to require $O(n)$ time, and thus the proof is completed. □

This algorithm has all the advantages of the basis-state de-quantised algorithm, but operates on a much larger ranger of input states, making it a much more powerful de-quantisation. Importantly, just like the basis-state de-quantisation, it is actually more efficient than the QFT algorithm. While this algorithm will not work on all separable input states like the tensor-contraction simulation in [6], it is a stronger de-quantisation in the sense that it gives a complete description of the output state as opposed to the probability of measuring a particular value, and is trivial to use as a subroutine in a larger de-quantisation.

5 Discussion

The ability to de-quantise the QFT algorithm brings up some interesting points. The two de-quantisations presented in this paper compute the Fourier transform on a restricted set of input states. On the other hand the standard QFT algorithm computes the Fourier transform on arbitrary separable or entangled input states. In fact, the standard QFT algorithm is a quantum implementation of the basis-state algorithm, but the linearity of quantum mechanics ensures that arbitrary input states are transformed by this simple algorithm. De-quantisation techniques such as the one presented, as well as those of [3, 4, 6], all have to efficiently simulate the linearity that is inherit in the quantum mechanical medium. The de-quantisations in this paper highlight the important distinction that should be made between the quantum Fourier transform and the quantum algorithm computing it. The QFT is a unitary transformation of an n-qubit state, while the QFT algorithm is a recipe for creating a sequence of local gates which computes the QFT on a given state. While these two notions are equivalent in quantum computation, when we depart from quantum

mechanics this is no longer the case, and the de-quantised algorithm does not suffice to compute the complete QFT.

It is interesting to note that both de-quantisations presented in this paper run in $O(n)$ time, more efficient than the $O(n^2)$ of the quantum algorithm. This is due to the restrictions imposed by measurement no longer being present when we develop a classical counterpart. This increase in efficiency is something not seen in other de-quantisations of the QFT which are based on the quantum circuit topology, and thus inherently and perhaps unnecessarily work within the restrictions the quantum circuit was designed under. The Separable De-quantised QFT algorithm computes the QFT on a large number of input states and it remains to be seen if this de-quantised algorithm can be applied to existing or new quantum algorithms to produce further de-quantisations. The fact that both the input and output states are separable also ensures the existence of a de-quantised inverse algorithm too, which is of practical significance.

Another issue worth noting is that we must be careful to consider the complexity involved in manipulating the complex amplitudes in a state-vector when performing de-quantisation. While it did not contribute to the complexity of the de-quantised algorithms presented in this paper, attention had to be paid to make sure this was the case, as this would not have been so if we implemented the directly obvious algorithm. In quantum computation, however, the amplitudes are just our representation of a property of physical states. It is these physical states, rather than the amplitudes, which are altered by unitary transformations, and as a result we observe the amplitudes changing. This reiterates the need for care when de-quantising, as the amplitudes have no a priori reason to be easily calculated, or computable at all for that matter.

6 Summary

We have shown that the quantum algorithm computing the QFT can be de-quantised into a classical algorithm which is more efficient and in many senses simpler than the quantum algorithm, primarily because the need to avoid measurement of the system is no longer present. However, the direct de-quantisation of the QFT algorithm leads to a classical algorithm which only acts on a basis-state. This difference is due to the linearity of quantum computation ensuring a basis-state algorithm computes the complete QFT, highlighting this linearity as a key feature in the power of the QFT. By examining the entangling power of the QFT we devised conditions for when the QFT leaves a separable state unentangled, and showed that this separability is invariant under phase-rotation of the input qubits. We extended our de-quantisation to work on this set of states without any loss of efficiency.

This de-quantisation of the QFT serves not only to illustrate more deeply the nature of the QFT, but also provides a useful tool for possibly de-quantising algorithms using the QFT with very little effort. Further, the techniques involved can help identify de-quantisable algorithms more easily, as well as aiding the

creation of new quantum algorithms and subroutines by deepening our understanding of what is needed to make a quantum algorithm so useful.

Acknowledgements

The author would like to thank Cristian Calude for many helpful discussions and much advice, as well as Tania Roblot for comments and suggestions. This work was in part supported by a University of Auckland Summer 2010 Fellowship.

References

1. J. Gruska, Quantum Computing, McGraw Hill, 1999.
2. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in: S. Goldwasser (Ed.), Proc. 35th Annual Symp. Found. Comput. Sci., IEEE Computer Society Press, 1994, pp. 124–134.
3. D. Aharonov, Z. Landau, J. Makowsky, The quantum FFT can be classically simulated, arXiv:quant-ph/0611156v2 (2007).
4. D. E. Browne, Efficient classical simulation of the quantum Fourier transform, New J. Phys. 9 (2007) 146.
5. R. Griffiths, C. Niu, Semiclassical Fourier transform for quantum computation, Phys. Rev. Lett. 76 (1996) 3228–3231.
6. N. Yoran, A. J. Short, Efficient classical simulation of the approximate quantum Fourier transform, Phys. Rev. A 76 (2007) 042321.
7. R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, Quantum algorithms revisited, Proc. R. Soc. Lond. A 1998 (1997) 339–354.
8. R. Jozsa, Geometric Issues in the Foundations of Science, Oxford University Press.
9. A. A. Abbott, The Deutsch-Jozsa problem: De-quantisation and entanglement, arXiv:0910.1990v2 (2009).
10. R. Jozsa, N. Linden, On the role of entanglement in quantum-computational speed-up, Proc. R. Soc. Lond. A 459 (2003) 2011–2032.
11. A. A. Abbott, C. S. Calude, Understanding the quantum computational speed-up via de-quantisation, EPTCS 26 (2010) 1–12.
12. C. S. Calude, De-quantizing the solution of Deutsch's problem, Int. J. Quantum Inf. 5 (2007) 409–415.
13. D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, Proc. R. Soc. Lond. A 400 (1985) 97–117.
14. D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation, Proc. R. Soc. Lond. A 439 (1992) 553–558.
15. S. Barnard, J. M. Child, Higher Algebra, Macmillan, London, 1936.
16. P. Jorrand, M. Mhalla, Separability of pure n-qubit states: two characterizations, Int. J. Found. Comput. Sci. 14 (2003) 797–814.

Axiomatization of Relativistic Physics in a Logical Framework

Hajnal Andréka, Judit X. Madarász, István Németi, Péter Németi, and Gergely Székely

Rényi Institute of Mathematics, Budapest
andreka@renyi.hu, madarasz@renyi.hu, nemeti@renyi.hu,
nemetip@gmail.com, turms@renyi.hu

Abstract. The aim of this talk is to give a logical introduction to relativity theory and relativistic hypercomputation. The talk is designed to give insight (for the logically minded) to these subjects.
We build up relativity theories (special, general, cosmological) as theories in the sense of mathematical logic. We intend to provide an easily understandable, logic based introduction to these theories. We also intend to give a logical insight to the most exotic and recent developments relevant to relativistic hypercomputation ranging from the recently discovered acceleration of the expanding universe through wormholes, timewarps and observational evidence for huge rotating black holes.
We axiomatize relativity theories within pure first-order logic using simple, comprehensible and transparent basic assumptions (axioms). We aim to prove the surprising predictions (theorems) of relativity theories from a few convincing axioms and to investigate the relationship between the axioms and the theorems.
In physics we do not know whether an axiom is true or not, we just presume so. Therefore, the role of the axioms (the role of statements that we assume without proofs) in physics is more fundamental than in mathematics. That is why we aim to formulate simple, logically transparent and intuitively convincing axioms. All the surprising or unusual predictions of a physical theory should be provable as theorems and not assumed as axioms. For example, the prediction "no observer can move faster than light" is a theorem in our approach and not an axiom.
There are many examples showing the benefits of using axiomatic method in the foundations of mathematics. That motivates our Hungarian school investigating logic and relativity to apply this method in the foundations of relativity theories. In any foundational work one should avoid tacit assumptions. First-order logic will be used as a device for forcing us to make all tacit assumptions explicit.
A novelty in our approach is that we try to keep the transition from special relativity to general relativity logically transparent and illuminating. We are going to "derive" the axioms of general relativity from that of special relativity in two natural steps. In the first step we extend special relativity of inertial observers to accelerated observers. This step provides us a theory which is strong enough to prove theorems even about gravitation via Einstein's Principle of Equivalence. In the second step we eliminate the difference between inertial and noninertial observers in the

level of axioms. This second natural step provides us a first-order logic axiomatization of general relativity suitable for further extensions and logical analysis.

In this talk we will put an emphasis on the spacetime aspects of relativity. At the same time, we indicate how the theories extend to the direction of covering relativistic dynamics including Einstein's famous insight $E = mc^2$.

Logical axiomatization of physics especially that of relativity theory is not at all a new idea, among others, it goes back to such leading mathematicians and philosophers as Hilbert, Gödel, Tarski, Reichenbach, Carnap, Suppes and Friedman. Our aims go beyond these approaches, because we not only axiomatize relativity theories, but also analyze their logical and conceptual structure.

Some of the questions we study to clarify the logical structure of relativity theories are:

- What is believed and why?
- Which axioms are responsible for certain predictions?
- What happens if we discard some axioms?
- Can we change the axioms and at what price?

We will also explore such frontier areas of cutting-edge science as, e.g., exploring the (remote) possibility of time travel via relativistic wormholes as suggested in papers of Thorne, Novikov, Visser, Yurtsever and others (e.g., "Timewarps"). We will touch upon the geometry of wormholes or timewarps (utilizing the firm logical foundation of our theory). We note that wormholes are highly relevant for hypercomputing, c.f., [4].

Among others, logical analysis makes relativity theory modular: we can replace some axioms with other ones, and our logical machinery ensures that we can continue working in the modified theory. This modularity might come handy, e.g., when we want to extend general relativity and quantum theory to a unified theory of quantum gravity.

References

1. Andréka, H., Madarász, J. X. and Németi, I., *Logic of space-time and relativity theory*. In: Handbook of spatial logics. Springer, 2007. pp.607-711. http://www.renyi.hu/pub/algebraic-logic/Logicofspacetime.pdf
2. Andréka, H., Madarász, J. X., Németi, I., Németi, P. and Székely, G., *Vienna Circle and Logical Analysis of Relativity Theory*, In: Stadler, F. (Ed.), Wiener Kreis und Ungarn, Veröffentlishungen des Instituts Wiener Kreis, Springer, to apper, 21pp. http://www.renyi.hu/pub/algebraic-logic/WKU-amnsz.pdf
3. Andréka, H., Németi, I. and Németi, P., *A logic based foundation and analysis of relativity theory*. Presentation at Logic and the foundations of physics, Brussels, 11-12 December 2008.
4. Andréka, H., Németi, I. and Németi, P., *General relativistic hypercomputing and foundation of mathematics*, Natural Computing 8(3): 499-516. 2009.
5. Andréka, H., Németi, I. and Németi, P., *Relativistic hypercomputing, physical realisticity*, Manuscript (based on the presentation of talk given at UC09 at Azores 2009), 59pp. Available from the authors.

6. Székely, G., *First-Order Logic Investigation of Relativity Theory with an Emphasis on Accelerated Observers* PhD thesis, Eötvös Loránt University, Budapest, 2009; 152pp. http://renyi.hu/~turms/phd/phd.pdf
7. Székely, G., *On Why-Questions in Physics*, In: Stadler, F. (Ed.), Wiener Kreis und Ungarn, Veröffentlishungen des Instituts Wiener Kreis, Springer, to apper, 9pp. http://renyi.hu/~turms/wqp.pdf

The Turing Machine and Uncertainty Principle

(Extended abstract)

Edwin Beggs [1], José Félix Costa* [1,2,3], and John Tucker [1]

[1] School of Physical Sciences
Swansea University, Singleton Park, Swansea, SA3 8PP
Wales, United Kingdom
E.J.Beggs@Swansea.ac.uk, J.V.Tucker@swansea.ac.uk
[2] Department of Mathematics, Instituto Superior Técnico
Universidade Técnica de Lisboa
fgc@math.ist.utl.pt
[3] Centro de Matemática e Aplicações Fundamentais do Complexo Interdisciplinar
Universidade de Lisboa

1 Introduction

Consider the classical model of a Turing machine with an oracle. The oracle is a one step external consultation device. The oracle may contain either non-computable information, or computable information provided just to speed up the computations of the Turing machine. Moreover, *the oracle is a set, i.e. a language*, e.g. over the input alphabet of the Turing machine. [4]

In this paper we will consider the abstract experimenter (e.g. the experimental physicist) as a Turing machine and the abstract experiment of measuring a physical quantity (using a specified physical apparatus) as an oracle to the Turing machine. The algorithm running in the Turing machine abstracts the experimental method of measurement (encoding the recursive structure of experimental actions) chosen by the experimenter.

It is standard to consider that to measure a real number μ,[5] e.g. the value of a physical quantity, the experimenter (now the Turing machine) should proceed by approximations. Thus, besides the value of μ, we will consider dyadic rational approximations (denoted by finite binary strings), and a procedure to measure μ *proved to be universal*.

What is intended to be measured? It can be a distance between two points, or an electric charge in a field, or the mass of a particle, etc. Measurable numbers were first considered a scientific enterprise by Geroch and Hartle in their famous paper [13], where they introduce the concept:

> *We propose, in parallel with the notion of a computable number in mathematics, that of a measurable number in a physical theory. The ques-*

* Corresponding author.

[4] By saying that a set is to be a language we are also emphasising that it is countable.

[5] Real numbers make part of the general setting in measurement theory.

tion of whether there exists an algorithm for implementing a theory may then be formulated more precisely as the question of whether the measurable numbers of the theory are computable.

Measurement is a scientific activity supported by a full theory developed since the beginning of the last century as a chapter of mathematical logic (see [11, 12, 15, 17, 4]), which is unexpectedly similar to oracle consultation but exhibiting new features in complexity theory. On the other side, scientific activity seen as algorithm running in a Turing machine is also not new in computational learning theory (see [16]).

Let us consider a very simple concrete example where the measurement of *inertial mass* is considered (see [6] for the complete case study): if we project a particle of known mass towards a particle of unknown mass, then the first will be reflected if its mass is less than the unknown mass, and it is projected forward together with the particle of unknown mass if its mass is greater than the unknown mass. Using binary search we are allowed, in principle, to read bit by bit the value of the unknown mass.[6] But we find a novelty: if we want to read the bits of μ using such a method, then the time needed for a single experiment is

$$\Delta t \ \sim \ \left| \frac{1}{m - \mu} \right|,$$

where m is the mass of the proof particle in that single experiment. This means that the time needed for a single experiment to read the bit i of the mass μ, using the proof particle of mass m of size i (number of its bits) is in the best case exponential in i.

This ideal experiment tells us that, if the abstract oracle to a Turing machine is to be replaced by an abstract physical measurement, then the time needed to consult the oracle is not any more a single step of computation but a number of time steps that will depend on the size of the query. Provided with such mathematical constructions, the main complexity classes involved in such computations, e.g. for the polynomial time case, change and deserve to be studied (see [6, 6]). New interesting classes emerge, namely those involved in the study of complexity of hybrid systems and analogue-digital systems such as mirror systems and neural nets (see [9, 18]). To sum up, *a first differentiation of physical oracles from classical oracles is a cost function T with a signature of a complexity function, the number of steps of busy waiting of the Turing machine as a function of the size of the query.*

In the physical world, it is not conceivable that a proof particle of mass m can be set with infinite precision. If we consider that precision is not infinite but unbounded, i.e., as big as we need, than we can continue reading the bits of μ. The same complexity classes are defined. But suppose that we reject unbounded

[6] Note that this activity of *measuring inertial mass* is not that much different from the activity of *measuring mass using a balance scale* and a toolbox of standard weights. It is only more simple to describe from a point of view of the dynamics involved in finding the unknown value.

precision to favor the most common and realistic a priori fixed precision criterion. Then we prove that, using stochastic methods, we are still able to read the bits of μ. To make our claim rigorous, we say that the lack of precision in measurement will not constitute an obstacle to the reading of the bits of μ.

Oracles should be regarded as information with possible error that take time to consult (see [1, 2, 5]).

However, the Turing machine imposes limitations to what is effectively accessible to physical observations. E.g., not all masses are measurable. Not due to the limitations in measurements where experimental errors occur, not because quantum phenomena puts a limit to measurements, but because of a more essential internal limitation of physicists conceived as Turing machines. The mathematics of computation theory does not allow the reading of bits of physical quantities beyond a certain limit: even if they could be measured with infinite precision by physics, they could not be measured by physical-mathematical reasons.

2 Reflexions

The measurement of a distance (*SME* for brief) taught us that oracles should be regarded as information with possible error (see [1, 2]). The measurement of a mass (*STME* for brief) taught us that oracles may take time to consult, may have a cost (a first experiment in measuring mass, the *CME* is analysed in [6]).

The reaction of the reader towards the *gedankenexperiment* of measuring mass considered in the preceding section might well be of discomfort: such devices can not be built. But the reader should notice that this reaction is a consequence of a diffuse philosophy that considers the Turing machine an object of a different kind: both the abstract physical machine and the Turing machine are non-realizable objects. For the implementation of the Turing machine the engineer would need either unbounded space and a physical support structure, or unbounded precision in some finite interval to code for the contents of the tape; each time the size of the written word in the working tape increases by one symbol, the precision needed will increase. The experiment can be set up to some precision in the same way that the Turing machine can be implemented up to some accuracy.

Knowing that both objects, the Turing machine and the measurement device, are of the same ideal nature, the reader may wonder what is the purpose of such an experiment from the computational point of view. The physical experiment exhibits the character of an oracle, an external device to the Turing machine. It gives to the concept of an oracle a new epistemology: the oracle is not any more an abstract entity, but an abstract physical entity; the oracle is not any more a one step transition of the Turing machine, but a device that needs time to be consulted; the oracle is not any more a relativization mechanism, but it has physical content: it can only be consulted up to some accuracy; moreover the degrees of accuracy in the consultation of the oracle can be studied. For some, this setting can be seen as that of a computer connected to an analogue device.[7]

[7] A kind of hybrid system.

As emergent result, we are led to the conclusion that infinite precision and unbounded precision are of the same ontological nature, as the computational process has taught us for decades. Different experiments imply a conclusion similar to that of the work on neural nets in the nineties ([18]): to compute up to time t,[8] only $O(f(t))$ bits of the unknown are needed, where f is a function depending on the undergoing experiment. As we will see, this result is more about the nature of numbers and arithmetic than about physics or neurodynamics.

Of relevance is the objective of such construction of a Turing machine connected with an abstract physical device (that can not be built) from the (physical) sciences point of view. The idea is also the same as a Turing machine is for computation science: to be able to describe limiting results and negative results. The limiting results are obtained in the perfect Platonic world. In the same way, limiting results of a computer are not about our *computers*, but about the limit of a *computer*. The same happens with physical oracles. The experiments allow us to study the limiting results on measurement.

It seems that the same methods allow us to inspect the concept of measurement in the physical sciences. Since measurement is made in most situations by comparisons between observables, what we describe applies not only to the measurement of *mass*, but also to the measurement of a *electric charge* or to the measurement of a *magnetic charge*.

Note that the apparent counterintuitive theoretical *gedankenexperiment* of increasing accuracy towards infinite precision is part of measurement theory in the context of a physical theory. In this respect, Geroch and Hartle write in [13]:

> *The notion "measurable" involves a mix of natural phenomena and the theory by which we describe those phenomena. Imagine that one had access to experiments in the physical world, but lacked any physical theory whatsoever. Then* no *number w could be shown to be measurable, for, to demonstrate experimentally that a given instruction set shows w measurable would require repeating the experiment an infinite number of times, for a succession of εs approaching zero. One could not even demonstrate that a given instruction set shows measurability of any number at all, for it could turn out that, as ε is made smaller, the resulting sequence of experimentally determined rationals simply fails to converge. It is only a* theory *that can guarantee otherwise.*

3 Looking Closer to the Scattering Experiment

We have studied the elastic collision between two particles of varying mass, considering them to be point masses interacting by contact (see [6]). Let us now prove that the same behaviour, namely in the protocol between the Turing machine and the apparatus, characterizes the general scatter machine, where scatterer particles with *negative electric charge* scatter firing particles with *negative*

[8] E.g., to simulate the physical system up to time t.

electrical charge too. This is a more conventional settlement of the scattering experiment.

Collisions usually involve projecting particles towards other particles that are at rest in the system of the laboratory. Let μ denote the mass of the particles at rest in the system of the laboratory and m denote the mass of the particles projected, dyadic rationals used as proof particles. Let θ_m be the deflexion angle of a firing particle in the system of the laboratory.

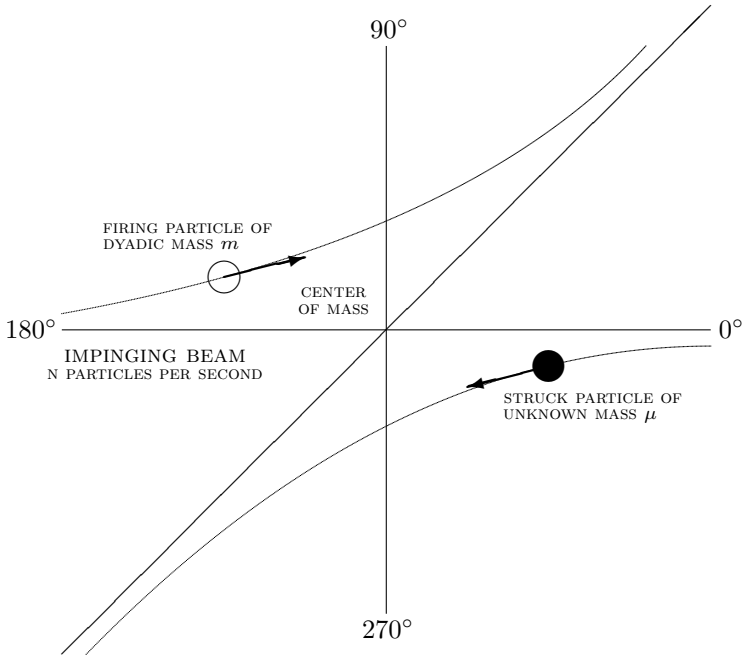


Fig. 1. Scattering a particle: view from the system of reference of the center of mass. When the particle of mass m is fired, the particle of unknown mass μ is at rest in the system of reference of the laboratory.

1. Suppose that $m < \mu$, i.e., the bombarding particle is lighter than the particle that is being struck. Then the maximum angle of scattering θ_m is always π.
2. Suppose that $m > \mu$. In this case we can easily see that the maximum of θ_m is less than $\pi/2$.
3. Suppose that $m = \mu$. The maximum of θ_m is $\pi/2$.

There is an abrupt discontinuity in the form of the cross section, since for $m = \mu$ the maximum value of θ_m is $\pi/2$, while for m very slightly less than μ, the maximum value of θ_m suddenly jumps to π. This is not a real physical discontinuity because the cross section itself, $q(\theta_m)$, approaches to zero for angles greater than $\pi/2$, as m approaches μ:

$$q(\theta_m) \;=\; \frac{\pi(Z_1 Z_2 e^2)^2}{4E^2}\,\frac{\cos(\theta_m/2)}{\sin^3(\theta_m/2)}\;. \tag{1}$$

The number of particles collected after time t is then given by the product of the total cross section and the number of particles projected by the beam that is $N_0 t$, where N_0 is the number of particles per unit time. The time taken for detection of particles of mass m less than μ is then inversely proportional to the difference of masses, i.e.,

$$t* \;=\; \frac{16\mu E^2 N\!\uparrow}{\pi^2 N_0 (Z_1 Z_2 e^2)^2}\;\frac{1}{|\mu - m|}\;, \tag{2}$$

where $N\!\uparrow$ is the level of detection in number of particles. Now we can describe the algorithm in full detail. This algorithm is of a different kind of those considered in [1, 6, 3].

Let $T : \mathbb{N} \to \mathbb{N}$ be the time given for the experiment to take place as a function (total map) of the size of the sequence of bits setting the value of the mass of the bombarding particles. The function T can be seen as a *schedule*, i.e., in each experiment, in order to read the $|m|$-bit of the mass μ, $T(|m|)$ gives the amount of time steps that *the experimenter accepts to wait* until resuming the experimental conditions. The function T can either be a computable function or a non-computable function of its argument.

A possible formalisation of Geroch and Hartle's concepts of measurable number (mass in the current case) is given in the following definition, adapted from our work in [6]:

Definition 1. *A mass μ is said to be* measurable *if there exists a computable schedule* [9] *T such that the digits of μ can be computed by performing the scattering experiment repeatedly. Otherwise, the mass is said to be* non-measurable.

The Turing machine is connected to the scatter machine (*STME*) in the same way as it would be connected to an oracle: we replace the query state with a *scattering state* (q_s), the *yes* state with a *less than* state (q_l), and the *no* state with a *more than* state (q_m). The resulting computational device is called the *analogue-digital scattering machine*, and we refer to the *unknown mass* of the struck particles of an analogue-digital scattering machine when meant to discuss the unknown mass of the corresponding *STME*. After setting the mass m, the *STME* will fire particles of mass m, wait $T(|m|)$ time units, and then check if some particles have been detected in the interval $]\frac{\pi}{2}, \frac{3\pi}{2}[$, then the Turing machine computation will be resumed in the state q_l. If no particles have been detected $]\frac{\pi}{2}, \frac{3\pi}{2}[$, then the Turing machine computation will be resumed in the state q_m. Details on the analogue-digital connection can be found, e.g., in [3].

The folllowing algorithm is parameterised to the time T given to the experimenter to test for the crossing of the threshold of detection of particles in the angular section $]\frac{\pi}{2}, \frac{3\pi}{2}[$.

[9] Truly speaking, it should be a time constructible function.

The scattering (Rutherford) algorithm: procedure to read the first n bits of an unknown mass μ — $SCATTERING(T)$

Input the required precision n — number of places to the right of the left leading 0;
$m, m_1 := 0$; $m_2 := 1$; $w := \lambda$; *counter* $:= 0$;
$\mu \in (0, 1)$;
While $|m_1| \leq n$ do begin {*loop* 1}

Loop {*loop* 2}
$m := (m_1 + m_2)/2$;
Fire the beam with particles of mass m ;
If firing particles are detected in the range $]\frac{\pi}{2}, \frac{3\pi}{2}[$ in time $T(|m|)$, then
Begin
$m_1 := m$;
Exit the loop
End;
Else $m_2 := m$
End Loop; {*loop* 2}
$m_2 := 1$; *counter* := *counter* $+ 1$;
If *counter* $>$ *limcounter* then return *timeout* ;

End While; {*loop* 1}
Output the dyadic rational denoted by m_1.

The question is to know if the algorithm (supposedly correct) allows mass to be measurable according with definition 1. This question is answered in the paper.

Proposition 1. *The AD-Protocol of the analogue-digital scattering machine is at least exponential in the size of the query, in the size of the mass of the bombarding particles.*

This proposition, generalised to any physical experiment of measurement, constitutes what we have called the BCT Conjecture, dicussed in [5, 4, 7, 3] in another context.

4 Uncertainty

The probability that the next bit of the non-dyadic unknown μ has been measured after time t is denoted by $\chi_\mu(t)$. It turns out that $\chi_\mu(t) \to 1$ when $t \to \infty$. E.g., a function that *resembles* such a probability is the inverse tangent

$$\frac{2}{\pi}\ \tan^{-1}(t)\ H(t)$$

where H is the Heaviside step function (the one derived from maximum entropy method is analised in the full paper). A probability distribution for such a function $\chi_\mu(t)$ is

$$q_\mu(t) = \frac{2}{\pi}\frac{1}{1+t^2}\ H(t)\ .$$

The conjugate (Fourier transform) of $q_\mu(t)$ is related to the sensibility to the frequency/energy (as we will see, in the case of the analogue-digital scatter machine, is related to the kinetic energy of the proof particle after the collision) required in the determination of the next bit of μ; it is denoted by $p_\mu(f)$. The idea is that, if the ignorance about μ is high, i.e., if the probability $\chi_\mu(t)$ gets close to 1 only for very large value of t, then *the energy associated with the determination of the next bit of μ should be very small.*

In our paper we study the common distributions $q_\mu(t)$ for physical oracle access. E.g., for a constant distribution q_μ within a window of cost of size $2t_0$, the normalization gives

$$p_\mu(f) \ = \ \sqrt{2t_0}\ \frac{\sin(2\pi f t_0)}{2\pi f t_0}\ . \tag{3}$$

This function in the limiting case $t_0 \to \infty$ can only be understood in the context of distributions; the function is 0 everywhere (for every f) except at the origin ($f = 0$), where it is undefined (it is infinite).

The functions $q_\mu(t)$ and $p_\mu(f)$ constitute a Fourier pair. We prove the Heisenberg's principle for any physical oracle (adapting from the classical theorem found, e.g., in [14]). For that purpose we assume that the Conjecture BCT holds; in consequence, the probability $q_\mu(t)$ can not be proved *a priori* to be 1 for finite t.

Proposition 2.

$$\widehat{q}_\mu\ \widehat{p}_\mu \geq \frac{1}{4\pi}$$

where $\widehat{q}_\mu$ is the spreads of the probability distribution of time needed to read the next bit of the unknown value μ — i.e., the uncertainty on the time — and $\widehat{p}_\mu$ is the spreads of the probability distribution of the frequency/energy needed for the same task — i.e., the uncertainty on the frequency/energy.

We can then say that our physical oracles are characterized by the following principle:

$$\widehat{q}_\mu\ \widehat{p}_\mu \geq \frac{1}{4\pi}\ . \tag{4}$$

For the dyadic rationals, the spreads of time are infinite and, consequently, the scatter in frequency/energy should be zero: it means that the experimental apparatus has to distinguish rest from motion with speed arbitrarily close to zero.

For the well-behaved real numbers, with a high dispersion of bits, the spreads of time are well delimited as much as the scatter of frequency/energy.

For the non-measurable real numbers, numbers with either arbitrarily large gaps of 0s between consecutive 1s or arbitrarily large gaps of 1s between consecutive 0s, the scatter of frequency/energy is small, meaning that the experimenter has to measure smaller and smaller energy values to differentiate between 0 and 1. However, in this case, the scatter of time as function of the next bit might not be a computable function. I.e., the value of $\widehat{q}_\mu$ might not be computable, such as the value of $\widehat{p}_\mu$ that might not be possible to estimate.

We don't pretend to establish any connection between accessibility to physical oracles and the Quantum Mechanics. However, let us point out that the relation between energy and time in the Quantum Mechanics is not that much Quantum Mechanics as noted by Bunge in [10].

Acknowledgements

Edwin Beggs, José Félix Costa and John Tucker would like to thank EPSRC for their support under grant EP/C525361/1. The research of José Félix Costa is also supported by FEDER and FCT Plurianual 2007.

References

1. Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 464(2098):2777–2801, 2008.
2. Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles II. Upper bounds. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 465(2105):1453–1465, 2009.
3. Edwin Beggs, José Félix Costa, and John Tucker. The impact of limits of computation on a physical experiment. 2009. Technical Report.
4. Edwin Beggs, José Félix Costa, and John V. Tucker. Computational Models of Measurement and Hempel's Axiomatization. In Arturo Carsetti, editor, *Causality, Meaningful Complexity and Knowledge Construction*, volume 46 of *Theory and Decision Library A*, pages 155–184. Springer, 2009.
5. Edwin Beggs, José Félix Costa, and John V. Tucker. Physical experiments as oracles. *Bulletin of the European Association for Theoretical Computer Science*, 97:137–151, 2009. An invited paper for the "Natural Computing Column".
6. Edwin Beggs, José Félix Costa, and John V. Tucker. Limits to measurement in experiments governed by algorithms. *Mathematical Structures in Computer Science*, 2010.
7. Edwin Beggs, José Félix Costa, and John V. Tucker. Physical oracles: The Turing machine and the Wheatstone bridge. *Studia Logica*, 95:271–292, 2010. Special issue: The contributions of Logic to the Foundations of Physics, Editors D. Aerts, S. Smets & J. P. Van Bendegem.
8. Olivier Bournez and Michel Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168(2):417–459, 1996.
9. Mario Bunge. *Foundations of Physics*, volume 10 of *Springer Tracts in Natural Philosophy*. Springer-Verlag, 1967.

10. Norman Robert Campbell. *Foundations of Science, The Philosophy of Theory and Experiment*. Dover, 1957.
11. Rudolf Carnap. *Philosophical Foundations of Physics*. Basic Books, 1966.
12. Robert Geroch and James B. Hartle. Computability and Physical Theories. *Foundations of Physics*, 16(6):533–550, 1986.
13. R. W. Hamming. *Digital Filters*. Prentice-Hall International, Inc., second edition, 1989.
14. Carl G. Hempel. Fundamentals of concept formation in empirical science. *International Encyclopedia of Unified Science*, 2(7), 1952.
15. Sanjay Jain, Daniel Osherson, James S. Royer, and Arun Sharma. *Systems That Learn. An Introduction to Learning Theory*. The MIT Press, 1999.
16. David H. Krantz, Patrick Suppes, R. Duncan Luce, and Amos Tversky. *Foundations of Measurement*. Dover, 2009.
17. Hava T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, 1999.

Foundations of Analog Algorithms

Olivier Bournez[1] and Nachum Dershowitz[2]

[1] LIX, Ecole Polytechnique, 91128 Palaiseau Cedex, FRANCE
Olivier.Bournez@lix.polytechnique.fr
[2] Tel Aviv University, School of Computer Science, Tel Aviv University, Ramat Aviv, 69978 ISRAEL
nachumd@post.tau.ac.il

Abstract. We propose a formalization of analog algorithms, extending the framework of abstract state machines to continuous-time models of computation.

The machine to be described here, like almost every contrivance, apparatus, or machine in practical use, is based very largely upon what has been accomplished by others who previously labored in the same field.
—Description of the U.S. Coast and Geodetic Survey tide-predicting machine, no. 2 (1915)

1 Introduction

Abstract state machines (ASMs) [12] constitute a most general model of sequential digital computation, one that can operate on any level of abstraction of data structures and native operations. By virtue of the Abstract State Machine Theorem of [13], any algorithm that satisfies three "Sequential Postulates" can be step-by-step emulated by an ASM. These postulates formalize the following intuitions: (I) we are talking about deterministic state-transition systems; (II) the information in states suffices to determine future transitions and may be captured by logical structures that respect isomorphisms; and (III) transitions are governed by the values of a finite and input-independent set of (variable-free) terms.

All notions of algorithms for classical discrete-time models of computation in computer science, like Turing machines, random-access memory (RAM) machines, as well as classical extensions of them, including oracle Turing machines, alternating Turing machines, and the like, [13] fall under the purview of the Sequential Postulates. This provides a basis for deriving computability theory, or even complexity theory, upon these very basic axioms about what an algorithm really is. In particular, adding a fourth axiom about initial states, yields a way to derive a proof of the Church-Turing Thesis [4,10,5].

Our goal in the current work is to adapt and extend ideas from work on ASMs to the analog case, that is to say, from notions of algorithms for digital models or systems to *analog systems*.

We do not want to deal here only with the issue of "continuous space", that is, discrete-time models or algorithms with real-valued operations, since these already fit comfortably within the standard ASM framework. See [1,2]. Indeed, algorithms for discrete-time analog models, like algorithms for the Blum-Shub-Smale model of computation [3], can also be covered by the settings of [13].

We want to deal with truly analog systems, that is to say continuous space and time systems. As surveyed in [7], several approaches have led to continuous-time models of computations. In particular, one approach inspired by continuous-time analog machines, has its roots in models of natural or artificial analog machinery. An alternate approach, one that can be refered to as inspired by continuous-time system theories, is broader in scope, and derives from research on continuous-time systems theory from a computational perspective. Hybrid systems and automata theory, for example, are two such sources.

At its beginning, continuous-time computation theory was mainly concerned with analog machines. Determining which systems can actually be considered as computational models is a very intriguing question. This relates to the philosophical discussion about what is a programmable machine. Nonetheless, there are some early examples of built analog devices that are generally accepted as programmable machines. They include Pascal's 1642 *Pascaline* [9], Hermann's 1814 *Planimeter*, Bush's landmark 1931 *Differential Analyzer* [6], as well as Bill Phillips' 1949 water-run *Financephalograph* [18]. Continuous-time computational models also include neural networks and systems that can be built using electronic analog devices. Such systems begin in some initial state and evolves over time in response to some input signal. Results are read off from the evolving state and/or from a terminal state.

Another line of development of continuous-time computation models has been motivated by hybrid systems, particularly by questions related to the hardness of their verification and control. Here models are not seen as models of necessarily analog machines, but as abstraction of systems about which one would like to establish some properties or derive verification algorithms.

Our aim is here to cover here all these models, with a uniform notion of computation and of algorithm.

We believe capturing the notion of algorithm or computation for analog systems is a first step towards a better understanding of computability theory for continuous-time systems. We refer to [7] for a survey and discussion on continuous-time computability theories.

Even this first step is a non-trivial task. Some work in this direction has been done for simple signals. See, for example, [8]. Simple (loop-free) examples are the geometric algorithms in [15]. An interesting approach to specifying some continuous-time evolutions, based on abstract state machines and using infinitesimals, is [16]. However, we believe that a general framework capturing general analog systems is wanting.

The rest of this paper is organized as follows. In the next section, we introduce dynamical transition systems, defining signals and transition systems. In Section 3, we introduce abstract dynamical systems. Then, in Section 4, we

define what an algorithmic dynamical system is. Finally, in Section 5, we define analog programs and provide some examples.

2 Dynamical Transition Systems

Analog systems can be thought of as "states" that evolve over "time". The systems we deal with receive inputs, called "signals", but do not otherwise interact with their environment.

2.1 Signals

Typically, a signal is a function from an interval of time to a "domain" value, or to a tuple of atomic domain values. For simplicity, we will presume that signals are indexed by real-valued time $\mathbb{T} = \mathbb{R}$, are defined only for a finite initial (open or closed) segment of $\mathbb{T}$, and take values in some domain D. Usually, the domain is more complicated than simple real numbers; it could be something like a tuple of infinitesimal signals. Every signal $u : \mathbb{T} \rightharpoonup D$ has a *length*, denoted $|u|$, such that $u(j)$ is undefined beyond $|u|$. To be more precise, the length of signals that are defined on any of the intervals $(0, \ell), [0, \ell), (0, \ell], [0, \ell]$ is ℓ. In particular, the length of the (always undefined) *empty* signal, ε, is 0, as is the length of any point signal, defined only at moment 0.

The *concatenation* of signals is denoted by juxtaposition, and is defined as expected, except that concatenation of a right-closed signal with a left-closed one is only defined if they agree on the signal value at those closed ends. The empty signal ε is a neutral element of the concatenation operation.

Let $\mathcal{U}$ be the set of signals for some particular domain D. The *prefix* relation on signals, $u \leq v$, holds if there is a $w \in \mathcal{U}$ such that $v = u\,w$. As usual, we write $u < v$ for *proper* prefixes ($u \leq v$ but $u \neq v$). It follows that $\varepsilon \leq u \leq uw$ for all signals $u, w \in \mathcal{U}$. And, $u \leq v$ implies $|u| \leq |v|$, for all u, v.

2.2 Transition Systems

Definition 1 (Transition System). *A* transition system $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{U}, \mathcal{T} \rangle$ *consists of the following:*

- *A nonempty set (or class)* $\mathcal{S}$ *of* states *with a nonempty subset (or subclass)* $\mathcal{S}_0 \subseteq \mathcal{S}$ *of* initial *states.*
- *A set* $\mathcal{U}$ *of input signals over some domain* D*.*
- *A* $\mathcal{U}$*-indexed family* $\mathcal{T} = \{\tau_u\}_{u \in \mathcal{U}}$ *of* state transformations $\tau_u : \mathcal{S} \to \mathcal{S}$*.*

It will be convenient to abbreviate $\tau_u(X)$ as just X_u, the state of the system after receiving the signal u, having started in state X. We will also use $X_{\widetilde{u}}$ as an abbreviation for the *trajectory* $\{X_v\}_{v<u}$, describing the past evolution of the state.

For simplicity, we are assuming that the system is deterministic. Notice that the ASM framework, that is to say the classical ASM framework for digital

algorithms, initialy defined for deterministic systems, has latter been extended to nondeterministic transitions in [14,11].

Definition 2 (Dynamical System). *A* dynamical system $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{U}, \mathcal{T} \rangle$ *is a transition system, where the transformations satisfy*

$$\tau_{uv} = \tau_v \circ \tau_u, \tag{1}$$

for all $u, v \in \mathcal{U}$*, and where* τ_ε *is the identity function on states.*

This implies that $X_{uv} = (X_u)_v$.

Remark 1. It follows from this definition that $\tau_{(uv)w} = \tau_{u(vw)}$, since composition is associative. It also follows that $\tau_a \circ \tau_a = \tau_a$, for point signal a, since then $aa = a$.

Timed Transitions *Timed transition systems* are a special case, where signals are the identity function and $D = \mathbb{T}$.

3 Abstract Dynamical Systems

3.1 Abstract States

A vocabulary $\mathcal{V}$ is a finite collection of fixed arity function symbols, some of which may be marked *relational*. A term whose outermost function name is relational is termed *Boolean*.

Definition 3 (Abstract Transition System). *An* abstract transition system *is a dynamical transition system whose states* $\mathcal{S}$ *are (first-order) structures over some finite vocabulary* $\mathcal{V}$*, such that the following hold:*

(*a*) *States are closed under isomorphism, so if* $X \in \mathcal{S}$ *is a state of the system, then any structure* Y *isomorphic to* X *is also a state in* $\mathcal{S}$*, and* Y *is an initial state if* X *is.*
(*b*) *Input signals are closed under isomorphism, so if* $u \in \mathcal{U}$ *is a signal of the system, then any signal* v *isomorphic to* u *(that is, maps to isomorphic values) is also a signal in* $\mathcal{U}$*.*
(*c*) *Transformations preserve the domain (base set); that is,* $\mathrm{Dom}\ X_u = \mathrm{Dom}\ X$ *for every state* $X \in \mathcal{S}$ *and signal* $u \in \mathcal{U}$*.*
(*d*) *Transformations respect isomorphisms, so, if* $X \cong_\zeta Y$ *is an isomorphism of states* $X, Y \in \mathcal{S}$*, and* $u \cong_\zeta v$ *is the corresponding isomorphism of input signals* $u, v \in \mathcal{U}$*, then* $X_u \cong_\zeta Y_v$*.*

In particular, system evolution is *causal* ("retrospective"): a state at any given moment is completely determined by past history and the current input signal. This is analogous to the Abstract State Postulate for discrete algorithms, as formulated by Gurevich, except that subsequent states X_u depend on the whole signal u, not just the prior state X and current input.

To keep matters simple, we are assuming (unrealistically) that all operations are total. Instead, we simply model partiality by including some *undefined* element $\perp$ in domains, as in most of the ASM literature. See however discussions in [1,2].

Vocabularies We will assume that the vocabularies of all states include the Boolean truth constants, the standard Boolean operations, equality, and function composition, and that these are always given their standard interpretations. We treat predicates as truth-valued functions, so states may be viewed as algebras.

There are idealized models of computation with reals, such as the BSS model [3], for which true equality of reals is available in all states. On the other hand, there are also models of computable reals, for which "numbers" are functions that approximate the idealized number to any desired degree of accuracy, and in which only partial equality is available. See [1,2] for how to extend the abstract-state-machine framework to deal faithfully with such cases.

3.2 Locations in States

Locations Since a state X is a structure, it interprets function symbols in $\mathcal{V}$, assigning a value b from Dom X to the "location" $f(a_1, \ldots, a_k)$ in X for every k-ary symbol $f \in \mathcal{V}$ and values $a_1, \ldots, a_k$ taken from Dom X. In this way, state X assigns a value $[\![t]\!]_X \in$ Dom X to any ground term t over $\mathcal{V}$. Similarly, a state X assigns the appropriate function value $[\![f]\!]_X$ to each symbol $f \in \mathcal{V}$.

States It is convenient to view each state as a collection of the graphs of its operations, given in the form of a set of location-value pairs, each written conventionally as $f(a_1, \ldots, a_k) \mapsto b$, for $a_1, \ldots, a_k, b \in$ Dom X. This allows one to apply set operations to states.

3.3 Updates of States

We need to capture the changes to a state that are engendered by a system. For a given abstract transition system, define its *update function* Δ as follows:

$$\Delta(X) = \lambda u.\ X_u \setminus X$$

We write $\Delta_u(X)$ for $\Delta(X)(u)$. The trajectory of a system may be recovered from its update function, as follows:

$$X_u = (X \setminus \nabla_u(X)) \cup \Delta_u(X) \tag{2}$$

where

$$\nabla_u(X) := \{\ell \mapsto [\![\ell]\!]_X : \ell \mapsto b \in \Delta_u(X) \text{ for some } b\}$$

are the location-value pairs in X that are updated by Δ_u.

4 Algorithmic Dynamic Systems

We say that states X and Y *agree*, with respect to a set of terms T, if $[\![s]\!]_X = [\![s]\!]_Y$ for all $s \in T$. This will be abbreviated $X =_T Y$. We also say that states X and Y are *similar*, with respect to a set of terms T, if or all terms $s, t \in T$, we have $[\![s]\!]_X = [\![t]\!]_X$ iff $[\![s]\!]_Y = [\![t]\!]_Y$. This will be abbreviated $X \sim_T Y$.

4.1 Algorithmicity

Definition 4 (Algorithmic Transitions). *An abstract transition system with states $\mathcal{S}$ over vocabulary $\mathcal{V}$ is* algorithmic *if there is a fixed* finite *set T of* critical *terms over $\mathcal{V}$, such that $\Delta_u(X) = \Delta_u(Y)$ for any two of its states $X, Y \in \mathcal{S}$ and signal $u \in \mathcal{U}$, whenever X and Y agree on T. In symbols:*

$$X =_T Y \Rightarrow \Delta_u(X) = \Delta_u(Y) \tag{3}$$

This implies

$$X_{\widetilde{u}} =_T Y_{\widetilde{u}} \Rightarrow \Delta_u(X) = \Delta_u(Y) \tag{4}$$

Furthermore, similarity should be preserved:

$$X_{\widetilde{u}} \sim_T Y_{\widetilde{v}} \Rightarrow X_{ua} \sim_T Y_{va} \tag{5}$$

where $a \in \mathcal{U}$ is any point signal ($|a| = 0$).

Following the reasoning in [13, Lemma 6.2], every new value assigned by $\Delta_u(X)$ to a location in state X is the value of some critical term. That is, if $\ell \mapsto b \in \Delta_u(X)$, then $b = [\![t]\!]_X$ for some critical $t \in T$.

Agreeability of states is preserved by algorithmic transitions:

Lemma 1. *For an algorithmic transition system with critical terms T, it is the case that*

$$X =_T Y \Rightarrow X_u =_T Y_u \tag{6}$$

for any states $X, Y \in \mathcal{S}$ and input signal $u \in \mathcal{U}$.

4.2 Flows and Jumps

A "jump" in a trajectory is a change in the dynamics of the system, in apposition to "flows", during which the dynamics is fixed. Formally, a jump corresponds to a change in the equivalences between critical terms, whereas, when the trajectory "flows", equivalences between critical terms is kept invariant. Accordingly, we will say that a trajectory $X_{\widetilde{u}}$ *flows* if all intermediate states X_w and X_v ($\epsilon < w < v < u$) are similar. It *jumps* at its end if there is no prefix $w < u$ such that all intermediate X_v, $w < v < u$, are similar to X_u. It *jumps* at its beginning if there is no prefix $w \leq u$ such that all intermediate X_v, $\epsilon < v < w$, are similar to X.

4.3 Analgorithms

Definition 5 (Analog Algorithm). *An* analog algorithm *(or "analgorithm") is an algorithmic (abstract) transition system, such that no trajectory has more than a finite number of (prefixes that end in) jumps.*

In other words, an analog algorithm is a signal-indexed deterministic state-transition system (Definitions 1 and 2), whose states are algebras that respect isomorphisms (Definition 3), whose transitions are governed by the values of a fixed finite set of terms (Definition 4), and whose trajectories do not change dynamics infinitely often (Definition 5).

4.4 Properties

System evolution is *causal* ("retrospective"): a state at any given moment is completely determined by past history and the current input signal.

Theorem 1. *For any analog algorithm, the trajectory can be recovered from the immediate past (or updates from the past). That is, X_u, for right-closed signal u, can be obtained (up to isomorphism) as a function of $X_{\widetilde{u}}$ (that is, the X_v, for $v < u$) plus the final input u_*.*

In fact, X_u depends on arbitrarily small segments $X_{u(t,|u|)}$ $(t < |u|)$ of past history.

5 Programs

5.1 Definition

Definition 6 (ASM). *An* ASM program *P over a vocabulary $\mathcal{V}$ is a finite text, taking one of the following forms:*

- *A* constraint *statement $v_1, \ldots, v_n$* **such that** *C, where C is a Boolean condition over $\mathcal{V}$ and the v_i are terms over $\mathcal{V}$ (usually subterms of C) whose values may change in connection with execution of this statement.*
- *A* parallel *statement $[P_1 \parallel \cdots \parallel P_n]$ ($n \geq 0$), where each of the P_i is an ASM program over $\mathcal{V}$. (If $n = 0$, this is "do nothing" or "skip".)*
- *A* conditional *statement* **if** *C* **then** *P, where C is a Boolean condition over $\mathcal{V}$, and P is an ASM program over $\mathcal{V}$.*

We can use an assignment statement $f(s_1, \ldots, s_n) := t$ as an abbreviation for $f(s_1, \ldots, s_n)$ **such that** $f(s_1, \ldots, s_n) = t$. But bear in mind that the result is instantaneous, so that $x := 2x$ is tantamount to $x := 0$, regardless of the prior value of x. Similarly, $x := x + 1$ is only possible if the domain includes an "infinite" value ∞ for which $\infty = \infty + 1$.

5.2 Examples

We restrict in a first step to analog algorithms that purely flow, that is to say with no jump.

In simple continuous-time systems, the state evolves continually, governed by ordinary differential equations, say. Flow programs invoke a time parameter, which we assume is supplied by the input signal.

Example 1 (Pendulum). The motion of an idealized simple pendulum is governed by the second-order differential equation

$$\theta'' + \frac{g}{L}\theta = 0$$

where θ is angular displacement, g is gravitational acceleration, and L is the length of the pendulum rod. Let the signal $u \in \mathcal{U}$ be just real time. States report the current angle $\theta \in \mathcal{V}$. All states are endowed with the same (or isomorphic) operations for real arithmetic, including sine and square root, interpreting standard symbols. Initial states contain values for g, L, and the initial angle θ_0 when the pendulum is released.

For small θ_0, the flow trajectory $\tau_t(X)$ can be specified simply by

$$\theta = \theta_0 \cdot \sin\left(\sqrt{\frac{g}{L}} \cdot \imath\right)$$

where $\imath$ is the input port and nothing but θ changes from state to state. The update function is, accordingly,

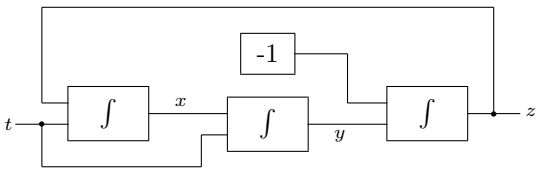
$$\Delta_t(X) = \left\{\theta \mapsto \theta_0 \cdot \sin\left(\sqrt{\frac{g}{L}} \cdot \imath\right)\right\}$$

Hence, the critical term is $\theta_0 \cdot \sin(\sqrt{g/L} \cdot \imath)$.

It can be described by program

$$[\theta \textbf{ such that } \theta = \theta_0 \cdot \sin\left(\sqrt{\frac{g}{L}} \cdot \imath\right)]$$

Example 2 (GPAC). One of the most famous models of analog computations is the General Purpose Analog Computer (GPAC) of Claude Shannon [17]. Here is a (non-mimimal) GPAC that generates sine and cosine: in this picture, the $\int$ signs denote some integrator, and the -1 denote some constant block.



If initial conditions are set up correctly, such a system will evolve according to the following initial value problem

$$\begin{cases} x' = z & x(0) = 1 \\ y' = x & y(0) = 0 \\ z' = -y & z(0) = 0\,, \end{cases}$$

It follows that $x(t) = \cos(t)$, $y(t) = \sin(t)$, $z = -\sin(t)$.

In other words, this simple GPAC that generates sine and cosine can be modeled implicitly as a system with initial state having $x = 1; y = 0; z = 0$ and by a program

$$[x, y, z \textbf{ such that } x' = z \wedge y' = x \wedge z' = x]$$

where we presume that x', y', z' denote derivatives of corresponding functions.

The proposed model can also adequately describe systems (like a bouncing ball) in which the dynamics change periodically:

Example 3. The physics of a bouncing ball are given by the explicit flow equations

$$v = v_0 - g \cdot t$$
$$x = v \cdot t$$

where g is the gravitational constant, v_0 is the velocity when last hitting the table, and t is the time signal—except that upon impact, each time $x = 0$, the velocity changes according to

$$v_0 = -k \cdot v$$

where k is the coefficient of impact. The critical Boolean term is $x = 0$. In any finite time interval, this condition changes value only finitely many times. □

It can be described by a program like

$$[\textbf{if } x \neq 0 \textbf{ then } x, v \textbf{ such that } v = v_0 - g \cdot t, x = v \cdot t \parallel \textbf{if } x = 0 \textbf{ then } v_0 := -k \cdot v],$$

where x stands for its height, v its speeed. Every time the ball bounces, its speed is reduced by a factor k.

References

1. Andreas Blass, Nachum Dershowitz, and Yuri Gurevich. Exact exploration. Technical Report MSR-TR-2009-99, Microsoft Research, Redmond, WA. July 2009. Submitted.
2. Andreas Blass, Nachum Dershowitz, and Yuri Gurevich. Exact Exploration and Hanging Algorithms. *Computer Science Logic* 2010, Brno, Czech Republic. Lecture Notes in Computer Science, Springer-Verlag, 2010.
3. Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc. (NS)*, 21:1–46, 1989.
4. Udi Boker and Nachum Dershowitz. The Church-Turing Thesis over Arbitrary Domains. Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday, Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich, eds., Lecture Notes in Computer Science, vol. 4800, Springer-Verlag, Berlin, pp. 199–229, 2008.
5. Udi Boker and Nachum Dershowitz. Three Paths to Effectiveness. Fields of Logic and Computation: Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday, Andreas Blass, Nachum Dershowitz, and Wolfgang Reisig, eds., Lecture Notes in Computer Science, vol. 6300, Springer-Verlag, Berlin, 2010.
6. V. Bush. The differential analyser. *Journal of the Franklin Institute*, 212(4):447–488, 1931.

7. Olivier Bournez and Manuel L. Campagnolo. A survey on continuous time computations. In *New Computational Paradigms. Changing Conceptions of What is Computable* (Cooper, S.B. and Löwe, B. and Sorbi, A., Eds.). New York, Springer-Verlag, pp. 383-423. 2008.
8. Joëlle Cohen and Anatol Slissenko. On implementations of instantaneous actions real-time ASM by ASM with delays. *Proc. of the 12th Intern. Workshop on Abstract State Machines (ASM'2005)*, Paris, France, pp. 387–396, 2005.
9. Doug Coward. Doug Coward's Analog Computer Museum, 2006. `http://dcoward.best.vwh.net/analog/`.
10. N. Dershowitz and Y. Gurevich. A natural axiomatization of computability and proof of Church's Thesis. *The Bulletin of Symbolic Logic*, 14(3):299-350, 2008.
11. Andreas Glausch and Wolfgang Reisig. A semantic characterization of unbounded-nondeterministic abstract state machines *Algebra and Coalgebra in Computer Science*, Lecture Notes in Computer Science, vol. 4624, Springer, Berlin, pp. 242–256, 2007.
12. Yuri Gurevich. Evolving algebras 1993: Lipari guide. In E. Börger, editor, *Specification and Validation Methods*, pages 9–36. Oxford University Press, 1995.
13. Yuri Gurevich. Sequential abstract-state machines capture sequential algorithms. *ACM Transactions on Computational Logic*, 1, 2000.
14. Yuri Gurevich and Tatiana Yavorskaya. On bounded exploration and bounded nondeterminism. Technical Report MSR-TR-2006-07, Microsoft Research, Redmond, WA. January 2006.
15. Wolfgang Reisig. On Gurevich's theorem on sequential algorithms. Acta Informatica, 39(5):273–305, 2003.
16. Heinrich Rust. Hybrid abstract state machines: Using the hyperreals for describing continuous changes in a discrete notation. In Y. Gurevich, P. Kutter, M. Odersky, and L. Thiele, eds., *International Workshop on Abstract State Machines (Monte Verita, Switzerland)*, TIK-Report 87, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, pp. 341–356, March 2000.
17. Claude E. Shannon. Mathematical theory of the differential analyser. *Journal of Mathematics and Physics*, 20:337–354, 1941.
18. Wikipedia. MONIAC computer. `http://en.wikipedia.org/wiki/MONIAC_Computer`.

Algebraic Characterizations of Complexity-Theoretic Classes of Real Functions

Olivier Bournez[1], Walid Gomaa[2,3], and Emmanuel Hainry[2,4]

[1] Ecole Polytechnique, LIX, 91128 Palaiseau Cedex, France
Olivier.Bournez@lix.polytechnique.fr
[2] Loria, BP 239 - 54506 Vandœuvre-lès-Nancy Cedex, France
[3] Alexandria University, Faculty of Engineering, Alexandria, Egypt
walid.gomaa@loria.fr
[4] Nancy Université, Université Henri Poincaré, Nancy, France
Emmanuel.Hainry@loria.fr

Abstract. Recursive analysis is the most classical approach to model and discuss computations over the real numbers.Recently, it has been shown that computability classes of functions in the sense of recursive analysis can be defined (or characterized) in an algebraic machine independent way, without resorting to Turing machines. In particular nice connections between the class of computable functions (and some of its sub- and sup-classes) over the reals and algebraically defined (sub- and sup-) classes of $\mathbb{R}$-recursive functions à la Moore 96 have been obtained. However, until now, this has been done only at the computability level, and not at the complexity level. In this paper we provide a framework that allows us to dive into the complexity level of real functions. In particular we provide the first algebraic characterization of polynomial-time computable functions over the reals. This framework opens the field of implicit complexity of analog functions, and also provides a new reading of some of the existing characterizations at the computability level.

Keywords: Recursive Analysis, Polynomial Time, Algebraic Characterization, Real Computation, Oracle Turing Machines

1 Introduction

Building a well founded theory of computation over the reals is a crucial task. However, computability over the reals is not as well understood as the corresponding notion over discrete objects where the Church-Turing thesis yields a clear equivalence between different computational models. When talking about continuous computation several approaches have been developed with various motivations but without so-clear relationships. Such approaches include the Blum-Shub-Smale (BSS) model [1, 2], Shannon's General Purpose Analog Computer (GPAC) [3], algebraically defined classes of functions over the reals à la Moore 96 ($\mathbb{R}$-recursive functions) [4], as well as the recursive analysis approach.

Recursive analysis was introduced by Turing [5], Grzegorczyk [6], and Lacombe [7]. It can be considered as the most classical approach to talk about

computability and complexity of functions over the real numbers, as its foundations are already present in Alan Turing's 1936 seminal paper. In recursive analysis, a function $f : \mathbb{R} \to \mathbb{R}$ is computable if there exists some computable functional, or Type 2 machine, that maps any sequence of rational numbers quickly converging to x to another sequence quickly converging to $f(x)$.

There is no hope to unify all approaches of continuous computations: for example the BSS models can not be conciliated with the recursive analysis viewpoint, as a non-continuous function can be computed in the BSS framework. However, if we put aside this latter model, which is more motivated by the algebraic complexity of problems rather than being a universal model, some recent works have shown strong connections between recursive analysis, Shannon's GPAC, and $\mathbb{R}$-recursive functions. These results basically state that all these paradigms are more or less equivalent: see [8, 9] or survey [10]. This can be considered somehow as yielding a kind of phenomenon for analog computations like the Church's thesis for discrete computations.

However, up till now discussions have mainly been restricted to the computability level, and not to the complexity level.

Connecting models, known to be related at the computability level, at the complexity level is an even more ambitious goal. An immediate deep problem is that of defining the traditional complexity notions for some of the models such as the GPAC. One reason is that there is no robust and well defined notion of time and space for these models, as shown by several attempts [4, 11, 12, 10].

We show in this paper that it is indeed possible to relate models at the complexity level when restricting to the recursive analysis and $\mathbb{R}$-recursive functions approaches. There is indeed an unambiguous, well developed, and rather well understood theory of complexity in recursive analysis [13]. We relate it to a subclass of $\mathbb{R}$-recursive functions, that is, to a machine-independent algebraically defined class of functions over the reals à la Moore 96 [4].

In particular this paper presents the first algebraic machine-independent characterization of polynomial-time computable functions in the sense of recursive analysis.

We provide as a side effect a whole framework for implicit complexity in recursive analysis that gives a way to relate computability and complexity over the reals to computability and complexity over the integers. We also extend [14], and prove that computable functions over the reals correspond to functions generable by Shannon's GPAC; we extend [9, 15, 8] and prove that computable functions and elementary-time computable functions correspond to natural subclasses of $\mathbb{R}$-recursive functions. In particular, unlike [14, 9, 15, 8], we provide characterizations that work even for non-Lipschitz functions (and that differ slightly for Lipschitz functions). This well founded framework may be a significant step towards a sane computability and complexity theory of functions over the reals.

Potential applications of polynomial-time characterizations include the possibility of proving whether a given function can be computed in polynomial time without resorting to efficiently program it, as well as the possibility of building

methods to automatically derive computational properties of programs/systems, in the lines of [16–18] for discrete programs.

We also believe in the pedagogical value of our characterizations. They yield ways to define computability and complexity over the reals without resorting to any kind of machinery in the spirit of (Type 1 or Type 2) Turing machines. This is a very natural and intuitive paradigm that avoids discrete machinery when talking about continuous computation.

2 Related Work

We prove our results by relating the notion of (polynomial-time) computable functions over the reals to the corresponding notion over the integers. Our setting is actually proved to be robust to approximations: one does not need to be able to compute exactly the corresponding class over the integers, but only some defined approximation of it in order to be able to compute the corresponding class over the reals.

Hence, our framework gives a way to rely on algebraic machine-independent characterizations of computable functions over the integers. Several such characterizations are known [19]: in particular, Kleene's functions are well known to capture exactly the discrete functions computable by Turing machines. Cobham [20], and later Bellantoni and Cook [21], were among the first to propose algebraically defined characterizations of polynomial-time computable discrete functions. Our main theorem relies on Bellantoni and Cook's ideas in [21]. Other machine independent characterizations of classical computability and complexity classes (see survey [19]) over the integers could also be considered.

Notice that our framework is different from the one proposed by Campagnolo and Ojakian in [22]: in particular, it has the main advantage of allowing to talk not only about the computability level but also about the complexity level. It should also be noticed that our characterization relies exclusively on functions over the reals, hence it can not be compared with approaches such as [23] or [24] which explore complexity of type 2 functionals. Algebraic characterizations of functions over more general domains, including the reals, have been obtained in [25]. However, the obtained characterization in this latter paper is rather different to the ones discussed here: on one hand, a more abstract setting that is not restricted to real functions is considered there, but on the other hand the discussion is only restricted to the computability level, and less in the spirit of the above mentioned models of continuous computation.

In this paper, for ease of presentation, we only consider functions defined over compact domains. The constructions described here can indeed be extended to functions over arbitrary domains.

3 Essentials of Recursive Analysis

In this section, we recall some basic definitions from recursive analysis: see [26, 13] for a full detailed presentation. Let $\mathbb{D} = \{\frac{a}{2^b} : \textit{ for integers } a, b \textit{ and } b \geq 0\}$ be the set of dyadic rationals. These are the rationals with finite binary representation.

Definition 1. *Assume $x \in \mathbb{R}$. A* Cauchy sequence *representing x is a function $\varphi_x : \mathbb{N} \to \mathbb{D}$ that converges at a* binary rate*: $\forall n \in \mathbb{N}$: $|x - \varphi_x(n)| \leq 2^{-n}$. Given $x \in \mathbb{R}$, let CF_x denote the class of Cauchy functions that represent x.*

Definition 2 (Computability of real functions). *Let f be a function $f : D \subseteq \mathbb{R} \to \mathbb{R}$, where D has only one connected component (in the following discussion we deal almost exclusively with either $D = [0, 1]$ or $D = \mathbb{R}$). We say that f is* computable *if there exists a function-oracle Turing machine $M^{()}$ such that for every $x \in D$, for every $\varphi_x \in CF_x$, and for every $n \in \mathbb{N}$ the following holds: $|M^{\varphi_x}(n) - f(x)| \leq 2^{-n}$.*

If $D = [0, 1]$, then we say f is *polytime computable* if the computation time of $M^{\varphi_x}(n)$ is bounded by $p(n)$ for some polynomial p. In case $D = \mathbb{R}$, we say f is *polytime computable* if the computation time of $M^{\varphi_x}(n)$ is bounded by $p(k, n)$ for some polynomial p where $k = \min\{j : x \in [-2^j, 2^j]\}$.

It is well known that continuity is a necessary condition for real computation, though it is not sufficient. The following definition introduces the notion of 'modulus of continuity' which in some sense quantifies the concept of continuity and provides a useful tool in the investigation of real computation [27].

Definition 3 (Modulus of continuity). *Consider a function $f : \mathbb{R} \to \mathbb{R}$. Then f has a* modulus of continuity *if there exists a function $m : \mathbb{N}^2 \to \mathbb{N}$ such that for all $k, n \in \mathbb{N}$ and for all $x, y \in [-2^k, 2^k]$ the following holds: if $|x - y| \leq 2^{-m(k,n)}$, then $|f(x) - f(y)| \leq 2^{-n}$. If f is defined over $[0, 1]$ the same definition holds except that the parameter k is not necessary anymore, that is $m : \mathbb{N} \to \mathbb{N}$.*

Notice that the existence of a modulus of continuity for a function f implies that this function is continuous. In analogy with [13, corollary 2.14], computability over unbounded domains can be characterized as follows [27].

Proposition 1. *Let a function $f : \mathbb{R} \to \mathbb{R}$. Then f is computable iff there exist two computable functions $m : \mathbb{N}^2 \to \mathbb{N}$ and $\psi : \mathbb{D} \times \mathbb{N} \to \mathbb{D}$ such that*

1. *m is a modulus of continuity for f,*
2. *ψ is an approximation function for f, that is, for every $d \in \mathbb{D}$ and every $n \in \mathbb{N}$ the following holds: $|\psi(d, n) - f(d)| \leq 2^{-n}$.*

When restricting attention to polytime computability two additional requirements need to be added to the previous proposition: (1) the modulus m is a polynomial function, that is $m(k, n) = (k + n)^b$ for some $b \in \mathbb{N}$ and (2) $\psi(d, n)$ is computable in time $p(\mathit{length}(d) + n)$ for some polynomial p.

4 Characterizing Polytime Real Complexity over Compact Domains

In this section, we prove that it is possible to relate computability over the reals to computability over the integers. We do it in two steps. In the first step, we consider the special case of Lipschitz functions. In the second step, we discuss how to avoid the Lipschitz hypothesis, and consider general functions.

Without loss of generality we assume that the compact domain is always the unit interval $[0,1]$. Let's first provide a preliminary first result to help explaining what we would like to get.

4.1 A preliminary first result

A real function over a compact interval can be characterized by the discrete projection of a function with domain $[0,1] \times \mathbb{R}$. The extra dimension can be viewed as representing the precision of the computed approximation.

Proposition 2 (Complexity over $[0,1]$ vs Complexity over $[0,1]\times\mathbb{R}$). *The following are equivalent:*

1. *a function $f : [0,1] \to \mathbb{R}$ is polytime computable,*
2. *there exists a polytime computable function $g : [0,1] \times \mathbb{R} \to \mathbb{R}$ such that:*

$$\forall x \in [0,1], \forall y \in \mathbb{N}\colon |g(x,y) - yf(x)| \leq 1. \tag{1}$$

We would like to talk about functions g with assertions like above but quantification is only done over the integers, that is to say about assertions like (1) but with something like $\forall x \in \mathbb{N}$ instead of $\forall x \in [0,1]$.

Moving to such a full integer characterization we are faced with the problem of how the notion of continuity, which is exclusive to real computable functions, can be transferred to the integer domain.

4.2 Lipschitz functions

For Lipschitz functions this is facilitated by the fact that such functions provide us with free information about their continuity properties. A real function $f\colon [0,1] \to \mathbb{R}$ is *Lipschitz* if there exists a constant $K \geq 0$ such that for all $x_1, x_2 \in [0,1]$ the following holds: $|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$.

Proposition 3 (Complexity over $[0,1]$ vs Complexity over $\mathbb{R} \times \mathbb{R}$). *Fix an arbitrary constant $\epsilon \geq 0$. Let f be a Lipschitz function on $[0,1]$. Then the following are equivalent:*

1. *f is polytime computable,*
2. *there exists a polytime computable function $g\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that:*

$$\forall x \in \mathbb{N}, \forall y \in \mathbb{N}^{\geq 1}, x \leq y\colon |g(x,y) - yf(\frac{x}{y})| \leq \epsilon \tag{2}$$

In order to interrelate with discrete complexity classes we suggest to employ the following notion of *approximation*.[5]

Definition 4 (Approximation). *Let $\mathcal{C}$ be a class of functions from $\mathbb{R}^2$ to $\mathbb{R}$. Let $\mathcal{D}$ be a class of functions from $\mathbb{N}^2$ to $\mathbb{N}$. Let f be a function defined on $[0,1]$.*

1. *We say that $\mathcal{C}$* approximates *$\mathcal{D}$ if for any function $g \in \mathcal{D}$, there exists some function $\tilde{g} \in \mathcal{C}$ such that for all $x, y \in \mathbb{N}$ we have*

$$|\tilde{g}(x,y) - g(x,y)| \leq 1/4 \tag{3}$$

2. *We say that f is $\mathcal{C}$-*definable *if there exists a function $\tilde{g} \in \mathcal{C}$ such that the following holds*

$$\forall x \in \mathbb{N}, \forall y \in \mathbb{N}^{\geq 1}, x \leq y \colon |\tilde{g}(x,y) - yf(\frac{x}{y})| \leq 3 \tag{4}$$

We then have the following result:[6]

Theorem 1 (Complexity over $[0,1]$ vs approximate complexity over $\mathbb{N}^2$). *Consider a class $\mathcal{C}$ of polytime computable real functions that approximates the class of polytime computable discrete functions. Assume that $f : [0,1] \to \mathbb{R}$ is Lipschitz. Then f is polytime computable iff f is $\mathcal{C}$-definable.*

In the right-to-left direction of the previous, Eq. (4) implicitly provides a way to efficiently approximate f from $\tilde{g} \restriction \mathbb{N}^2$. Computability of f is possible, in particular at the limit points, from the fact that it is Lipschitz (hence continuous), and efficiency is possible by the fact that $\tilde{g}$ is polytime computable. The left-to-right direction relates polytime computability of real functions to the corresponding discrete notion.

4.3 Avoiding the Lipschitz hypothesis

The major obstacle to avoid the Lipschitz hypothesis is how to implicitly encode the continuity of f in discrete computations. This is done in two steps: (1) encoding the modulus of continuity which provides information at arbitrarily small rational intervals (however, it does not tell anything about the limit irrational points) and (2) bounding the behavior of the characterizing function g both at unit intervals and at its integer projection.

We need another notion of 'approximation' that is a kind of converse to that given in Definition 4.

Definition 5 (Polytime computable integer approximation). *A function $g : \mathbb{R}^d \to \mathbb{R}$ is said to have a* polytime computable integer approximation *if there exists some polytime computable function $h : \mathbb{N}^d \to \mathbb{N}$ with $|h(\bar{x}) - g(\bar{x})| \leq 1$ for all $\bar{x} \in \mathbb{N}^d$.*

[5] Notice that the choice of the constants $\frac{1}{4}$ and 3 in Definition 4 is arbitrary.

[6] Note that all these results still hold if we replace 'polytime computable' by just 'computable'.

A sufficient condition is that the restriction of function g to integers is polytime computable. The choice of the constant 1 is then due to the fact that this is the best estimated error when trying to compute the floor of a real function. Now we define a special class of functions that will be used to implicitly describe information about the smoothness of real functions; its role can be compared to that of the moduli of continuity.

Definition 6. *Consider a function $T\colon \mathbb{N} \to \mathbb{N}$ and define $\#_T : \mathbb{R}^{\geq 1} \to \mathbb{R}$ by $\#_T[x] = 2^{T(\lfloor \log_2 x \rfloor)}$. When T is a polynomial function with $T(x) = \Theta(x^k)$ we write $\#_k$ to simplify the notation.*

The following proposition is then the non-Lipschitz version of Proposition 3.

Proposition 4 (Complexity over $[0,1]$ vs complexity over $\mathbb{R}\times\mathbb{R}$). *Fix an arbitrary constant $\epsilon \geq 0$. The following are equivalent:*

1. *a function $f : [0,1] \to \mathbb{R}$ is polytime computable,*
2. *there exists some function $g\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ such that*
   (a) g has a polytime computable integer approximation,
   (b) for some integer k,

$$\forall x \in [0,1], \forall y \in \mathbb{R}^{\geq 1}\colon |g(x \cdot \#_k[y], y) - yf(x)| \leq \epsilon, \tag{5}$$

   (c) for some integer M,

$$\forall x_1, x_2 \in \mathbb{R}^{\geq 0}, y \in \mathbb{R}^{\geq 1}\colon |x_1 - x_2| \leq 1 \Rightarrow |g(x_1, y) - g(x_2, y)| \leq M \tag{6}$$

We need to consider real functions that are well behaved relative to their restriction to $\mathbb{N}^2$. For ease of notation, we will use $[a,b]$ to denote $[a,b]$ or $[b,a]$, according to whether $a < b$ or the contrary.

Definition 7 (Peaceful functions). *A function $g : \mathbb{R}^2 \to \mathbb{R}$ is said to be* peaceful *if $\forall x \in \mathbb{R}^{\geq 0}, \forall y \in \mathbb{N}\colon g(x,y) \in [g(\lfloor x \rfloor, y), g(\lceil x \rceil, y)]$. We say that a class $\mathcal{C}$ of real functions* peacefully approximates *some class $\mathcal{D}$ of integer functions, if the subclass of peaceful functions of $\mathcal{C}$ approximates $\mathcal{D}$.*

Definition 8. *Let $\mathcal{C}$ be a class of functions from $\mathbb{R}^2$ to $\mathbb{R}$. Let us consider a function $f : [0,1] \to \mathbb{R}$ and a function $T\colon \mathbb{N} \to \mathbb{N}$.*

1. *We say that f is T-$\mathcal{C}$-*definable *if there exists some peaceful function $g \in \mathcal{C}$ such that*

$$\forall x \in \mathbb{N}, \forall y \in \mathbb{N}^{\geq 1}, x \leq \#_T[y]\colon |g(x,y) - yf(\frac{x}{\#_T[y]})| \leq 2, \tag{7}$$

2. *We say that f is T-*smooth *if there exists some integer M such that*

$$\forall x, x' \in \mathbb{R}^{\geq 0}, \forall y \in \mathbb{R}^{\geq 1}, x, x' \leq \#_T[y]\colon \\ |x - x'| \leq 1 \Rightarrow y|f(\frac{x}{\#_T[y]}) - f(\frac{x'}{\#_T[y]})| \leq M \tag{8}$$

Notice the similarity in the role that $\#_T[y]$ plays in the previous definition and as a modulus of continuity for f. Now we can have the non-Lipschitz version of Theorem 1.

Theorem 2. *(Complexity over $[0,1]$ vs approximate complexity over $\mathbb{N}^2$) Consider a class $\mathcal{C}$ of real functions that peacefully approximates polytime computable discrete functions, and whose functions have polytime computable integer approximations.*[7] *Then the following are equivalent:*

1. *a function $f\colon [0,1] \to \mathbb{R}$ is polytime computable,*
2. *there exists some integer k such that*
   (a) *f is n^k-$\mathcal{C}$-definable,*
   (b) *f is n^k-smooth.*

Proof. (1) $\Rightarrow$ (2) : Let $f\colon [0,1] \to \mathbb{R}$ be a polytime computable function. By Proposition 4 for $\epsilon = 3/4$, there exists some function g with a polytime computable integer approximation h such that (5) holds. Now, by the hypothesis of this theorem, there exists some peaceful $\tilde{h} \in \mathcal{C}$ such that $\forall x, y \in \mathbb{N}$: $|\tilde{h}(x,y) - h(x,y)| \leq 1/4$. Hence $\forall x, y \in \mathbb{N}\colon |\tilde{h}(x,y) - g(x,y)| \leq 1 + \frac{1}{4} = \frac{5}{4}$.

Finally, we have[8] (through change of variables in Eq. (5) and restricting the domains of the variables to $\mathbb{N}$)

$$\forall x \in \mathbb{N}, \forall y \in \mathbb{N}^{\geq 1}, x \leq \#_k[y]\colon |\tilde{h}(x,y) - yf(\frac{x}{\#_k[y]})| \leq \frac{5}{4} + \frac{3}{4} = 2 \qquad (9)$$

Hence, condition 2a holds. Now, by (2c) of Proposition 4, we know that for all $x \in \mathbb{R}^{\geq 0}$, $y \in \mathbb{R}^{\geq 1}$, and $\delta \in [0,1]$: $|g(x+\delta, y) - g(x,y)| \leq M$ for some integer M. Then by using Eq. (5) (after variable change and renaming), condition (2b) is satisfied.

(2) $\Rightarrow$ (1) : Let $g \in \mathcal{C}$ be a peaceful function that n^k-$\mathcal{C}$-defines f. Proof is by applying Proposition 4 as follows. From the hypothesis of this theorem g has a polytime computable integer approximation, hence condition 2a of Proposition 4 is satisfied. Condition 2a of the current theorem is equivalent to condition 2b of Proposition 4 by: (1) letting $\epsilon = 2$, (2) renaming of the variables, and (3) observing that the proof of Proposition 4 can be easily adapted to a new version of condition 2b for which x and y take only integer values. Using the fact that g is peaceful (controlling the behavior of g between integer points) condition (2c) of Proposition 4 can be easily verified.

The previous theorem can be generalized to any complexity class as indicated by the following corollary.

[7] A sufficient condition for that is that restrictions to integers of functions from $\mathcal{C}$ are polytime computable.

[8] Note that all these results still hold if we replace 'polytime computable' by just 'computable'.

Corollary 1. *Let $\mathcal{D}$ be a class of time-constructive functions from $\mathbb{N}$ to $\mathbb{N}$ that includes polynomial functions and closed under composition. Consider a class $\mathcal{C}$ of functions that peacefully approximate the class of discrete functions computable in time $\mathcal{D}$; and whose functions have integer approximations computable in time $\mathcal{D}$.* [9] *Then the following are equivalent:*

1. *a function $f\colon [0,1] \to \mathbb{R}$ is computable in time $\mathcal{D}$,*
2. *there exists some $T \in \mathcal{D}$ such that*
   (a) *f is T-$\mathcal{C}$-definable,*
   (b) *f is T-smooth.*

Proof. The proof is similar to that of the previous theorem. It should be noted that if f is computable in time bounded by $\mathcal{D}$ then it has a modulus in $\mathcal{D}$. This is a direct consequence of [13, Theorem 2.19].

5 Applications

In this section we apply the above results to algebraically characterize some computability and complexity classes of real functions. We first obtain some restatements and extensions of already known results, using our framework. We then provide new results, in particular, the main result given by theorems 3 and 4 and corollary 2 which provide algebraic machine independent characterizations of polynomial time computable functions.

5.1 GPAC-generable functions

The General Purpose Analog Computer, introduced by Claude Shannon in [3] to model a mechanical device, can be seen in a modern perspective as what can be computed using analog electronics. It consists of circuits interconnecting basic blocks that can be constants, adders, multipliers, and integrators. GPAC-computable functions have been characterized in different ways since the introduction of the model. In the following we will use Graça and Costa's characterization by PIVP (Polynomial Initial Value Problems) [28]. A function is said to be PIVP if it is a component of the solution of a differential equation of the following form:

$$\begin{cases} y(t_0) = y_0 \\ y'(t) = p(t, y) \end{cases}$$

with $y : \mathbb{R}^n \to \mathbb{R}$ and p is a vector of polynomial functions. The next lemma follows from the constructions in [14]:

Lemma 1. *PIVP functions is a class of computable functions that peacefully approximate total (discrete) recursive functions.*

[9] A sufficient condition for that is restrictions to integers of functions from $\mathcal{C}$ are computable in time $\mathcal{D}$.

Then the following result follows directly from Theorem 1 (and Footnote 6), and from Corollary 1 (and Footnote 8).

Proposition 5 (Variation of [8]). *A Lipschitz function $f : [0,1] \to \mathbb{R}$ is computable iff it is PIVP-definable.*

Proposition 6 (Extension of [8]). *Let $f : [0,1] \to \mathbb{R}$ be some T-smooth function, for some total recursive function $T : \mathbb{N} \to \mathbb{N}$. Then f is computable iff it is T-PIVP-definable.*

5.2 Particular classes of $\mathbb{R}$-recursive functions

A function algebra $\mathcal{F} = [\mathcal{B}; \mathcal{O}]$ is the smallest class of functions containing a set of basic functions $\mathcal{B}$ and their closure under a set of operations $\mathcal{O}$.

Elementarily computable functions: class $\mathcal{L}$ Let us now consider the class $\mathcal{L}$ defined in [15]: $\mathcal{L} = [0, 1, -1, \pi, U, \theta_3; COMP, LI]$, where π is the mathematical constant $\pi = 3.14..$, U is the set of projection functions, $\theta_3(x) = \max\{0, x^3\}$, $COMP$ is the classical composition operation, LI is Linear Integration. From the constructions of [15], we know that this class captures the discrete elementary functions. In addition the following lemma follows from the constructions in [9].

Lemma 2. *$\mathcal{L}$ is a class of real functions computable in elementary time that peacefully approximates total discrete elementarily computable functions.*

Again using the above results we can obtain characterizations of the class of elementarily computable analysis functions:

Proposition 7 (Variation of [15]). *A Lipschitz function $f : [0,1] \to \mathbb{R}$ is computable in elementary time iff it is $\mathcal{L}$-definable.*

Proposition 8 (Extension of [15]). *Let $f : [0,1] \to \mathbb{R}$ be some T-smooth function, for some elementary function $T : \mathbb{N} \to \mathbb{N}$. Then f is computable in elementary time iff it is T-$\mathcal{L}$-definable.*

As in [15, 9], we can also characterize in a similar way the functions computable in time $\mathcal{E}_n$ for $n \geq 3$, where $\mathcal{E}_n$ represents the n-th level of the Grzegorczyk hierarchy.

Recursive functions: class $\mathcal{L}_\mu$ Let us now consider the class $\mathcal{L}_\mu$ defined in [9]: $\mathcal{L}_\mu = [0, 1, U, \theta_3; COMP, LI, UMU]$, where a zero-finding operator UMU has been added. This class is known (see [9]) to extend the class of total (discrete) recursive functions; from the constructions in this latter paper one can show:

Lemma 3. *$\mathcal{L}_\mu$ is a class of computable functions that peacefully approximate the class of total discrete recursive functions.*

And hence, as a consequence of Theorem 1 and Corollary 1, we obtain:

Proposition 9 (Variation of [9]). *A Lipschitz function $f : [0,1] \to \mathbb{R}$ is computable iff it is $\mathcal{L}_\mu$-definable.*

Proposition 10 (Extension of [9]). *Let $f : [0,1] \to \mathbb{R}$ be some T-smooth function, for some total recursive function $T : \mathbb{N} \to \mathbb{N}$. Then f is computable iff it is T-$\mathcal{L}_\mu$-definable.*

5.3 Main result: polytime computable functions

We are now ready to provide our main result: an algebraic characterization of polytime computable functions over the reals.

To do so, we define a class of real functions which are essentially extensions to $\mathbb{R}$ of the Bellantoni-Cook class [21]. This latter class was developed to exactly capture discrete polytime computability in an algebraic machine-independent way. In the next definition any function $f(x_1, \ldots, x_m; y_1, \ldots, y_n)$ has two types of arguments (see [21]): *normal* arguments which come first followed by *safe* arguments using ';' for separation. For any $n \in \mathbb{Z}$ we call $[2n, 2n+1]$ an even interval and $[2n+1, 2n+2]$ an odd interval.

Definition 9. *Define the function algebra*

$$\mathcal{W} = [0, 1, +, -, U, p, c, parity; SComp, SI]$$

1. *zero-ary functions for the constants* 0 *and* 1,
2. *a binary addition function:* $+(;x,y) = x + y$,
3. *a binary subtraction function:* $-(;x,y) = x - y$,
4. *a set of projection functions* $U = \{U_i^j : i, j \in \mathbb{N}, i \leq j\}$ *where:* $U_i^{m+n}(x_1, \ldots, x_m; x_{m+1}, \ldots, x_{m+n}) = x_i$,
5. *a polynomial conditional*[10] *function* c *defined by:* $c(;x,y,z) = xy + (1-x)z$.
6. *a continuous parity function:* $parity(;x) = \max(0, 2/\pi sin(\pi x))$. *Hence,* $parity(;x)$ *is non-zero if and only if* x *lies inside an even interval. Furthermore, for any* $n \in \mathbb{Z}$ *the following holds:* $\int_{2n}^{2n+1} parity(;x)dx = 1$.
7. *a continuous predecessor function* p *defined by:* $p(;x) = \int_0^{x-1} parity(;t)dt$. *Note that when* x *belongs to an even interval* $p(;x)$ *acts exactly like* $\lfloor \frac{x}{2} \rfloor$. *On an odd interval* $[2n+1, 2n+2]$, *it grows continuously from* n *to* $n+1$.
8. *a safe composition operator SComp: given a vector of functions* $\bar{g}_1(\bar{x};) \in \mathcal{W}$, *a vector of functions* $\bar{g}_2(\bar{x};\bar{y}) \in \mathcal{W}$, *and a function* $h \in \mathcal{W}$ *of arity* $len(\bar{g}_1) + len(\bar{g}_2)$ *(where len denotes the vector length). Define new function*

$$f(\bar{x};\bar{y}) = h(\bar{g}_1(\bar{x};); \bar{g}_2(\bar{x};\bar{y})) \tag{10}$$

It is clear from the asymmetry in this definition that normal arguments can be repositioned in safe places whereas the opposite can not happen.

[10] If $x = 1$, the conditional is equal to y; if $x = 0$, it is equal to z. Between 0 and 1, it stays between y and z.

9. *safe integration operator*[11] *SI: given functions* $g, h_0, h_1 \in \mathcal{W}$. *Let* $p'(;x) = p(;x-1)+1$. *Define a new function solution of the ODE:*

$$\begin{aligned} f(0,\bar{y};\bar{z}) =& g(\bar{y};\bar{z}) \\ \partial_x f(x,\bar{y};\bar{z}) =& \, parity(x;)[h_1(p(x;),\bar{y};\bar{z}, f(p(x;),\bar{y};\bar{z})) \\ & - f(2p(x;),\bar{y};\bar{z})] \\ & + parity(x-1;)[h_0(p'(x;),\bar{y};\bar{z}, f(p'(x;),\bar{y};\bar{z})) \\ & - f(2p'(x;)-1,\bar{y};\bar{z})] \end{aligned} \tag{11}$$

This operator closely matches Bellantoni and Cook's predicative recursion on notations: if x *belongs to an even interval, we apply* h_0 *to its predecessor* $p(x;)$*; if* x *belongs to an odd interval, we apply* h_1 *to* $p'(x;) = \lfloor x/2 \rfloor$.

This class $\mathcal{W}$ is based on the Bellantoni-Cook's constructions and normal/safe arguments ideas in order to have the following properties, proved by induction.

Proposition 11. *1. Class* $\mathcal{W}$ *preserves the integers, that is for every* $f \in \mathcal{W}$, $f \restriction \mathbb{Z}\colon \mathbb{Z} \to \mathbb{Z}$.
2. Every polytime computable discrete function has a peaceful extension in $\mathcal{W}$.
3. Every function in $\mathcal{W}$ *is polytime computable.*

The proposition indicates that $\mathcal{W}$ is a class of polytime computable real functions that approximates polytime computable discrete functions. Hence, using Theorem 1 the following result is obtained.

Theorem 3. *A Lipschitz function* $f : [0,1] \to \mathbb{R}$ *is polytime computable iff it is* $\mathcal{W}$*-definable.*

Additionally, the previous proposition implies that any function in $\mathcal{W}$ has polytime computable integer approximation, hence using Corollary 1, we can get the following result.

Theorem 4. *Let* $f : [0,1] \to \mathbb{R}$ *be some* n^k*-smooth function for some* k. *Then* f *is polytime computable iff it is* n^k*-*$\mathcal{W}$*-definable.*

Notice that $\mathcal{C}$-definability of a function can be seen as a schema that builds a function f from a function $\tilde{g} \in \mathcal{C}$ (see definition of $\mathcal{C}$-definability). Hence, the class of polytime computable functions can be algebraically characterized in a machine-independent way as follows.

Corollary 2. *Let* $Def[\mathcal{C}]$ *stand for* $\mathcal{C}$*-definability. Then a function* $f\colon [0,1] \to \mathbb{R}$ *is polytime computable iff either (1)* f *is Lipschitz and belongs to* $Def[0,1,+,-,U, p, c, parity; SComp, SI]$ *or (2)* f *is* n^k*-smooth and belongs to* n^k*-*$Def[0,1,+,-,U, p, c, parity; SComp, SI]$.

[11] Notice also that for simplicity we misuse the basic functions (and p') so that their arguments are now in normal positions (the alternative is to redefine a new set of basic functions with arguments in normal positions).

Remark 1. It follows from our constructions that one could have put $*(;x,y) = xy$ as a basic function, from which $c(;x,y,z) = +(;*(;x,y),*(;-(;1,x),z))$ would be definable. In the same spirit adding π, $1/\pi$, $\sin(;x) = \sin(x)$, and $\max(;x,y) = max(x,y)$ would yield $parity(;x)$.

Remark 2. $parity(;x)$ can be replaced by any function with above mentioned properties.

References

1. L. Blum, F. Cucker, M. Shub, S. Smale, Complexity and Real Computation, Springer, 1998.
2. L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bull. Amer. Math. Soc. 21 (1) (1989) 1–46.
3. C. E. Shannon, Mathematical theory of the differential analyzer, J. Math. Phys. MIT 20 (1941) 337–354.
4. C. Moore, Recursion theory on the reals and continuous-time computation, Theoret. Comput. Sci. 162 (1) (1996) 23–44.
5. A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc. 2 (42) (1936) 230–265.
6. A. Grzegorczyk, Computable functionals, Fund. Math. 42 (1955) 168–202.
7. D. Lacombe, Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles III, C. R. Acad. Sci. Paris 241 (1955) 151–153.
8. O. Bournez, M. L. Campagnolo, D. S. Graça, E. Hainry, Polynomial differential equations compute all real computable functions on computable compact intervals, J. Complexity 23 (3) (2007) 317–335.
9. O. Bournez, E. Hainry, Recursive analysis characterized as a class of real recursive functions, Fund. Inform. 74 (4) (2006) 409–433.
10. O. Bournez, M. L. Campagnolo, A survey on continuous time computations, in: S. Cooper, B. Löwe, A. Sorbi (Eds.), New Computational Paradigms. Changing Conceptions of What is Computable, Springer, New York, 2008, pp. 383–423.
11. E. Asarin, O. Maler, A. Pnueli, Reachability analysis of dynamical systems having piecewise-constant derivatives, Theoret. Comput. Sci. 138 (1) (1995) 35–65.
12. K. Ruohonen, Event detection for ODEs and nonrecursive hierarchies, in: Proceedings of the Colloquium in Honor of Arto Salomaa. Results and Trends in Theoretical Computer Science (Graz, Austria, June 10-11, 1994), Vol. 812 of Lecture Notes in Comput. Sci., Springer, Berlin, 1994, pp. 358–371.
13. K.-I. Ko, Complexity Theory of Real Functions, Birkhäuser, 1991.
14. D. S. Graça, M. L. Campagnolo, J. Buescu, Robust simulations of Turing machines with analytic maps and flows, in: S. B. Cooper, B. Löwe, L. Torenvliet (Eds.), CiE 2005: New Computational Paradigms, Vol. 3526 of Lecture Notes in Comput. Sci., Springer, 2005, pp. 169–179.
15. M. L. Campagnolo, C. Moore, J. F. Costa, An analog characterization of the Grzegorczyk hierarchy, J. Complexity 18 (4) (2002) 977–1000.
16. M. Hofmann, Type systems for polynomial-time computation, habilitation thesis (1999).
17. N. D. Jones, The expressive power of higher-order types or, life without CONS, J. Funct. Programming 11 (1) (2001) 5–94.

18. J.-Y. Marion, J.-Y. Moyen, Efficient first order functional program interpreter with time bound certifications, in: LPAR, Vol. 1955 of Lecture Notes in Comput. Sci., Springer, 2000, pp. 25–42.
19. P. Clote, Computational models and function algebras, in: E. R. Griffor (Ed.), Handbook of Computability Theory, North-Holland, Amsterdam, 1998, pp. 589–681.
20. A. Cobham, The intrinsic computational difficulty of functions, in: Y. Bar-Hillel (Ed.), Proceedings of the International Conference on Logic, Methodology, and Philosophy of Science, North-Holland, Amsterdam, 1965, pp. 24–30.
21. S. Bellantoni, S. Cook, A new recursion-theoretic characterization of the polytime functions, Comput. Complexity 2 (1992) 97–110.
22. M. L. Campagnolo, K. Ojakian, The methods of approximation and lifting in real computation, in: Computability and Complexity in Analysis (CCA 2006), Vol. 167 of Electron. Notes Theor. Comput. Sci., 2007, pp. 387–423.
23. R. Constable, Type two computational complexity, in: Proc. fifth annual ACM symposium on Theory of computing, 1973, pp. 108–121.
24. B. M. Kapron, S. A. Cook, A new characterization of type-2 feasibility, SIAM J. Comput. 25 (1) (1996) 117–132.
25. V. Brattka, Computability over topological structures, in: S. B. Cooper, S. S. Goncharov (Eds.), Computability and Models, Kluwer Academic Publishers, New York, 2003, pp. 93–136.
26. K. Weihrauch, Computable Analysis: an Introduction, Springer, 2000.
27. W. Gomaa, Characterizing polynomial time computability of rational and real functions, in: Barry Cooper and Vincent Danos (Ed.), Proceedings of DCM 2009, Vol. 9 of EPTCS, 2009, pp. 54–64.
28. D. S. Graça, J. F. Costa, Analog computers and recursive functions over the reals, J. Complexity 19 (5) (2003) 644–664.

Incompleteness in Multimodal Logics: a Barrier for Quantum Computing?

Juliana Bueno-Soler[1] and Walter Carnielli[2]

[1] Center for Natural and Human Sciences - CCNH
Federal University of ABC
Santo André, SP, Brazil juliana.bueno@ufabc.edu.br,
WWW home page: http://lattes.cnpq.br/5824391030945544
[2] GTAL/CLE and Department of Philosophy
State University of Campinas
P.O. Box 6133, 13083-970
Campinas, SP, Brazil walter.carnielli@cle.unicamp.br,
WWW home page: http://www.cle.unicamp.br/prof/carnielli

Abstract. We show that some classes of multi-modal paraconsistent logics endowed with some weak forms of negation are incompletable with respect to Kripke semantics. We argue that this shortcoming, more than just a logical predicament, may be relevant for the attempts to model quantum information in (multi)modal logical terms. However, such incompleteness in principle does not affect the *modal possible-translations semantics*, which may be a way out of the incompleteness jungle.

1 Quantic Phenomena, Negation and Modality

Three-quarters of a century after Birkhof and von Neumann's proposal (cf. [1]) of a logical system able to make comprehensible (from the classical viewpoint) some 'paradoxes' of quantum mechanics, logicians, philosophers and computer scientists are still dealing with the question. In the meantime, paraconsistent logic has emerged, and it is considered by many that regarding the roots of a quantum deductive system from a paraconsistent perspective represents a clear mathematical and philosophical advantage from several viewpoints.

The well-known proof-theorist G. Takeuti in [2] had already adverted that 'quantum logic is drastically different from the classical logic or the intuitionistic logic." Thirty years later we do not know how drastic the difference is, but many evidences point to the kinship between quantum reasoning (the logic of orthomodular lattices) and paraconsistent deduction.

H. Aoyama has shown in [3] that such a kinship can be rigorously supplied: he proved, by syntactical means, that quantum logic is related to a paraconsistent logic (in the form of a dual intuitionistic logic which he calls **DI**; see also [4]). The question of the duality between the intuitionistic and the paraconsistent paradigms of thought has been object of concern since years; in [5] several classes of anti-intuitionistic logics are defined and compared with familiar paraconsistent calculi. It is proven that, although a duality between intuitionistic

and paraconsistent reasoning archetypes subsist, this duality can be carried out up to a certain limit only.

The connection between the quantum and the paraconsistent paradigms is not accidental. The paraconsistent Turing machines, studied in [6], allow for a partial simulation of superposed states of quantum computing, and in some cases entangled states and relative phase. Such machines permit to define paraconsistent algorithms which solve (under certain restrictions) the well-known Deutsch's and Deutsch-Jozsa problems.

On the other hand, it is well accepted that traditional quantum logic has connections to classical modal logic as the system the so-called Brouwerian system **B** and its extensions. As argued in [7], by considering not only propositions in quantum logic as expressing results of quantum experiments, but including propositions about the *possibility* of results of experiments, there is a quite interesting translation between the quantum propositional logic **QP** and an extension of **B**.

In a more usual notation (cf. e.g. [8]) the system **B** is axiomatized by the following modal schemas:

PC All the theorems of the Propositional Calculus **PC**
(K) $\Box(p \supset q) \supset (\Box p \supset \Box q)$
(T) $\Box p \supset p$
(B) $p \supset \Box\Diamond p$

closed under the following derivation rules:

(US) Uniform Substitution: for each variable p and sentense β, if $\vdash \alpha$ then $\vdash \alpha[p/\beta]$
(MP) Modus Ponens: q is deducible from p and $p \supset q$;
(Nec) Necessitation: if $\vdash p$, then $\vdash \Box p$.

To this system a new rule is added, defining the system $\mathbf{B}^+$:

$$\Box p \supset \Box q, \Box q \supset \Diamond\Box p \vdash \Box q \supset \Box p$$

.

Now, the quantum propositional logic **QP** is the logic of the propositions built from connectives $\wedge$ (conjunction) and $\backsim$ (orthomodular negation) valid in all orthomodular lattices L. A translation + is defined in [7] between the quantum propositional logic **QP** and $\mathbf{B}^+$ as follows:

$\alpha^+ = \Box\alpha$
$(\alpha \wedge \beta)^+ = \alpha^+ \wedge \beta^+$
$(\backsim \alpha)^+ = \Box\neg\alpha^+$

where $\neg$ is the usual Boolean negation of normal modal logics.

It can be shown (theorems 1 and 2 in [7] that the mapping + acts as a strong translation between **QP** and $\mathbf{B}^+$, in the sense that (for $\Gamma \cup \{\alpha\}$ in the language of **QP**) it holds:

$$\Gamma \vdash_{QP} \alpha \text{ iff } \Gamma^+ \vdash_{\mathrm{BR}+} \alpha^+$$

This translation permits the propositions of the modal system $\mathbf{B}^+$ which are images of sentences of **QP** by the translation + to be interpreted as records of results of experiments with atomic objects. The mapping +, however, is not a bijective, and the interpretation of other propositions of $\mathbf{B}^+$ (not images of **QP**) remain vague or uninterpreted.

There is an intuitive reason why this translation from the quantum propositional logic **QP** (regarded as a logic of orthomodular lattices) to the modal logic $\mathbf{B}^+$ should work: the orthomodular negation $\backsim$, weaker than classical negation, can be understood as "it is not possible that" or "necessarily not that", and in this sense is loosely connected to intuitionistic logic, which has, on its turn, some connections to the Brouwerian system **B** (see, to this respect, note 5 to chapter 3, page 70, in [9]). These connections, even if somewhat loose, help to explain the rationale of translating $\backsim \alpha$ into $\Box\neg\alpha$.

This intricate relationship, we suggest, helps to substantiate Takeuti's warning in [2]. Indeed, quantum logic is drastically different from the classical logic, or from the intuitionistic logic alone: it is somehow the "logic of the duality" between intuitionism and paraconsistency. With such an understanding, the translation between the paraconsistent (or dual-intuitionistic) logic $\mathbf{DI}^+$ and the intuitonistic logic $\mathbf{LJ}^+$ defined by Aoyama (definition 2.9 in [3]) makes $\wedge$ to correspond dually to $\vee$, and $\forall$ to correspond dually to $\exists$, while paraconsistent (material) implication $\supset$ corresponds dually to the intuitionistic pseudo-difference operator $\dot{-}$.

This means, as pointed out in [3], that the "dual" algebra of a complete Heyting algebra (i.e., the complete Brouwerian algebra) is not a proper model for the paraconsistent logic **DI**. This is in line with the well-known difficulties in characterizing the proper algebraic counterparts of paraconsistent logic (to this respect, an innovative proposal is given in [10] and sharpened in [11]).

The concept of quantum logic, in this way, not only boasts a modal character but has been extended to multi-modal frameworks, as in [12]. From this perspective, obtaining logical properties such as completeness, finite model property and decidability turn out to be relevant issues.

Properties of quantum knowledge have been compared to epistemic properties of (group) knowledge, and even quantum entanglement can be regarded from a formal epistemic viewpoint: two particles (or systems) are entangled if they potentially carry, without any communication: non-trivial information about each other. But this requires taking logic seriously, and in particular the salient features of negation.

2 From no Negation to Degrees of Negation

The dependence of quantum phenomena from properties of negation is striking: for instance, traditional quantum logic is negation-free, but the collection of testable properties of a quantum system is not closed under classical negation (and neither under classical disjunction). Not only a weak (paraconsistent) kind of negation is needed to explain some aspects of quantum superposition (as

done in [6]), but stronger forms of negation, with modal flavor, are in order. The modal character of quantum phenomena is made still more salient when we observe, as sanctioned by the discussion of Section 1, that a "no" answer for a measurement of P does not establish the negation of P, but the *impossibility* of obtaining a "yes" in any measurement for P. Such a modal-like negation is conveyed, for instance, by the orthocomplement $\sim P$ of P (as part of the theory of non-distributive, orthomodular lattices).

This naturally leads to a multimodal epistemic logics, which can to characterize quantum properties with computational contents, such as entanglement, superposition, quantum gates, etc. So in [13], for instance, a logic for composite systems, joining ideas from traditional quantum logic with multimodal concepts is proposed; however, the resulting logic, although sound, is not shown to be complete. A natural question thus is: are sophisticated logics of this kind doomed to incompleteness?

The partnership between paraconsistent logics and modal logics is not new: see e.g. [14] (example 93) and specially [15] for discussions and references. A systematization of the construction of classes of cathodic (with weak negations) and anodic (purely positive) modal systems is done in [15], where it is shown that these classes are semantically characterizable in two different ways: by means of Kripke-style semantics, and also by means of modal possible-translations semantics.

However, as we show here, modal extensions of cathodic systems $\mathbf{PI}^{k,l,m,n}$, as the system **PIVB**, cannot be semantically characterized by means of Krypke-style semantics.

A *propositional language* for a system $\mathbf{S}$ is composed by an infinite set Var of sentential variables p, q, r and so on, and operators in the set $\Sigma = \{\supset, \wedge, \Box, \Diamond, \neg, \circ\}$. The special connective $\circ$ plays a crucial role in paraconsistent logics, as it expresses the notion of *consistency* of a formula in the object-language level (more details in [14]). Although the system in the class $\mathbf{PI}^{k,l,m,n}$ we treat here do not contain $\circ$ in the language, it is convenient to mention this connective since is appears in most of the systems in [15].

The collection For of *sentences* of $\mathbf{S}$ is defined as usual in modal logics. The elements of For are represented by lowercase Greek letters α, β, γ, and subsets of For are represented by uppercase Greek letters Γ, Δ, Π. When necessary, the collection of sentences will be denoted by $For_{\mathbf{S}}$ instead of For only. Consider $\Gamma \cup \{\alpha\} \subseteq For$ and let $\vdash \subseteq \wp(For) \times For$ be a *consequence relation*, where $\wp(For)$ is the power set of the set For.

In systems containing the consistency operator $\circ$, as discussed in [14], a form of classical negation can be defined, usually called *strong negation*, defined as

$$\sim\alpha \stackrel{\text{Def}}{=} \alpha \supset [p \wedge (\neg p \wedge \circ p)]$$

From this definition all the relevant properties of classical negation are derivable, what is useful to show several expected metamathematical results which depend upon negation.

A useful notion is that of a *bi-valuation function* $v : For \longrightarrow \{0, 1\}$, where 1 denotes the "true" value and 0 denotes the "false" value:

(*Biv*.1)	$p \in Var$	implies	$v(p) = 1$ or $v(p) = 0$;
(*Biv*.2)	$v(\alpha \supset \beta) = 1$	iff	$v(\alpha) = 0$ or $v(\beta) = 1$;
(*Biv*.3)	$v(\alpha \wedge \beta) = 1$	iff	$v(\alpha) = 1$ and $v(\beta) = 1$;
(*Biv*.4)	$v(\alpha) = 0$	implies	$v(\neg\alpha) = 1$;
(*Biv*.5)	$v(\circ\alpha) = 1$	implies	$v(\alpha) = 0$ or $v(\neg\alpha) = 0$;
(*Biv*.6)	$v(\neg\neg\alpha) = 1$	implies	$v(\alpha) = 1$;
(*Biv*.7)	$v(\neg \circ \alpha) = 1$	implies	$v(\alpha) = 1$ and $v(\neg\alpha) = 1$.

Such conditions on valuations permit us to obtain completeness results w.r.t. bi-valuations for each paraconsistent system, endowed with the operator $\circ$. Another semantic characterization for the paraconsistent systems **PI**, **mbC**, **bC** and **Ci** can be attained w.r.t. possible-translations semantics, as discussed in [14]). From the viewpoint of combination of logics, the cathodic systems could be seen as a result of fusion (a particular case of fibring) between modal logic and non-modal logic as discussed in [16]. Several results about preservation of completeness in fibring have been obtained, but in all cases classical negation (instead of a paraconsistent negation) is involved. But such preservation results cannot be applied when negation is not strong enough, which makes room for incompleteness, as we show here.

3 Incompleteness and Degrees of Negation

As cathodic systems in the class $\mathbf{PI}^{k,l,m,n}$ cannot define any form of classical negation, a different treatment of such systems is required since they are intrinsically bi-modal systems (i.e., modalities cannot be inter-defined). This characteristic permit us to obtain an incompleteness result in this class. Whether or not, however, incompleteness results can be attained to other classes of cathodic systems is left as an open problem.

In the sequel it will be shown that the system **PIVB**, obtained by extending the system $\mathbf{PI}^{0,0,0,0}$ with the axiom (**VB**) is an incomplete system.

Let **PIVB** be the system obtained from $\mathbf{PI}^{0,0,0,0}$ by adding van Benthem's axiom:

(**VB**) $\quad \Diamond\Box p \vee \Box[\Box(\Box q \supset q) \supset q]$

The strategy of the argument is to show that the class of frames adequate for **PIVB** also validates a non-theorem of **PIVB**. This is shown by means of *general frames*. Consider the following sentence:

(**MV**) $\quad \Box p \vee \Diamond\Box p$

The next result shows that all frames that validate (**VB**) also validate (**MV**).

Lemma 1. *If* $\mathcal{F} \Vdash \Diamond\Box p \vee \Box[\Box(\Box q \supset q) \supset q]$ *then* $\mathcal{F} \Vdash \Diamond\Box p \vee \Box p$.

Proof. The same argument used for van Benthem's system (see the original proof in [17], or e.g. lemma 5.1.1 of [8]). □

The following definition is a generalization of the notion of general frame used by van Benthem in [17].

Definition 1. *A general frame is a triple* $\mathfrak{G} = \langle W, R, \Pi \rangle$ *where* $\mathfrak{F} = \langle W, R \rangle$ *is a non-trivial relational frame and* Π *is any collection of subsets of* W *called admissible sets closed under the following operations:*
(a) If $X \in \Pi$ *then* $\overline{X} \in \Pi$*;*
(b) If $X, Y \in \Pi$ *then* $\overline{X} \cup Y \in \Pi$*;*
(c) If $X, Y \in \Pi$ *then* $X \cap Y \in \Pi$*;*
(d) If $X \in \Pi$ *then* $\{w \in W : \forall w' \in W(wRw' \text{ implies } w' \in X)\} \in \Pi$*;*
(e) If $X \in \Pi$ *then* $\{w \in W : \exists w' \in W(wRw' \text{ and } w' \in X)\} \in \Pi$*.*

The following particular general frame $\mathfrak{G}_0 = \langle W_0, R_0, \Pi_0 \rangle$ will be helpful, defined as:

- $W_0 = \mathbb{N} \cup \{\omega, \omega + 1\}$
- $w_i R_0 w_j$ iff $\begin{cases} w_i = \omega + 1 \text{ and } w_j = \omega \\ w_i \neq \omega + 1 \text{ and } w_j < w_i \end{cases}$
- The collection Π_0 of admissible subsets of W_0 is specified in the following way:
  (a) $\omega \notin A$ and A is finite;
  (b) $\omega \in A$ and the complement of A is finite.

Let $V : Var \longrightarrow \wp(W)$ be an *implicit valuation*, where for each variable p, $V(p)$ represents the set of worlds in which p is an element.

Definition 2. *A model* $\mathfrak{M}$ *is called admissible if, for each* α*,* $V(\alpha)$ *is an admissible set.*

Lemma 2. $\mathfrak{G}_0$ *is a general frame.*

Proof. We need to show that Π_0 satisfies the conditions of Definition 1. The proof of clauses (b)–(e) appears in lemma 6.4 of [18]. It remains to show clause (a), i.e., $A \in \Pi_0$ implies $\overline{A} \in \Pi_0$. Supposing $A \in \Pi_0$, we need to show that $\omega \in \overline{A}$ and that $\overline{\overline{A}}$ is a finite set.

There are two possibilities to be considered:

1. Suppose A is a finite and that $\omega \notin A$.
   If $\omega \notin A$ then $\omega \in \overline{A}$. It follows that $\overline{\overline{A}}$ is a finite set. Since $\overline{\overline{A}} = A$ then, by hypothesis, we have that $\overline{\overline{A}}$ is a finite set.
2. Suppose $\overline{A}$ is a finite set and $\omega \in A$.
   Clearly, from the hypothesis $\omega \notin \overline{A}$ and $\overline{A}$ is a finite set.

Therefore, in both cases $\overline{A} \in \Pi_0$. □

A sentence α is said to be $\mathfrak{G}_0$-*valid* if α is valid in all admissible models on $\mathfrak{G}_0$. The next lemma specifies the conditions for a model to be considered admissible.

Lemma 3. *A model $\mathfrak{M}$ based on $\mathfrak{G}$ is admissible if $V(p)$ is admissible for all variables p.*

Proof. We need to show that, if $V(p)$ is an admissible set, then $V(\alpha)$ is also an admissible set, where α is a formula. The proof is by induction on the (usual) complexity of α, and the argument uses Definition 1. □

The next theorem shows that (**MV**) is not valid in the particular general frame $\mathfrak{G}_0$, which means that the system **PIVB** cannot be characterized by any class $\mathcal{F}$ of frames, since this class also validates a non-theorem of **PIVB**, as shown in Lemma 1.

Theorem 1. *(**MV**) is not $\mathfrak{G}_0$-valid.*

Proof. Let $\mathfrak{M}_0$ be the model based on $\mathfrak{G}_0$ at which $V(p) = \varnothing$, i.e., p is false in all $w \in W_0$. It is clear that $\mathfrak{M}_0$ is admissible: indeed, since $\varnothing \in \Pi_0$ and $\omega \notin \varnothing$ and $\varnothing$ is finite, clause (a) of the definition of admissible sets is satisfied. It is easy to check that $v(\Diamond\Box p \vee \Box p, \omega + 1) = 0$, hence $\mathfrak{M}_0 \not\models \Diamond\Box p \vee \Box p$. Therefore $\mathfrak{G}_0 \not\models$ (**MV**). □

To finish, it remains to be shown that **PIVB** is an incomplete system, i.e., we need to show that the axiom (**PI**) is valid in $\mathfrak{G}_0$.

Theorem 2. *Any theorem of **PIVB** is $\mathfrak{G}_0$-valid.*

Proof. The only case that needs to be considered is the axiom (**PI**). Details on other cases can be found in theorem 6.7 of [18]. Indeed, it can be readily shown that for each $w \in \mathbb{N} \cup \{\omega, \omega + 1\}$, $v((\mathbf{PI}), w) = 1$. Supposing that $v((\mathbf{PI}), w) = 0$ a contradiction appears in the world w, since there is no modalities involved. □

We can thus obtain the incompleteness result for **PIVB** with respect to the intended class of frames:

Theorem 3. ***PIVB** is an incompletable modal cathodic system with respect to Kripke semantics.*

Proof. On the one hand, Lemma 1 guarantees that any frame that validates (**VB**) also validates (**MV**). On the other hand, Theorem 2 grants that the model $\mathfrak{M}_0$ based on a general frame $\mathfrak{G}_0$ validates all theorems of **PIVB**, while Theorem 1 shows that this same model invalidates (**MV**), i.e., this shows that the sentence (**MV**) can be neither a theorem of **PIVB**, nor of any of its extensions. Therefore no frame can characterize (an extension of) **PIVB**, since each frame will also validate a non-theorem (**MV**). □

4 Afterword

The notion of *possible-translations semantics* introduced in the nineties for paraconsistent logics (see [19] for a revised approach) is an inspiration for the so-called exogenous approach to quantum logic proposed in [20]. The exogenous approach has subtle (but very relevant) distinction in comparison to the (Kripkean) possible-worlds approach to quantum logic (see also [21]). From this perspective, the possible-translations semantics has an inherent interrelation to quantum reasoning.

On the other hand, from a purely logic viewpoint, the incompleteness result (Theorem 3) is interesting in at least two aspects. Firstly, since the system **PIVB** is an extension of $\mathbf{PI}^{0,0,0,0}$, proven in [15] to be complete w.r.t. bi-valued relational models. Therefore the incompleteness result maintains a parallel with van Benthem's result in [17], in the sense of obtaining incomplete system which extends a complete one. A second, more pertinent aspect, concerns (a very plausible) immunity of the second semantics, the *modal possible-translations semantics*, with respect to incompleteness. Although we have no proof of such an immunity, it is conceivable to expect that no incompleteness result with respect to modal possible-translations semantics could be obtained, since this kind of semantics is very general.

An open question is whether other incompleteness results could be obtained starting from other classes of cathodic systems, where a form of classical negation is definable. The difficulty to obtain an incomplete result for those classes (following van Benthem's method and our generalizations) concerns the fact that their language includes formulas of the kind $\circ\alpha$ and the complications of defining appropriate notions of admissible sets involving this kind of formulas.

Are there other suitable methods in such cases? Any answer to such questions, positive or even negative, would be illuminating, and specially relevant to the efforts of devising multimodal epistemic logics aimed to characterize quantum information.

What this means is that an insistence on expecting quantum logic to behave in conformity with traditional modalities, with their perspicuous threats of incompleteness with regard to Kripke semantics may be unreasoned (not forgetting that an important result by Goldblatt in [22] shows that orthomodularity is not first-order definable).

But there are other ways of maintaining the modal kinship between quantum and modal reasoning, and at the same time the triple-sided affinity among quantum logic, paraconsistent logic and intuitionistic logic: it is possible to define a paraconsistent (cathodic) version of the modal Brouwerian system **B**. The axioms for the intuitionistic-paraconsistent system **KTBCi** are the following (with the usual derivation rules of Necessitation (**Nec**), Modus Ponens (**MP**) and Uniform Substitution (**US**)):

(A1) $p \supset (q \supset p)$
(A2) $(p \supset q) \supset [(p \supset (q \supset r)) \supset (p \supset r)]$
(A3) $(p \supset r) \supset [((p \supset q) \supset r) \supset r]$
(A4) $p \supset [q \supset (p \wedge q)]$
(A5) $(p \wedge q) \supset p$
(A6) $(p \wedge q) \supset q$
(PI) $(p \vee \neg p)$
(mbC) $\circ p \supset [p \supset (\neg p \supset q)]$
(bC) $\neg\neg p \supset p$
(Ci) $\neg \circ p \supset (p \wedge \neg p)$
(K) $\Box(p \supset q) \supset (\Box p \supset \Box q)$
(T) $\Box p \supset p$
(B) $p \supset \Box\Diamond p$

KTBCi is semantically complete with respect to Kripke semantics, and also with respect to modal possible-translations semantics (cf. [15]). An analogous extension from **B** to $\mathbf{B}^+$, as done in [7], could be thought here (although completeness in this case would be unknown; perhaps the extended system would doomed to incompleteness as well).

The system **PIVB** being so elementary, incompleteness results of the kind obtained in Theorem 3 may indeed represent barriers to logical expressibility of quantum flows of information, and thus to quantum computation; it is to be remarked that incompleteness results for anodic (i.e., purely positive) modal logics have also been obtained in [23]. Although incompleteness in the case of negationless modal logics is categorical, the case of systems endowed with some degree of negation as $\mathbf{PI}^{k,l,m,n}$ of cathodic may be rescued, as remarked, by means of the modal possible-translations semantics (as in [15], but see [14] for discussions and for historical references). How high this barrier can be we do not know.

References

1. G. Birkhoff and J. von Neumann, "The logic of quantum mechanics," *Annals of Mathematics*, vol. 37, pp. 823–843, 1936.
2. G. Takeuti, "Quantum set theory," in *Current Issues in Quantum Logic* (E. M. I. S. S. P. Sciences, ed.), pp. 303–322, New York: Plenum, 1981.
3. H. Aoyama, "LK, LJ, dual intuitionistic logic, and quantum logic," *Notre Dame Journal of Formal Logic*, vol. 45, no. 4, pp. 193–213, 2004.
4. M. L. D. Chiara and R.Giuntini, "Paraconsistent ideas in quantum logic," *Synthese*, vol. 125, no. 1-2, pp. 55–68, 2000.
5. A. B. Brunner and W. A. Carnielli, "Anti-intuitionism and paraconsistency," *Journal of Applied Logic*, vol. 3, no. 1, pp. 161–184, 2005.
6. J. C. Agudelo and W. A. Carnielli, "Paraconsistent machines and their relation to quantum computing," *Journal of Logic and Computation*, vol. 20, no. 2, pp. 573–595, 2010.
7. H. Dishkant, "Imbedding of the quantum logic in the modal system of brower," *The Journal Of Symbolic Logic*, vol. 42, no. 3, pp. 321–328, 1977.
8. W. A. Canielli and C. Pizzi, *Modalities and Multimodalities*. Springer, 2008.

9. G. Hughes and M. Cresswell, *A New Introduction to Modal Logic*. Routledge, 1996.
10. J. Bueno-Soler and W. A. Carnielli, "Possible-translations algebraization for paraconsistent logics," *Bulletin of the Section of Logic*, vol. 34, no. 2, pp. 77–92, 2005. Preprint available at *CLE e-Prints*, vol. 5, n. 6, 2005. `http://www.cle.unicamp.br/e-prints/vol\_5,n\_6,2005.html`.
11. J. Bueno-Soler, M. E. Coniglio, and W. A. Carnielli, "Possible-translations algebraizability," in *Handbook of Paraconsistency* (D. G. J.-Y. Béziau, W. A. Carnielli, ed.), (England), pp. 321–340, College Publications, London, 2007.
12. K. Tokuo, "Extended quantum logic," *Journal of Philosophical Logic*, vol. 32, pp. 549–563, 2003.
13. A. Baltag and S. Smets, "Lqp: the dynamic logic of quantum information," *Mathematical Structures in Computer Science*, vol. 16, no. 3, pp. 491–525, 2006.
14. W. A. Carnielli, M. E. Coniglio, and J. Marcos, "Logics of formal inconsistency," in *Handbook of Philosophical Logic* (D. Gabbay and F. Guenthner, eds.), vol. 14, (Amsterdam), pp. 1–93, Springer-Verlag, 2007.
15. J. Bueno-Soler, "Two semantical approaches to paraconsistent modalities," *Logica Universalis*, vol. 4, no. 1, pp. 137–160, 2010.
16. W. A. Carnielli, M. E. Coniglio, D. Gabbay, P. Gouveia, and C. Sernadas, *Analysis and Synthesis of Logics*. Sringer, 2007.
17. J. van Benthem, "Two simple incomplete logics," *Theoria*, vol. 44, pp. 25–37, 1978.
18. J. Bueno-Soler, *Multimodalidades anódicas e catódicas: a negação controlada em lógicas multimodais e seu poder expressivo (Anodic and cathodic multimodalities: controled negation in multimodal logics and their expressive power)*. Ph.D Thesis, in Portuguese, IFCH-Unicamp, Campinas, Brazil, 2009.
19. W. A. Carnielli, "Possible-translations semantics for paraconsistent logics," in *Frontiers of Paraconsistent Logic: Proceedings of the I World Congress on Paraconsistency* (D. Batens, C. Mortensen, G. Priest, and J. P. van Bendegem, eds.), Logic and Computation Series, pp. 149–163, Baldock: Research Studies Press, King's College Publications, 2000.
20. P. Mateus and A. Sernadas, "Weakly complete axiomatization of exogenous quantum propositional logic," *Information and Computation*, vol. 204, no. 5, pp. 771–794, 2006.
21. R. Chadha, P. Mateus, A. Sernadas, and C. Sernadas, *Extending classical logic for reasoning about quantum systems*, pp. 325–372. Elsevier, 2009.
22. R. Goldblatt, "Orthomodularity is not elementary," *The Journal of Symbolic Logic*, vol. 49, no. 2, pp. 401–404, 1984.
23. J. Bueno-Soler, "Completeness and incompleteness for anodic modal logics," *Journal of Applied Non-classical Logics*, vol. 19, no. 3, pp. 291–310, 2009.

Memory Cost of Simulating Quantum Mechanics

Adán Cabello

Departamento de Física Aplicada II, Universidad de Sevilla, E-41012 Sevilla, Spain
adan@us.es
www.adancabello.com

Abstract. Simulating the predictions of quantum mechanics by means of hidden variable models requires that individual physical systems store a minimum amount of memory. We investigate the minimum memory required to simulate some specific predictions of quantum mechanics related to quantum nonlocality and contextuality. The required memory becomes larger than the information carrying capacity of the corresponding quantum system and the density of memory increases with the complexity of the system. This suggests a new approach to the problem of hidden variables in quantum mechanics, and provides a new insight into the reasons why quantum resources outperform classical ones.

Keywords: Contextuality, Entanglement, Nonlocality, Memory

1 Introduction

Some predictions of quantum mechanics (QM) cannot be reproduced either by local hidden variables (HV) models (those in which the results of local measurements may not depend on spacelike separated events) [1, 2] or by noncontextual HV models (those in which the results of measurements may depend on which other compatible observables are measured) [3–5]. However, QM can be reproduced with nonlocal and contextual HV models [6–8]. While the nonlocal communication cost for simulating quantum nonlocality has been extensively investigated [9, 10], so far no attention has been paid to another essential resource needed to simulate QM: Memory.

The purpose of this paper is to investigate what is the minimum classical memory required to simulate the predictions of QM for several simple scenarios wherein the predictions of QM force HV models to be contextual or nonlocal. All of these scenarios involve a finite number of possible measurements. The basic assumption is that any HV model of an individual physical system can be seen as a finite state machine that generates an output (the result of the measurement) based on its current state (the state of the HV) and input (the observable being measured); that is, by a k-input n-state Mealy automaton [11, 12]. A Mealy automaton consists of a sextuple $(\Sigma, \Gamma, S, s_0, \delta, \omega)$, where Σ is the input alphabet (the set of observables to be measured), Γ is the output alphabet (the possible outcomes of these measurements), $S = \{s_1, \ldots, s_n\}$ is a finite set of states (the set of states of the HV), s_1 is the initial state (an element of S); δ is the

state-transition, which is a function of the state and the input, $\delta : S \times \Sigma \to S$, and ω is the output function, which is a function of the state and the input, $\omega : S \times \Sigma \to \Gamma$. The memory needed for the automaton is therefore at least $\log_2 n$ bits. Here we are only interested in the memory necessary to identify the state the automaton is in. A realistic implementation of the automaton would require extra memory (for instance, to store the program), but this memory is fixed and independent of the number and type of quantum predictions to be simulated.

2 Memory cost of nonlocality

2.1 The Bell-CHSH inequality

We first investigate the memory required to simulate the maximum quantum violation of the Bell-Clauser-Horne-Shimony-Holt (Bell-CHSH) inequality [1, 2],

$$\beta \equiv \langle A_0 B_0 \rangle + \langle A_0 B_1 \rangle + \langle A_1 B_0 \rangle - \langle A_1 B_1 \rangle \leq 2, \tag{1}$$

where A_i are local observables on Alice's qubit, and B_i are local observables on Bob's qubit. All these observables have possible results -1 or $+1$, and Alice's (Bob's) choice of local measurement is assumed to be spacelike separated from Bob's (Alice's) result. The maximum quantum violation of inequality (1) is $\beta_{\rm QM} = 2\sqrt{2} \approx 2.83$ [13].

We assume that there is no restriction to the nonlocal communication between Alice and Bob's qubits, and that every pair of qubits can be described by a single four-input (A_0, A_1, B_0, and B_1) n-state Mealy automaton. We consider the possibility that some pairs of qubits can be described by one-state Mealy automata (that is, by local HV models), which require $\log_2 1 = 0$ bits of memory, while other pairs of qubits are described by nonlocal automata with a different number of states.

Result 1: $(\sqrt{2}-1)\log_2 3 \approx 0.66$ bits per pair suffice to simulate the maximum quantum violation of the Bell-CHSH inequality (1).

Proof: The maximum quantum violation of inequality (1), $\beta_{\rm QM} = 2\sqrt{2} \approx 2.83$, can be caused by a mixture of one-state Mealy automata giving $\beta = 2$, with probability $2 - \sqrt{2} \approx 0.59$, and three-state Mealy automata giving $\beta = 4$, with probability $\sqrt{2} - 1 \approx 0.41$. Therefore, the required average memory in bits per pair of qubits is

$$\mu_{\rm CHSH} = (\sqrt{2} - 1)\log_2 3 \approx 0.66. \tag{2}$$

The only way of obtaining $\beta = 4$ is the following: When Alice measures A_i and Bob measures B_j (whatever the order in which these measurements are carried out: first Alice and then Bob, or first Bob and then Alice), their results

satisfy

$$A_0 = B_0, \tag{3a}$$
$$A_0 = B_1, \tag{3b}$$
$$A_1 = B_0, \tag{3c}$$
$$A_1 = -B_1. \tag{3d}$$

In addition, it is natural to assume that the automaton satisfies repeatability, that is, if Alice measures A_i twice on the same qubit (with no other measurement in between), she will obtain the same result, and likewise for Bob. Therefore, sequences like B_i, A_j, A_j, B_i, A_j must satisfy both (3) and the fact that in every measurement the result of B_i (and A_j) turns out to be the same. These restrictions cannot be simulated either with one-state Mealy automata (that is, with local HV models) or with two-state Mealy automata (to prove it, check that none of the 32768 possible four-input two-state Mealy automata satisfies the conditions). However, it can be easily seen that the three-state automaton characterized by the following table satisfies both conditions:

$$\begin{array}{rcccc} & A_0 & A_1 & B_0 & B_1 \\ s_1 \equiv & +1 & +3 & +1 & +2 \\ s_2 \equiv & +1 & -2 & -2 & +2 \\ s_3 \equiv & -3 & +3 & +1 & -3 \end{array} \tag{4}$$

In (4) each line represents a state s_i of the automaton and each column contains the result of the measurement of the corresponding observable and the state of the automaton after that measurement: The result is given by the sign of the entry, and the state after the measurement by the absolute value of the entry. The automaton is assumed to be initially in the state s_1. For instance, if the measurements are A_1 (first) and (then) B_1, then the automaton gives the result $A_1 = +1$, then changes its state from s_1 into s_3, and then gives the result $B_1 = -1$ (and ends up in the state s_3). Note that we can make the marginal probabilities equal to $1/2$ for all inputs and outputs by suitably choosing different one-state Mealy automata with $\beta = 2$ and different three-state Mealy automata with $\beta = 4$ like (4). ■

2.2 Perpetual Popescu-Rohrlich boxes

Result 2: A bit of memory per pair of qubits suffices to simulate a Popescu-Rohrlich (PR) box [14] that does not require initialization.

Proof: A PR box satisfies the correlations (3) and does not allow signaling between Alice and Bob. These two requirements entail the marginal probabilities to be equal to $1/2$ for all inputs and outputs,

$$p(A_i = +1) = p(A_i = -1) = \frac{1}{2}, \tag{5a}$$
$$p(B_i = +1) = p(B_i = -1) = \frac{1}{2}, \tag{5b}$$

for $i \in \{0, 1\}$. The Mealy automaton (4) does not satisfy (5). This problem can be solved by preparing a suitable mixture of automata like (4), or by using the following four-state Mealy automaton which satisfies (3), repeatability, and (5):

$$\begin{array}{rcccc} & A_0 & A_1 & B_0 & B_1 \\ s_1 \equiv & +1 & +3 & +1 & +2 \\ s_2 \equiv & +1 & -2 & -4 & +2 \\ s_3 \equiv & -4 & +3 & +1 & -3 \\ s_4 \equiv & -4 & -2 & -4 & -3 \end{array} \tag{6}$$

The notation used in (6) is the same as in (4). Note that (6) simulates a PR box regardless of which of the four states s_i we choose as initial state. This means that the automaton does not require any initialization and can be used as a perpetual PR box. ■

2.3 Chained Bell inequalities

We can also use this method to obtain the memory cost of the maximum violation of any Bell inequality in which the maximum quantum violation does not saturate the maximum possible violation. For example, the bipartite N-setting Bell inequality of Braunstein and Caves (BC) [15, 16], in which Alice can choose one out of N alternative experiments A_1, A_3, ..., A_{2N-1}, and Bob one out of N alternative experiments B_2, B_4, ..., B_{2N}, each of them having outcomes $+1$ or -1,

$$\begin{aligned} \gamma \equiv & \langle A_1 B_2 \rangle + \langle B_2 A_3 \rangle + \langle A_3 B_4 \rangle + \langle B_4 A_5 \rangle + \cdots \\ & + \langle A_{2N-1} B_{2N} \rangle - \langle B_{2N} A_1 \rangle \leq 2N - 2. \end{aligned} \tag{7}$$

The maximum quantum violation of (7) is $\gamma_{\mathrm{QM}} = 2N \cos(\pi/2N)$ [17]. This violation can be caused by a mixture of local (one-state) automata giving $\gamma = 2N - 2$ with probability $p = N[1 - \cos(\pi/2N)]$ and (three-state, as can be easily checked) nonlocal automata giving $\gamma = 2N$ with probability $1 - p$. Therefore,

$$\mu_{\mathrm{BC}}(N) = N[1 - \cos(\pi/2N)] \log_2 3 \tag{8}$$

bits of memory per pair of qubits suffice to reproduce the maximum violation of the BC inequality (7). Note that $\mu_{\mathrm{BC}}(2) = \mu_{\mathrm{CHSH}}$ and that $\mu_{\mathrm{BC}}(N)$ grows with N and tends to $\log_2 3 \approx 1.58$ bits of memory per pair of qubits when N tends to infinity.

3 Memory cost of quantum contextuality

3.1 Quantum contextuality for a specific qutrit state

A similar method can be applied to calculate the memory cost of simulating the quantum violation of any noncontextual inequality (that is, any inequality

satisfied by any noncontextual theory). A specially important case is given by the violation of the Klyachko, Can, Binicioğlu, and Shumovsky (KCBS) inequality [18], which is the simplest noncontextual inequality violated by a single qutrit,

$$\kappa \equiv -\langle C_0 C_1\rangle - \langle C_1 C_2\rangle - \langle C_2 C_3\rangle - \langle C_3 C_4\rangle - \langle C_4 C_0\rangle \leq 3, \tag{9}$$

where C_i are observables with possible results -1 or $+1$ on a single qutrit system. The maximum quantum violation on a single qutrit system is $\kappa_{\rm QM} = 4\sqrt{5} - 5 \approx 3.94$ [19]. To test this inequality, the experimenter can measure C_i (first) and (then) C_{i+1} on a single qutrit initially prepared in a specific state.

Result 3: $(2\sqrt{5} - 4)\log_2 3 \approx 0.75$ bits per qutrit suffice to simulate the maximum quantum violation of the KCBS inequality (9).

Proof: The maximum quantum violation of inequality (9), $\kappa_{\rm QM} = 4\sqrt{5} - 5 \approx 3.94$, can be caused by a mixture of one-state Mealy automata giving $\kappa = 3$, with probability $5 - 2\sqrt{5} \approx 0.53$, and three-state Mealy automata giving $\kappa = 5$, with probability $2\sqrt{5} - 4 \approx 0.47$. Therefore, the average memory needed is

$$\mu_{\rm KCBS} = (2\sqrt{5} - 4)\log_2 3 \approx 0.75 \tag{10}$$

bits per qutrit.

The only way of obtaining $\kappa = 5$ is the following: When the experimenter measures C_i (first) and (then) C_{i+1}, or C_{i+1} and C_i, the results must satisfy

$$C_0 = -C_1, \tag{11a}$$
$$C_1 = -C_2, \tag{11b}$$
$$C_2 = -C_3, \tag{11c}$$
$$C_3 = -C_4, \tag{11d}$$
$$C_4 = -C_0. \tag{11e}$$

The following three-state automaton satisfies (11) and repeatability:

$$\begin{array}{rccccc} & C_0 & C_1 & C_2 & C_3 & C_4 \\ s_1 \equiv & +2 & -1 & +1 & -1 & +3 \\ s_2 \equiv & +2 & -2 & +1 & +2 & -2 \\ s_3 \equiv & -3 & +3 & +1 & -3 & +3 \end{array} \tag{12}$$

■

3.2 Perpetual Kochen-Specker boxes

Result 4: A bit of memory per pair suffices to simulate a Kochen-Specker-Kyachko (KS) box [20] that does not require initialization.

Proof: A KS box gives $\kappa = 5$ and does not allow signaling between two observers such that one of them chooses the first measurement and the other the second one. The following automaton satisfies these requirements for any initial

state:

$$\begin{array}{rccccc} & C_0 & C_1 & C_2 & C_3 & C_4 \\ s_1 \equiv & +1 & -1 & +1 & -2 & -3 \\ s_2 \equiv & -2 & +4 & +1 & -2 & +2 \\ s_3 \equiv & +3 & -1 & -4 & +3 & -3 \\ s_4 \equiv & -4 & +4 & -4 & +3 & +2 \end{array} \tag{13}$$

■

3.3 State-independent quantum contextuality

Now consider 15 dichotomic observables $XI, \ldots, ZZ$ with possible results -1 or $+1$. The following inequality containing 15 mean values must be satisfied by any noncontextual HV model:

$$\begin{aligned} \nu \equiv & \langle XI\ IX\ XX\rangle + \langle XI\ IY\ XY\rangle + \cdots + \langle ZI\ IZ\ ZZ\rangle \\ & + \langle XX\ YZ\ ZY\rangle + \langle XY\ YX\ ZZ\rangle + \langle XZ\ YY\ ZX\rangle \\ & - \langle XX\ YY\ ZZ\rangle - \langle XY\ YZ\ ZX\rangle - \langle XZ\ YX\ ZY\rangle \leq 9. \end{aligned} \tag{14}$$

However, for any initial state of a two-qubit system, if one chooses $XI = \sigma_x^{(1)} \otimes I^{(2)}, \ldots, ZZ = \sigma_z^{(1)} \otimes \sigma_z^{(2)}$, where $\sigma_x^{(1)} \otimes I^{(2)}$ denotes the tensor product of the X Pauli matrix of the first qubit times the identity matrix for the second qubit, then one obtains

$$\nu_{\mathrm{QM}} = 15, \tag{15}$$

which is the maximum possible violation of inequality (14) [21]. Even more interestingly, this is the first known example of a quantum prediction which, in order to be simulated by a HV model, requires an automaton *with more memory than the information carrying capacity of the corresponding quantum system* given by its Holevo's bound [28]. In other words, it can be proven [29] that no four-state automaton can simulate (15); any automaton requires

$$\mu > 2 \tag{16}$$

bits of memory per pair of qubits.

Inequality (14) is an extended version of the inequality proposed in [22], which has recently stimulated several experiments [23–27].

We can even go further and consider the natural extension to three qubits of the inequality (14) by considering all possible observables of the form $A \otimes B \otimes C$, where $A, B, C \in \{\sigma_x, \sigma_y, \sigma_z, I\}$, and all sets of four mutually compatible observables such that their product is $\pm I$ (here denoting the identity matrix of the Hilbert space of all three qubits). The quantum violation of this inequality requires *more memory per qubit* than that required to simulate (15). The reason is simple: Every subset of two qubits should satisfy (15) plus additional restrictions. Indeed, we can also consider the corresponding inequality for $4, 5, \ldots, n$ qubits and more memory per qubit is needed in every step. The required density of memory to reproduce QM *increases* with n.

4 Conclusions

The minimum density of memory (in bits per qubit) required for simulating the predictions of QM for a finite set of observables on a system of n qubits *grows* with n. This suggests a new proof of the impossibility of HV models in QM. If we assume that QM is correct and there is a bound for the density of memory a physical system can store, then QM must be complete, in the sense that no HV model can simulate the predictions of QM if the system is complex enough. We have a new physical basis on which to prove the impossibility of going beyond QM with HV theories, different than the assumption of locality. That is, we can prove that there are no more detailed extensions of QM, even if there is no upper bound to the velocity in which causal influences can propagate [30]. Moreover, this also points out that one of the reasons why quantum resources outperform classical ones is that a k-state quantum system can perform tasks which are beyond the reach of any k-state classical automaton.

Acknowledgments

The author acknowledges support from the Spanish MCI Project No. FIS2008-05596.

References

1. J.S. Bell, Physics (Long Island City, NY) **1**, 195 (1964).
2. J.F. Clauser, M.A. Horne, A. Shimony, and R.A. Holt, Phys. Rev. Lett. **23**, 880 (1969).
3. E.P. Specker, Dialectica **14**, 239 (1960).
4. J.S. Bell, Rev. Mod. Phys. **38**, 447 (1966).
5. S. Kochen and E.P. Specker, J. Math. Mech. **17**, 59 (1967).
6. D. Bohm, Phys. Rev. **85**, 166 (1952).
7. B.R. La Cour, Phys. Rev. A **79**, 012102 (2009).
8. A.Y. Khrennikov, *Contextual Approach to Quantum Formalism* (Springer, Berlin, 2009).
9. B.F. Toner and D. Bacon, Phys. Rev. Lett. **91**, 187904 (2003).
10. S. Pironio, Phys. Rev. A **68**, 062102 (2003).
11. G.H. Mealy, Bell Systems Technical J. **34**, 1045 (1955).
12. C.H. Roth Jr., *Fundamentals of Logic Design* (Thomson, Stanford, CT, 2009).
13. B.S. Tsirelson, Lett. Math. Phys. **4**, 93 (1980).
14. S. Popescu and D. Rohrlich, Found. Phys. **24**, 379 (1994).
15. S.L. Braunstein and C.M. Caves, in *Bell's Theorem, Quantum Theory, and Conceptions of the Universe*, edited by M. Kafatos (Kluwer, Dordrecht, 1989), p. 27.
16. S.L. Braunstein and C.M. Caves, Ann. Phys. (N.Y.) **202**, 22 (1990).
17. S. Wehner, Phys. Rev. A **73**, 022110 (2006).
18. A.A. Klyachko, M.A. Can, S. Binicioğlu, and A.S. Shumovsky, Phys. Rev. Lett. **101**, 020403 (2008).
19. P. Badziąg, I. Bengtsson, A. Cabello, H. Granström, and J.-Å. Larsson, Found. Phys. 10.1007/s10701-010-9433-3 (2010); arXiv:0909.4713.

20. J. Bub and A. Stairs, Found. Phys. **39**, 690 (2009).
21. A. Cabello, arXiv:1002.3135.
22. A. Cabello, Phys. Rev. Lett. **101**, 210401 (2008).
23. G. Kirchmair, F. Zähringer, R. Gerritsma, M. Kleinmann, O. Gühne, A. Cabello, R. Blatt, and C.F. Roos, Nature (London) **460**, 494 (2009).
24. H. Bartosik, J. Klepp, C. Schmitzer, S. Sponar, A. Cabello, H. Rauch, and Y. Hasegawa, Phys. Rev. Lett. **103**, 040403 (2009).
25. E. Amselem, M. Rådmark, M. Bourennane, and A. Cabello, Phys. Rev. Lett. **103**, 160405 (2009).
26. B.H. Liu, Y.F. Huang, Y.X. Gong, F.W. Sun, Y.S. Zhang, C.F. Li, and G.C. Guo, Phys. Rev. A **80**, 044101 (2009).
27. O. Moussa, C.A. Ryan, D.G. Cory, and R. Laflamme, Phys. Rev. Lett. **104**, 160501 (2010).
28. A.S. Holevo, Probl. Peredachi. Inf. **9**, 3 (1973) [Probl. Inf. Transm. (USSR) **9**, 177 (1973)].
29. M. Kleinmann, O. Gühne, J.R. Portillo, J.-Å. Larsson, and A. Cabello, arXiv:1007.3650.
30. A. Cabello, Found. Phys. (2010).

Experimental Evidence of Quantum Randomness Incomputability

Cristian S. Calude[1], Michael J. Dinneen[1], Monica Dumitrescu[2], and Karl Svozil[3]

[1] Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand
cristian@cs.auckland.ac.nz
[2] Faculty of Mathematics and Computer Science, University of Bucharest,Str. Academiei 14, 010014 Bucharest, Romania
monadumitrescu@gmail.com
[3] Institute for Theoretical Physics, University of Technology Vienna, Wiedner Hauptstrasse 8-10/136, 1040 Vienna, Austria

Abstract. In contrast with software-generated randomness (called pseudo-randomness), quantum randomness is provable incomputable, i.e. it is not exactly reproducible by any algorithm. We provide experimental evidence of incomputability — an asymptotic property — of quantum randomness by performing finite tests of randomness inspired by algorithmic information theory.

1 Quantum Indeterminacy

The irreducible indeterminacy of individual quantum processes postulated by Born [1–3] implies that there exist physical "oracles," which are capable to effectively produce outputs which are incomputable. Indeed, quantum indeterminism has been proved [4] under some "reasonable" side assumptions implied by Bell-, Kochen-Specker- and Greenberger-Horne-Zeilinger-type theorems. Yet, as quantum indeterminism is nowhere formally specified, it is important to investigate which (classes of) measurements lead to randomness, what are the reasons for possible distinctions, whether or not the kinds of randomness "emerging" in different classes of quantum measurements are "the same" or "different," and what are the phenomenologies or signatures of these randomness classes. Questions about "degrees of (algorithmic) randomness" are studied in algorithmic information theory. Here are just four types, among an infinity of others: (i) standard pseudo-randomness produced by software like *Mathematica* or *Maple* which are not only Turing computable but cyclic; (ii) pseudo-randomness produced by software which is Turing computable but not cyclic (e.g., digits of π, the ratio between the circumference and the diameter of an ideal circle, or Champernowne's constant); (iii) Turing incomputable, but not algorithmically random; (iv) algorithmically random [5–7]. One can ask: in which of these four classes do we find quantum randomness? Operationally, in the extreme form, Born's postulate could be interpreted to allow for the production of "random" finite

strings; hence quantum randomness could be of type (iv). (Here the quotation mark refers to the fact that randomness for finite strings is too "subjective" to be meaningful for our analysis. The legitimacy of the experimental approach comes from characterizations of random sequences in terms of the degrees of incompressibility of their finite prefixes. [5–7].) A sequence which is not algorithmically random but Turing incomputable can, for instance, be obtained from an algorithmically random sequence $x_1x_2\cdots x_n\cdots$ by inserting a 0 in between any adjacent original bits, i.e. obtaining the sequence $x_10x_20\cdots 0x_n0\cdots$ This transformation destroys algorithmic randomness because obvious correlations have appeared; Turing incomputability is invariant under this transformation because a copy of the original sequence is embedded in the new one. Yet much more subtler correlations among subsequences of Turing incomputable sequences may exist, thus making them compressible and algorithmically nonrandom. There is no *a priori* reason to interpret Born's indeterminism by its strongest formal expression; i.e., in terms of algorithmic randomness.

Quantum randomness produced by quantum systems which have no classical interpretation is provable [4] Turing incomputable. More precisely, if the experiment would run under ideal conditions "to infinity," the resulting infinite sequence of bits would be Turing incomputable; i.e., no Turing machine (or algorithm) could reproduce exactly this infinite sequence of digits. This result has many consequence; here is one example. The experiment could produce a billion of 0s, but not all bits produced will be 0. A stronger form of incomputability holds true: every Turing machine (or algorithm) can reproduce exactly only finitely many scattered digits of that infinite sequence. Yet this proof stops short of showing that the sequence produced by such a quantum experiment is algorithmically random; i.e., it is unknown whether or not such a sequence is or is not algorithmically random. One of the strategies toward answering this question is to empirically perform tests "against" the algorithmic randomness hypothesis.

Our (more modest) aim is to present tests capable of distinguishing computable from incomputable sources of "randomness" by examining (long, but) finite prefixes of infinite sequences. Such differences are guaranteed to exist by [4], but, because computability is an asymptotic property, there was no guarantee that finite tests can "pick" differences in the prefixes that we have analyzed.

2 Tests of Experimental Quantum Indeterminacy

Based on Born's postulate, several quantum random number generators based on beam splitters have recently been proposed and realized [8–15]. In what follows a detailed analysis of bit strings of length 2^{32} obtained by two such quantum random number generators will be presented — the first analysis of a set of quantum bits of this size (the size correlates well with the square root of the cycle length used by cyclic pseudo-random generators; randomness properties of longer strings generated in this way are impaired). We will compare the performance of quantum random number generators with software-generated number generators on randomness inspired by algorithmic information theory (which

complement some commonly used statistical tests implemented in "batteries" of test suites such as, for instance, *diehard* [16], *NIST* [17], or *TestU01* [18]). The standard test suites are often based on tests which are not designed for physical random number generators, but rather to quantify the quality of the cyclic pseudo-random numbers generated by algorithms. As we would like to separate "truly" random sequences from software-generated random sequences, the emphasis is on the former type of tests.

The tests based on algorithmic information theory directly analyze randomness, and thus the strongest possible form of incomputability. They differ from tests employed in the standard randomness batteries as they depend on irreducible algorithmic information content, which is constant for algorithmic pseudo-random sequences. Some tests are related to each other, as for instance sequences which are not Borel normal (cf. below) could be algorithmically compressed; the analysis of results helps understanding subtle differences at the edge of incomputability/algorithmic randomness. All tests depend on the size of the analyzed strings; the legitimacy of our approach is given by the fact that algorithmic randomness of an infinite sequence can be "uniformly read" in its prefixes (cf. [7]).

3 Data Sources

The analyzed quantum data consist of 10 quantum random strings generated with the commercially available *Quantis* device [19], based on research of a group in Geneva [11], as well as 10 quantum random strings generated by the *Vienna IQOQI* group [20]. The pseudo-random data consist of 10 pseudo-random strings produced by *Mathematica* 6 [21], and 10 pseudo-random strings produced by *Maple* 11 [22], as well as 10 strings of 2^{32} bits from the binary expansion of π obtained from the University of Tokyo's supercomputing center [23].

The signals of the *Quantis* device are generated by a light emitting diode producing photons which are then transmitted toward a beam splitter (a semi-transparent mirror) and two single-photon detectors (detectors with single-photon resolution) to record the outcomes associated with the symbols "0" and "1," respectively [19]. Due to hardware imbalances which are difficult to overcome at this level, Quantis processes this raw data by un-biasing the sequence by a von Neumann type normalization: The biased raw sequence of zeroes and ones is partitioned into fixed subsequences of length two; then the even parity sequences "00" and "11" are discarded, and only the odd parity ones "01" and "10" are kept. In a second step, the remaining sequences are mapped into the single symbols $01 \mapsto 0$ and $10 \mapsto 1$, thereby extracting a new unbiased sequence at the cost of a loss of original bits [24, p. 768].

This normalization method requires that the events are (temporally) uncorrelated and thus independent. (For the sake of a simple counterexample, the von Neumann normalization of the sequences $010101\cdots$ or $1100110011\cdots$ are the constant-0 sequence $000\cdots$ and the empty sequence.) Under the independence hypothesis, the normalized sequences are Borel normal with probability

one [25]; e.g., all finite subsequences of length n occur with their expected asymptotic frequencies 2^{-n}. (Alas, see [26] for some pitfalls when transforming such sequences.)

The signals of the Vienna Institute for Quantum Optics and Quantum Information (IQOQI) group were generated with photons from a weak blue LED light source which impinged on a beam splitter without any polarization sensitivity with two output ports associated with the codes "0" and "1," respectively [10]. There was *no* pre- or post-processing of the raw data stream, in particular no von Neumann normalization as discussed for the Quantis device; however the output was constantly monitored (the exact method is subject to a patent pending). In very general terms, the setup needs to be running for at least one day to reach a stable operation. There is a regulation mechanism which keeps track of the bias between "0" and "1," and tunes the random generator for perfect symmetry. Each data file was created in one continuous run of the device lasting over hours.

We have employed the *extended cellular automaton generator* default of *Mathematica* 6's pseudo-random function. It is based on a particular five-neighbor rule, so each new cell depends on five nonadjacent cells from the previous step [21]. *Maple* 11 uses a Mersenne Twister algorithm to generate a random pseudo-random output [22].

4 Testing Incomputability and Randomness

The tests we performed can be grouped into: (i) two tests based on algorithmic information theory, (ii) statistical tests involving frequency counts (Borel normality test), (iii) a test based on Shannon's information theory, and (iv) a test based on random walks.

In Figures 1–5 the graphical representation of the results is rendered in terms of box-and-whisker plots, which characterize groups of numerical data through five characteristic summaries: test minimum value, first quantile (representing one fourth of the test data), median or second quantile (representing half of the test data), third quantile (representing three fourths of the test data), and test maximum value. Mean and standard deviation of the data representing the results of the tests are calculated. Tables containing the experimental data and the programs used to generate the data can be downloaded from our extended paper [27].

4.1 Book stack randomness test

The *book stack* (also known as "move to front") test [28, 29] is based on the fact that compressibility is a symptom of less randomness.

The results, presented in Figure 1 and Table 1, are derived from the original count, the count after the application of the transformation, and the difference. The key metric for this test is the count of ones after the transformation. The book stack encoder does not compress data but instead rewrites each byte

with its index (from the top/front) with respect to its input characters being stacked/moved-to-front. Thus, if a lot of repetitions occur (i.e., a symptom of non-randomness), then the output contains more zeros than ones due to the sequence of indices generally being smaller numerically.

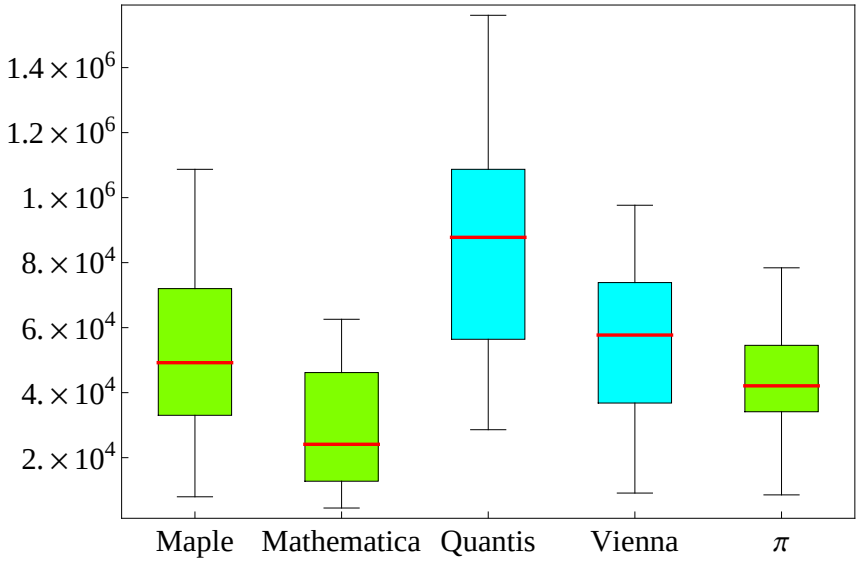


Fig. 1. (Color online) Box-and-whisker plot for the results of the "book stack" randomness test.

4.2 Solovay-Strassen probabilistic primality test

The second algorithmic test, based on the *Solovay-Strassen probabilistic primality test*, uses Carmichael (composite) numbers which are "difficult" to factor, to determine the quality of randomness by computing how fast the probabilistic primality test reaches the verdict "composite" [30, 31].

To test whether a positive integer n is prime, we take k natural numbers uniformly distributed between 1 and $n-1$, inclusive, and, for each chose i, check whether the predicate $W(i,n)$ holds. If this is the case we say that "i is a witness of n's compositeness". If $W(i,n)$ holds for at least one i then n is composite; otherwise, the test is inconclusive, but in this case if one declares n to be prime then the probability to be wrong is smaller than 2^{-k}.

This is due to the fact that at least half i's from 1 to $n-1$ satisfy $W(i,n)$ if n is indeed composite, and *none* of them satisfy $W(i,n)$ if n is prime [30]. Selecting k natural numbers between 1 and $n-1$ is the same as choosing a binary string s of length $n-1$ with k 1's such that the ith bit is 1 iff i is selected. Ref. [31]

Table 1. Statistics for the results of the "book stack" randomness test.

Descriptive statistics	min	Q1	median	Q3	max	mean	sd
Maple	7964	34490	49220	69630	108700	53410	33068.58
Mathematica	4508	13020	24110	43450	62570	27940	19406.03
Quantis	28600	60480	87780	106700	156100	89990	41545.76
Vienna	9110	38420	57720	73220	97660	53860	27938.92
π	8551	35480	42100	52870	78410	41280	20758.46

contains a proof that, if s is a long enough algorithmically random binary string, then n is prime iff $Z(s, n)$ is true, where Z is a predicate constructed directly from conjunctions of negations of W [4].

A Carmichael number is a composite positive integer k satisfying the congruence $b^{k-1} \equiv 1 (\mathrm{mod}\, k)$ for all integers b relative prime to k. Carmichael numbers are composite, but are difficult to factorize and thus are "very similar" to primes; they are sometimes called pseudo-primes. Carmichael numbers can fool Fermat's primality test, but less the Solovay-Strassen test. With increasing values, Carmichael numbers become "rare" [5].

We used the Solovay-Strassen test for all Carmichael numbers less than 10^{16}—computed in Ref. [32, 33]—with numbers selected according to increasing prefixes of each sample string till the algorithm returns a non-primality verdict. The metric is given by the length of the sample used to reach the correct verdict of non-primality for all of the 246683 Carmichael numbers less than 10^{16}. [We started with $k = 1$ tests (per each Carmichael number) and increase k until the metric goal is met; as k increases we always use new bits (never recycle) from the sample source strings.] The results are presented in Figure 2 and Table 2.

Table 2. Statistics for the results based on the Solovay-Strassen probabilistic primality test.

Descriptive statistics	min	Q1	median	Q3	max	mean	sd
Maple	93.0	96.0	101.0	113.5	120.0	104.9	10.57723
Mathematica	93.0	97.0	109.0	132.3	142.0	113.5	19.60867
Quantis	99.0	103.3	113.0	121.3	130.0	112.6	10.66875
Vienna	82.0	100.3	104.5	109.0	119.0	103.5	11.03781
π	84.0	91.8	106.0	110.8	128.0	104.7	10.66875

[4] In fact, every "decent" Monte Carlo simulation algorithm in which tests are chosen according to an algorithmic random string produces a result which is not only true with high probability, but *rigorously correct* [34].

[5] There are 1,401,644 Carmichael numbers in the interval $[1, 10^{18}]$.

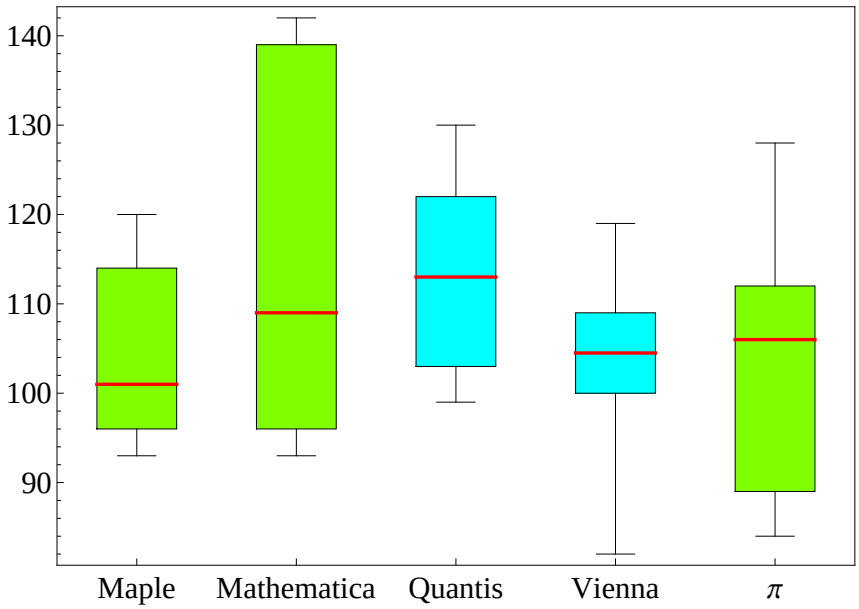


Fig. 2. (Color online) Box-and-whisker plot for the results based on the Solovay-Strassen probabilistic primality test.

4.3 Borel normality test

Borel normality — requesting that every binary string appears in the sequence with the correct probability 2^{-n} for a string of length n — served as the first mathematical definition of randomness [25]. A sequence is (Borel) normal if every binary string appears in the sequence with the right probability (which is 2^{-n} for a string of length n). A sequence is normal if and only it is incompressible by any information lossless finite-state compressor [35], so normal sequences are those sequences that appear random to any finite-state machine.

Every algorithmic random infinite sequence is Borel normal [36]. The converse implication is not true: there exist computable normal sequences (e.g., Champernowne's constant).

Normality is invariant under finite variations: adding, removing, or changing a finite number of bits in any normal sequence leaves it normal. Further, if a sequence satisfies the normality condition for strings of length $n+1$, then it also satisfies normality for strings of length n, but the converse is not true.

Normality was transposed to strings in Ref. [36]. In this process one has to replace limits with inequalities. As a consequence, the above two properties, which are valid for sequences, are no longer true for strings.

For any fixed integer $m > 1$, consider the alphabet $B_m = \{0, 1\}^m$ consisting of all binary strings of length m, and for every $1 \leq i \leq 2^m$ denote by $N_i^m(x)$

the number of occurrences of the lexicographical ith binary string of length m in the string x (considered over the alphabet B_m). By $|x|_m$ we denote the length of x over B_m; $|x|_1 = |x|$. A string x is Borel normal if for every natural $1 \leq m \leq \log_2 \log_2 |x|$,

$$\left| \frac{N_j^m(x)}{|x|_m} - 2^{-m} \right| \leq \sqrt{\frac{\log_2 |x|}{|x|}},$$

for every $1 \leq j \leq 2^m$. In Ref. [36] it is shown that almost all algorithmic random strings are Borel normal.

First test we count the maximum, minimum and difference of non-overlapping occurrences of m-bit ($m = 1, \ldots, 5$) strings in each sample string. Then we tested the Borel normality property for each sample string and found that almost all strings pass the test, with some notable exceptions. We found that several of the Vienna sequences failed the expected count range for $m = 2$ and a few of the Vienna sequences were outside the expected range for $m = 3$ and $m = 4$ (some less then the expected minimum count and some more than the expected maximum count). The only other bit sequence that was outside the expected range count was one of the Mathematica sequences that had a too big of a count for $k = 1$. Figure 3 depicts a box-and-whisker plot of the results. This is followed by statistical (numerical) details in Table 3.

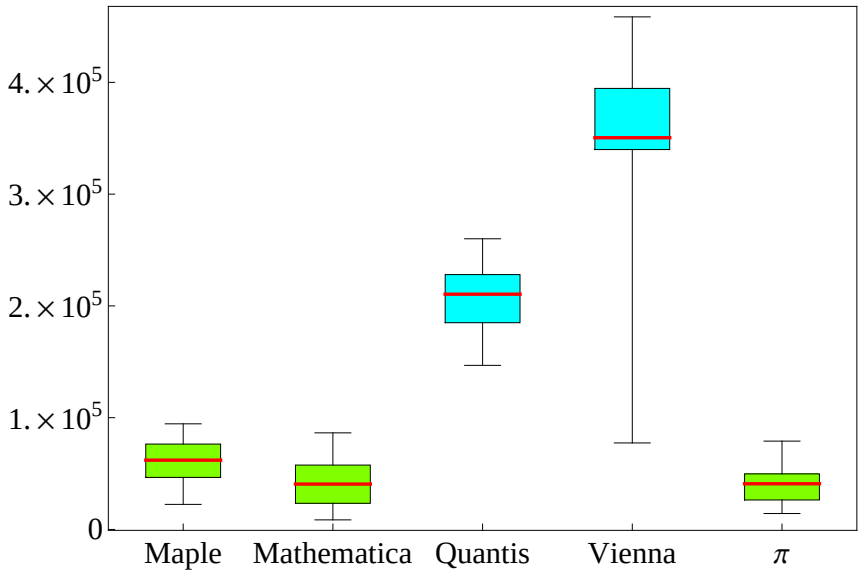


Fig. 3. (Color online) Box-and-whisker plot for the results for tests of the Borel normality property.

Table 3. Statistics for the results for tests of the Borel normality property.

Descriptive statistics	min	Q1	median	Q3	max	mean	sd
Maple	22430	47170	61990	76130	94510	60210	21933.52
Mathematica	8572	25500	40590	55650	86430	41870	23229.77
Quantis	146800	185100	210500	226600	260000	207200	33515.65
Vienna	77410	340200	350500	392500	260000	337100	103354.3
π	14260	28860	40880	47860	79030	40220	17906.21

4.4 Test based on Shannon's information theory

The next test computes "sliding window" estimations of the Shannon entropy $L_n^1, \ldots, L_n^t$ according to the method described in [37]: a smaller entropy is a symptom of less randomness. The results are presented in Figure 4 and Table 4.

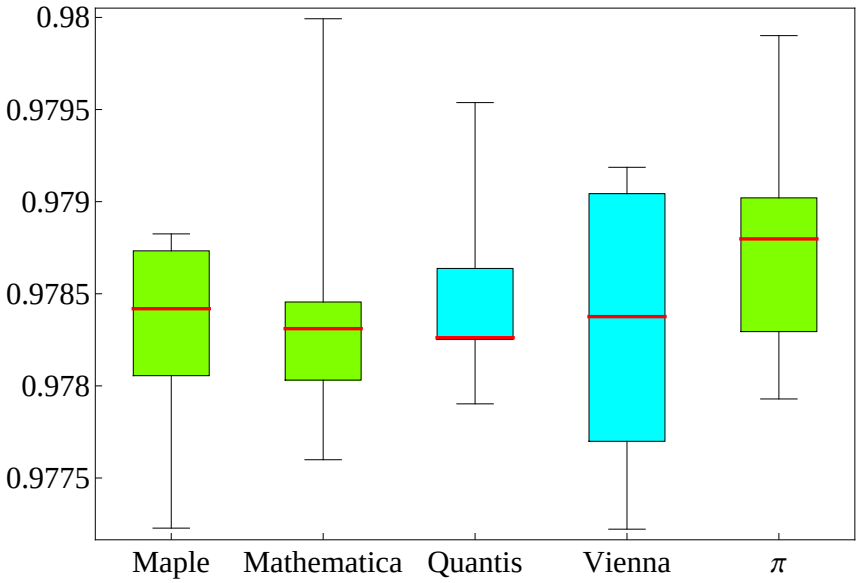


Fig. 4. (Color online) Box-and-whisker plot for average results in "sliding window" estimations of the Shannon entropy.

Table 4. Statistics for average results in "sliding window" estimations of the Shannon entropy.

Descriptive statistics	min	Q1	median	Q3	max	mean	sd
Maple	0.9772	0.9781	0.9784	0.9787	0.9788	0.9783	0.0005231617
Mathematica	0.9776	0.9781	0.9783	0.9785	0.9800	0.9783	0.0006654936
Quantis	0.9779	0.9783	0.9783	0.9786	0.9795	0.9784	0.0004522699
Vienna	0.9772	0.9777	0.9784	0.9790	0.9792	0.9783	0.0006955834
π	0.9779	0.9784	0.9788	0.9790	0.9799	0.9788	0.0006062724

4.5 Test based on random walks

A symptom of non-randomness of a string is detected when the plot generated by viewing a sample sequence as a 1D random walk meanders "less away" from the starting point (both ways); hence the max-min range is the metric.

The fifth test is thus based on viewing a random sequence as a one-dimensional *random walk;* whereby the successive bits, associated with an increase of one unit *per* bit of the x-coordinate, are interpreted as follows: 1 ="move up," and 0 ="move down" on the y-axis. In this way a measure is obtained for how far away one can reach from the starting point (in either positive or negative) from the starting y-value of 0 that one can reach using successive bits of the sample sequence. Figure 5 and Table 5 summarize the results.

Table 5. Statistics for the results of the random walk tests.

Descriptive statistics	min	Q1	median	Q3	max	mean	sd
Maple	67640	88730	126400	162500	180500	125300	42995.59
Mathematica	73500	84760	98110	103400	120300	96450	14685.34
Quantis	138200	161600	209000	250200	294200	211300	55960.23
Vienna	92070	130200	155600	167600	226900	152900	36717.55
π	58570	70420	82800	91920	107500	82120	14833.75

5 Statistical Analysis of Randomness Tests Results

In what follows the significance of results corresponding to each randomness test applied to all five sources are analyzed by means of some statistical comparison tests. The Kolmogorov-Smirnov test for two samples [38] determines if two datasets differ significantly. This test has the advantage of making no

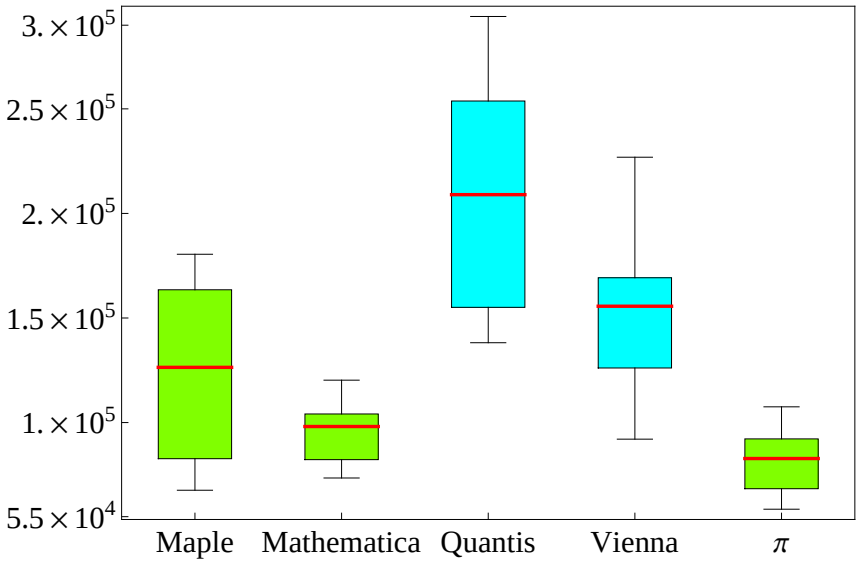


Fig. 5. (Color online) Box-and-whisker plot for the results of the random walk tests.

prior assumption about the distribution of data; i.e., it is non-parametric and distribution free.

The Kolmogorov-Smirnov test returns a p-value, and the decision "the difference between the two datasets is statistically significant" is accepted if the p-value *is less than* 0.05*;* or, stated pointedly, if the probability of taking a wrong decision is less than 0.05. Exact p-values are only available for the two-sided two-sample tests with no ties.

In some cases we have tried to double-check the decision "no significant differences between the datasets" at the price of a supplementary, plausible distribution assumption. Therefore, we have performed the Shapiro-Wilk test for normality [39] and, if normality is not rejected, we have assumed that the datasets have normal (Gaussian) distributions. In order to be able to compare the expected values (means) of the two samples, the Welch t-test [40], which is a version of Student's test, has been applied. In order to emphasize the relevance of p-values less than 0.05 associated with Kolmogorov-Smirnov, Shapiro-Wilk and Welch's t-tests, they are printed in boldface and discussed in the text.

5.1 Book stack randomness test

The results of the Kolmogorov-Smirnov test associated with the "book-stack" tests are enumerated in Table 6. Statistically significant differences are identified for Quantis *versus* Mathematica and π.

As more compression is a symptom of less randomness, the corresponding ranking of samples is as follows: $\langle \text{Quantis} \rangle = 89988.9 > \langle \text{Vienna} \rangle = 53863.8 >$

Table 6. Kolmogorov-Smirnov test for the "book-stack" tests.

Kolmogorov-Smirnov test p-values	Mathematica	Quantis	Vienna	π
Maple	0.4175	0.1678	0.9945	0.4175
Mathematica		**0.0021**	0.1678	0.4175
Quantis			0.1678	**0.0123**
Vienna				0.4175

$\langle \text{Maple} \rangle = 53411.6 > \langle \pi \rangle = 41277.5 > \langle \text{Mathematica} \rangle = 27938.3$. The Shapiro-Wilk tests results are presented in Table 7.

Table 7. Shapiro-Wilk test for the "book-stack" tests.

Shapiro-Wilk test	Maple	Mathematica	Quantis	Vienna	π
p-value	0.7880	0.4819	0.7239	0.8146	0.5172

Since normality is not rejected for any string, we apply the Welch's t-test for the comparison of means. The results are enumerated in Table 8. Significant differences between the means are identified for the following sources: (i) Quantis *versus* all other sources (Maple, Mathematica, Vienna, π); and (ii) Vienna *versus* Mathematica and Maple (as already mentioned).

Table 8. Welch's t-test for the "book-stack" tests.

p-value	Mathematica	Quantis	Vienna	π
Maple	0.0535	**0.0436**	0.974	0.3412
Mathematica		**0.0009**	**0.0283**	0.1551
Quantis			**0.0368**	**0.0054**
Vienna				0.2690

5.2 Solovay-Strassen probabilistic primality test

The Kolmogorov-Smirnov test results for this test are presented in Table 9, where no significant differences are detected.

The Shapiro-Wilk test results are presented in Table 10. Since there is no clear pattern of normality for the data, the application of Welch's t-test is not appropriate.

Table 9. Kolmogorov-Smirnov test for the Solovay-Strassen tests.

Kolmogorov-Smirnov test p-values	Mathematica	Quantis	Vienna	π
Maple	0.7591	0.4005	0.7591	0.7591
Mathematica		0.7591	0.7591	0.7591
Quantis			0.4005	0.7591
Vienna				0.9883

Table 10. Shapiro-Wilk test for the Solovay-Strassen tests.

Shapiro-Wilk test	Maple	Mathematica	Quantis	Vienna	π
p-value	0.0696	**0.0363**	0.4378	0.6963	0.4315

5.3 Borel test of normality

The results of the Kolmogorov-Smirnov test are presented in Table 11.

Table 11. Kolmogorov-Smirnov test for the Borel normality tests.

Kolmogorov-Smirnov test p-values	Mathematica	Quantis	Vienna	π
Maple	0.4175	$< \mathbf{10^{-4}}$	**0.0002**	0.1678
Mathematica		$< \mathbf{10^{-4}}$	**0.0002**	0.9945
Quantis			**0.0002**	$< \mathbf{10^{-4}}$
Vienna				**0.0002**

Statistically significant differences are identified for (i) Quantis *versus* Maple, Mathematica and π; (ii) Vienna *versus* Maple, Mathematica and π; and (iii) Quantis *versus* Vienna.

Note that

1. Pseudo-random strings pass the Borel normality test for comparable, relatively small (with respect to quantum strings; cf. below), numbers of counts: if the angle brackets $\langle x \rangle$ stand for the statistical mean of tests on x, then $\langle \text{Maple} \rangle = 60210$, $\langle \text{Mathematica} \rangle = 41870$, $\langle \pi \rangle = 40220$).
2. Quantum strings pass the Borel normality test only for "much larger numbers" of counts ($\langle \text{Quantis} \rangle = 207200$, $\langle \text{Vienna} \rangle = 337100$).

As a result, the Borel normality test detects and identifies statistically significantly differences between all pairs of computable and incomputable sources of "randomness."

5.4 Test based on Shannon's information theory

The results of the Kolmogorov-Smirnov test are presented in Table 12. No significant differences are detected. The descriptive statistics data for the results of this test indicates almost identical distributions corresponding to the five sources.

Table 12. Kolmogorov-Smirnov test for Shannon's information theory tests.

Kolmogorov-Smirnov test p-values	Mathematica	Quantis	Vienna	π
Maple	0.7870	0.7870	0.7870	0.1678
Mathematica		0.7870	0.4175	0.0525
Quantis			0.4175	0.1678
Vienna				0.4175

The results of the Shapiro-Wilk test associated with a test based on Shannon's information theory are presented in Table 13. Since there is no clear pattern of normality for the data, the application of Welch's t-test is not appropriate.

Table 13. Shapiro-Wilk test for Shannon's information theory tests.

Shapiro-Wilk test	Maple	Mathematica	Quantis	Vienna	π
p-value	0.1962	**0.0189**	**0.0345**	0.3790	0.8774

5.5 Test based on random walks

The Kolmogorov-Smirnov test results associated with test based on random walks are presented in Table 14. Statistically significant differences are identified for: (i) Quantis *versus* all other sources (Maple, Mathematica, Vienna and π); (ii) Vienna *versus* Mathematica, Vienna (as already mentioned) and π; and (iii) Maple *versus* π.

Quantum strings move farther away from the starting point than the pseudo-random strings; i.e., $\langle Quantis \rangle > \langle Vienna \rangle > \langle Maple \rangle > \langle Mathematica \rangle > \langle \pi \rangle$.

Table 14. Kolmogorov-Smirnov test for the random walk tests.

Kolmogorov-Smirnov test p-values	Mathematica	Quantis	Vienna	π
Mathematica	0.1678	**0.0123**	0.4175	0.0525
Quantis		$< \mathbf{10^{-4}}$	**0.0021**	0.1678
Vienna			0.0525	$< \mathbf{10^{-4}}$
π				**0.0002**

It was quite natural to double-check the conclusion "Quantis and Vienna do not exhibit significant differences." Hence we run the Shapiro-Wilk test, which concludes that normality is not rejected; cf. Table 15.

Table 15. Shapiro-Wilk test for the random walk tests.

Shapiro-Wilk test	Maple	Mathematica	Quantis	Vienna	π
p-value	0.2006	0.9268	0.5464	0.8888	0.9577

Next, we apply the Welch's t-test for the comparison of means. The results are given in Table 16. Significant differences between the means are identified for the following sources: (i) Quantis *versus* all other sources (Maple, Quantis, Vienna, π); (ii) Vienna *versus* Mathematica, Quantis (as already mentioned) and π; (iii) Maple *versus* π.

6 Summary

Tests based on algorithmic information theory analyze algorithmic randomness, the strongest possible form of incomputability. In this respect they differ from tests employed in the standard test batteries, as the former depend on

Table 16. Welch's t-tests for the random walk tests.

p-value	Mathematica	Quantis	Vienna	π
Maple	0.06961	**0.0013**	0.1409	**0.0119**
Mathematica		$< \mathbf{10^{-4}}$	**0.0007**	**0.0435**
Quantis			**0.0143**	$< \mathbf{10^{-4}}$
Vienna				**0.0001**

irreducible algorithmic information content, which is constant for algorithmic pseudo-random generators. Thus the set of randomness tests performed for our analysis could in principle be expected to be "more sensitive" with respect to differentiating between quantum randomness and algorithmic types of "quasi-randomness" than statistical tests alone.

All tests have produced evidence — with different degrees of statistical significance — of differences between quantum and non-quantum sources. In summary:

1. For the test for Borel normality — the strongest discriminator test — statistically significant differences between the distributions of datasets are identified for (i) *Quantis* versus *Maple*, *Mathematica* and π; (ii) *Vienna* versus *Maple*, *Mathematica* and π; and (iii) *Quantis* versus *Vienna*.
   Not only that the average number of counts is larger for quantum sources, but the increase is quite significant: *Quantis* is $3.5 - 5$ times larger than the corresponding average number of counts for software-generated sources, and *Vienna* is $5 - 8$ times larger than those values.
2. For the test based on random walks, statistically significant differences between the distributions of datasets are identified for: (i) *Quantis* versus all other sources (*Maple*, *Mathematica*, *Vienna* and π); (ii) *Vienna* versus *Mathematica*, *Vienna* and π. Quantum strings move farther away from the starting point than the pseudo-random strings; i.e., $\langle Quantis \rangle > \langle Vienna \rangle > \langle Maple \rangle > \langle Mathematica \rangle > \langle \pi \rangle$.
3. For the "book-stack" test, significant differences between the means are identified for the following sources: (i) *Quantis* versus all other sources (*Maple*, *Mathematica*, *Vienna*, π); and (ii) *Vienna* versus *Mathematica* and *Maple*.
4. For the test based on Shannon's information theory, as well as for the Solovay-Strassen test, *no significant differences* among the five chosen sources are detected. In the first case the reason may come from the fact that averages are the same for all samples. In the second case the reason may be due to the fact that the test is based solely on the behavior of algorithmic random strings and not on a specific property of randomness.

We close with a cautious remark about the impossibility to formally or experimentally "prove absolute randomness." Any claim of randomness can only be secured *relative* to, and *with respect* to, a more or less large class of laws or behaviors, as it is impossible to inspect the hypothesis against an infinity of — and

even less so all — conceivable laws. To rephrase a statement about computability [41, p. 11], *"how can we ever exclude the possibility of our presented, some day (perhaps by some extraterrestrial visitors), with a (perhaps extremely complex) device that "computes" and "predicts" a certain type of hitherto "random" physical behavior?"*

Acknowledgements

We are grateful to Thomas Jennewein and Anton Zeilinger for providing us with the quantum random bits produced at the University of Vienna by Vienna IQOQI group, for the description of their method, critical comments and interest in this research.

We thank: a) Alastair Abbott, Hector Zenil and Boris Ryabko for interesting comments, b) Ulrich Speidel for his tests for which some partial results have been reported in our extended paper [27], c) Stefan Wegenkittl for critical comments of various drafts of this paper and his suggestions to exclude some tests, d) the anonymous Referees for constructive suggestions.

Calude gratefully acknowledges the support of the Hood Foundation and the Vienna University of Technology. Svozil gratefully acknowledges support of the CDMTCS at the University of Auckland, as well as of the Ausseninstitut of the Vienna University of Technology.

References

1. M. Born, "Zur Quantenmechanik der Stoßvorgänge," Zeitschrift für Physik **37,** 863–867 (1926).
2. M. Born, "Quantenmechanik der Stoßvorgänge," Zeitschrift für Physik **38,** 803–827 (1926).
3. A. Zeilinger, "The message of the quantum," Nature **438,** 743 (2005).
4. C. S. Calude and K. Svozil, "Quantum Randomness and Value Indefiniteness," Advanced Science Letters **1,** 165–168 (2008).
5. P. Martin-Löf, "The definition of random sequences," Information and Control **9,** 602–619 (1966).
6. G. J. Chaitin, *Exploring Randomness* (Springer Verlag, London, 2001).
7. C. Calude, *Information and Randomness—An Algorithmic Perspective*, 2 ed. (Springer, Berlin, 2002).
8. K. Svozil, "The quantum coin toss—Testing microphysical undecidability," Physics Letters A **143,** 433–437 (1990).
9. J. G. Rarity, M. P. C. Owens, and P. R. Tapster, "Quantum Random-number Generation and Key Sharing," Journal of Modern Optics **41,** 2435–2444 (1994).
10. T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, "A Fast and Compact Quantum Random Number Generator," Review of Scientific Instruments **71,** 1675–1680 (2000).
11. A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden, "Optical quantum random number generator," Journal of Modern Optics **47,** 595–598 (2000).

12. M. Hai-Qiang, W. Su-Mei, Z. Da, C. Jun-Tao, J. Ling-Ling, H. Yan-Xue, and W. Ling-An, "A Random Number Generator Based on Quantum Entangled Photon Pairs," Chinese Physics Letters **21,** 1961–1964 (2004).
13. P. X. Wang, G. L. Long, and Y. S. Li, "Scheme for a quantum random number generator," Journal of Applied Physics **100,** 056107 (2006).
14. M. Fiorentino, C. Santori, S. M. Spillane, R. G. Beausoleil, and W. J. Munro, "Secure self-calibrating quantum random-bit generator," Physical Review A (Atomic, Molecular, and Optical Physics) **75,** 032334 (2007).
15. K. Svozil, "Three criteria for quantum random-number generators based on beam splitters," Physical Review A (Atomic, Molecular, and Optical Physics) **79,** 054306 (2009).
16. G. Marsaglia (unpublished).
17. A. Rukhin *et al.*, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22* (National Institute of Standards and Technology (NIST), 2001).
18. P. L'Ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," ACM Transactions on Mathematical Software (TOMS) **33,** 22 (2007).
19. ID Quantique SA, *QUANTIS. Quantum number generator* (idQuantique, Geneva, Switzerland, 2001-2010).
20. T. Jennewein, Institut für Quantenoptik und Quanteninformation (IQOQI), *Quantum number generator* (2003), personal communication.
21. Wolfram Research, Inc., *Mathematica Edition: Version 6.0. Mathematica random generator. ExtendedCA default."* (Wolfram Research, Inc., Waterloo, Ontario, 2007).
22. Maplesoft, *Maple Edition: Version 11. Maple random generator* (Maplesoft, Champaign, Illinois, 2007).
23. Y. Kanada and D. Takahashi, *Calculation of π up to 4 294 960 000 decimal digits* University of Tokyo, 1995.
24. J. von Neumann, "Various Techniques Used in Connection With Random Digits," National Bureau of Standards Applied Math Series **12,** 36–38 (1951), reprinted in *John von Neumann, Collected Works, (Vol. V)*, A. H. Traub, editor, MacMillan, New York, 1963, p. 768–770.
25. É. Borel, "Les probabilités dénombrables et leurs applications arithmétiques," Rendiconti del Circolo Matematico di Palermo (1884 - 1940) **27,** 247–271 (1909).
26. P. Hertling, "Simply Normal Numbers to Different Bases," Journal of Universal Computer Science **8,** 235–242 (2002).
27. C. S. Calude, M. J. Dinneen, M. Dumitrescu, and K. Svozil, "How Random Is Quantum Randomness? (Extended Version)," Report CDMTCS-372, Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, Auckland, New Zealand (2009) .
28. B. Y. Ryabko and A. I. Pestunov, ""Book stack" as a new statistical test for random numbers," Problemy Peredachi Informatsii **40,** 73–78 (2004).
29. B. Y. Ryabko and V. A. Monarev, "Using information theory approach to randomness testing," J. Statist. Plann. Inference **133,** 95–110 (2005).
30. R. Solovay and V. Strassen, "A Fast Monte-Carlo Test for Primality," SIAM Journal on Computing **6,** 84–85 (1977), corrigendum in Ref. [42].
31. G. J. Chaitin and J. T. Schwartz, "A note on Monte Carlo primality tests and algorithmic information theory," Communications on Pure and Applied Mathematics **31,** 521–527 (1978).

32. R. G. Pinch (unpublished).
33. R. G. Pinch, "The Carmichael numbers up to 10^{21}," In *Proceedings of Conference on Algorithmic Number Theory 2007. TUCS General Publication No 46*, A.-M. Ernvall-Hytönen, M. Jutila, J. Karhumäki, and A. Lepistö, eds., pp. 129–131 (Turku Centre for Computer Science, Turku, Finland, 2007).
34. C. Calude and M. Zimand, "A relation between correctness and randomness in the computation of probabilistic algorithms," Internat. J. Comput. Math. **16,** 47–53 (1984).
35. J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," IEEE Transactions on Information Theory **24,** 530–536 (1978).
36. C. Calude, "Borel Normality and Algorithmic Randomness," in *Developments in Language Theory*, G. Rozenberg and A. Salomaa, eds., (World Scientific, Singapore, 1994), pp. 113–129.
37. A. D. Wyner, "Shannon Lecture: Typical Sequences and All That: Entropy, Pattern Matching, and Data Compression," IEEE Information Theory Society (1994).
38. W. J. Conover, *Practical Nonparametric Statistics* (John Wiley & Sons, New York, 1999), p. 584.
39. S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," Biometrika **52,** 591–611 (2005).
40. B. L. Welch, "The generalization of "Student's" problem when several different population variances are involved," Biometrika 34 (1947).
41. M. Davis, *Computability and Unsolvability* (McGraw-Hill, New York, 1958).
42. R. Solovay and V. Strassen, "Erratum: A Fast Monte-Carlo Test for Primality," SIAM Journal on Computing **7,** 118 (1978).

Fermat's Last Theorem and Chaoticity

Elena Calude

Institute of Information and Mathematical Sciences, Massey University at Albany,
Private Bag 102-904, North Shore MSC New Zealand
e.calude@massey.ac.nz

Abstract. Proving that a dynamical system is chaotic is a central problem in chaos theory [11]. In this note we apply the computational method developed in [4, 2, 3] to show that Fermat's last theorem is in the lowest complexity class $\mathfrak{C}_{U,1}$. Using this result we prove the existence of a two-dimensional Hamiltonian system for which the proof that the system has a Smale horseshoe is in the class $\mathfrak{C}_{U,1}$, i.e. it is not too complex.

1 Introduction

A system is chaotic if small differences in initial conditions could yield widely diverging outcomes; for such a system long-term prediction is in general impossible. Even deterministic systems whose dynamics are fully determined by their initial conditions, and no random elements are involved, can be chaotic [13, 8].

There are only few "bridges" between chaotic dynamical systems and complexity theories, in particular algorithmic information theory. Recently, [9] showed that in classical chaotic dynamics, typicality corresponds exactly to Schnorr randomness; this means that a chaotic system may produce a computable sequence of bits provided the initial point is suitable chosen, but this event has probability zero (the set of initial points can be infinite).

Virtually any "interesting" question about non-trivial dynamical systems is undecidable. Undecidability does not imply the impossibility to prove non-trivial properties of dynamical systems, in particular, chaoticity: it says that there is no general method, new specifically designed methods are required for different problems.

How difficult is to prove chaoticity? Building on results in [15, 12] in [7] a two-dimensional Hamiltonian system $\mathcal{H}$ was constructed with the property that in Zermelo-Fraenkel set theory with the Axiom of Choice (ZFC) proving the existence of a Smale horseshoe in $\mathcal{H}$ is equivalent to proving Fermat's last theorem. We say that "ZFC proves s" in case there is a proof in ZFC for s. We can now state more precisely the result described above:

Theorem 1 [7] *Assume* ZFC *is arithmetically sound (that is, any theorem of arithmetic proved by* ZFC *is true). Then, one can effectively construct in the formal language of* ZFC *the expression describing a two-dimensional Hamiltonian system $\mathcal{H}$ such that* ZFC *proves that $\mathcal{H}$ has a Smale horseshoe iff* ZFC *proves Fermat's last theorem.*

The choice of the Fermat's last theorem in [7] was motivated by the contrast between the short length of this elementary statement and the belief that any proof of the theorem has to be very complicated; this belief was indeed confirmed by the proof in [17].

Is the excruciating long proof of the Fermat's last theorem [17] a good indication that any proof that the corresponding two-dimensional Hamiltonian system is chaotic should be very complex, hence proving chaoticity is a difficult problem?

First, the fact that the known proof is complex is not a proof that every proof for Fermat's last theorem is complex.

Secondly, the result proven in [10]—which shows that in ZFC one can (effectively) find infinite sets of trivially true theorems which require as long proofs as the hardest theorems—indicates that the length of a proof may not be an adequate complexity measure for how complicated/deep the theorem is. In the words of Hartmanis [10]:

> In every formalization, infinite sets of trivial theorems will require very long proofs. ... It also gives a warning that a necessarily long proof in a formal system does not certify that the result is non-trivial.

Using the method developed in [4, 2, 3] we prove that Fermat's last theorem is in the class $\mathfrak{C}_{U,1}$, hence from this point of view its complexity is low.

The paper is organised as follows. In the next section we present a short proof for Theorem 1; in section 3 we briefly describe the complexity measure; in section 4 we use this measure to obtain an upper bound on the complexity of Fermat's last theorem which shows that this statement is in the class $\mathfrak{C}_{U,1}$ and this bound is transferred to the statement regarding the chaoticity of a specific two-dimensional Hamiltonian system via the equivalence in Theorem 1; in section 5 we present some conclusions and an open problem.

2 A Proof for Theorem 1

In [7] Theorem 1. was proved using Richardson'e lemma from Richardson [15], Caviness [5] and Wang [16]. We present here a shorter direct proof avoiding the use of Richardson'e lemma.

Proof. Let b be Cantor's bijection on non-negative integers: $b(i,j) = (i+j)(i+j+1)/2 + i$. Denote by $\mathrm{pr}_1, \mathrm{pr}_2$ the inverses of b: $b(\mathrm{pr}_1(t), \mathrm{pr}_2(t)) = t$, $\mathrm{pr}_1(b(i,j)) = i, \mathrm{pr}_2(b(i,j)) = j$.

Fermat's last theorem states that the equality $(x+1)^{m+3} + (y+1)^{m+3} = (z+1)^{m+3}$ is not valid for every non-negative integers x, y, z, m. It is seen that this is equivalent to the statement $\forall n[\mathrm{Pred}(n)]$, where $\mathrm{Pred}(n)$ is the computable predicate

$$(\mathrm{pr}_1(\mathrm{pr}_1(\mathrm{pr}_1(n))))^{\mathrm{pr}_2(n)} + (\mathrm{pr}_2(\mathrm{pr}_1(\mathrm{pr}_1(n))))^{\mathrm{pr}_2(n)}$$
$$\neq (\mathrm{pr}_2(\mathrm{pr}_1(n)))^{\mathrm{pr}_2(n)} \text{ and } n \geq 272.$$

We denote by h and k the Hamiltonian for the two-dimensional system with a Smale horseshoe as defined by Holmes and Marsden [12] (their Example 4) and the Hamiltonian for the free particle, respectively. Clearly, the systems h and k can be represented in the formal language of ZFC. Define the Hamiltonian $\mathcal{H}_\pi$ as a linear combination of h, k:

$$\begin{aligned}\mathcal{H}_\pi(q_1,\ldots,q_n,p_1,\ldots,p_n) = \mathrm{Pred}(m)\cdot h(q_1,\ldots,q_n,p_1,\ldots,p_n)\\ +(1-\mathrm{Pred}(m))\cdot k(q_1,\ldots,q_n,p_1,\ldots,p_n).\end{aligned}$$

In view of its definition, $\mathcal{H}_\pi$ can be represented in the formal language of ZFC and we have

$$\mathcal{H}_\pi(q_1,\ldots,q_n,p_1,\ldots,p_n) = h(q_1,\ldots,q_n,p_1,\ldots,p_n) \text{ iff ZFC proves } \pi,$$

hence

ZFC proves π iff ZFC proves that $\mathcal{H}_\pi$ has a Smale horseshoe.

3 A Complexity Measure

In this section we present a complexity measure [4, 2, 3] for Π_1-statements (i.e. statements of the form "$\forall n\, Pred(n)$", where $Pred$ is a computable predicate) defined by means of register machine programs.

We use a fixed "universal formalism" for programs, more precisely, a universal self-delimiting Turing machine U. The machine U (which is fully described below) has to be *minimal* in the sense that none of its instructions can be simulated by a program for U written with the remaining instructions.

To every Π_1–problem $\sigma = \forall m P(m)$ we associate the algorithm $\Pi_P = \inf\{n : P(n) = \text{ false}\}$ which systematically searches for a counter-example for σ. There are many programs (for U) which implement Π_P; without loss of generality, any such program will be denoted also by Π_P. Note that σ is true iff $U(\Pi_P)$ never halts.

The complexity (with respect to U) of a Π_1–problem σ is defined by the length of the smallest-length program (for U) Π_P—defined as above—where minimisation is calculated for all possible representations of σ as $\sigma = \forall n P(n)$:[1]

$$C_U(\sigma) = \min\{|\Pi_P| : \sigma = \forall n P(n)\}.$$

Because the complexity C_U is incomputable, we can work only with upper bounds for C_U. As the exact value of C_U is not important, following [3] we classify Π_1–problems into the following classes:

$$\mathfrak{C}_{U,n} = \{\sigma : \sigma \text{ is a } \Pi_1\text{–problem}, C_U(\sigma) \le n \text{ kbit}^2\}.$$

[1] For C_U it is irrelevant whether σ is known to be true or false. In particular, the program containing the single instruction halt is not a Π_P program, for any P.

[2] A kilobit (kbit or kb) is equal to 2^{10} bits.

We briefly describe the syntax and the semantics of a register machine language which implements a (natural) minimal universal prefix-free binary Turing machine U used for evaluating the complexity of Fermat's last theorem, a Π_1–problem.

Any register program (machine) uses a finite number of registers, each of which may contain an arbitrarily large non-negative integer.

By default, all registers, named with a string of lower or upper case letters, are initialised to 0. Instructions are labeled by default with 0,1,2,...

The register machine instructions are listed below. Note that in all cases R2 and R3 denote either a register or a non-negative integer, while R1 must be a register. When referring to R we use, depending upon the context, either the name of register R or the non-negative integer stored in R.

=R1,R2,R3

If the contents of R1 and R2 are equal, then the execution continues at the R3-th instruction of the program. If the contents of R1 and R2 are not equal, then execution continues with the next instruction in sequence. If the content of R3 is outside the scope of the program, then we have an illegal branch error.

&R1,R2

The contents of register R1 is replaced by R2.

+R1,R2

The contents of register R1 is replaced by the sum of the contents of R1 and R2.

!R1

One bit is read into the register R1, so the contents of R1 becomes either 0 or 1. Any attempt to read past the last data-bit results in a run-time error.

%

This is the last instruction for each register machine program before the input data. It halts the execution in two possible states: either successfully halts or it halts with an under-read error.

A *register machine program* consists of a finite list of labeled instructions from the above list, with the restriction that the halt instruction appears only once, as the last instruction of the list. The input data (a binary string) follows immediately after the halt instruction. A program not reading the whole data or attempting to read past the last data-bit results in a run-time error. Some programs (as the ones presented in this paper) have no input data; these programs cannot halt with an under-read error.

The instruction `=R,R,n` is used for the unconditional jump to the n-th instruction of the program. For Boolean data types we use integers $0 =$ `false` and $1 =$ `true`.

For longer programs it is convenient to distinguish between the main program and some sets of instructions called "routines" which perform specific tasks for another routine or the main program. The call and call-back of a routine are executed with unconditional jumps.

To compute an upper bound on the complexity of the Fermat last theorem we need to compute the size in bits of the program Π_{Fermat}, so we need to uniquely code in binary the programs for U. To this aim we use a prefix-free coding as follows.

The binary coding of special characters (instructions and comma) is the following (ε is the empty string):

special characters	code	instruction	code
,	ε	+	111
&	01	!	110
=	00	%	100

Table 1

For registers we use the prefix-free regular code $\text{code}_1 = \{0^{|x|}1x \mid x \in \{0,1\}^*\}$. Here are the codes of the first 14 registers:[3]

register	code_1	register	code_1
R_1	010	R_8	0001001
R_2	011	R_9	0001010
R_3	00100	R_{10}	0001011
R_4	00101	R_{11}	0001100
R_5	00110	R_{12}	0001101
R_6	00111	R_{13}	0001110
R_7	0001000	R_{14}	0001111

Table 2

For non-negative integers we use the prefix-free regular code $\text{code}_2 = \{1^{|x|}0x \mid x \in \{0,1\}^*\}$. Here are the codes of the first 16 non-negative integers:

integer	code_2	integer	code_2	integer	code_2	integer	code_2
0	100	4	11010	8	1110010	12	1110110
1	101	5	11011	9	1110011	13	1110111
2	11000	6	1110000	10	1110100	14	111100000
3	11001	7	1110001	11	1110101	15	111100001

[3] The register names are chosen to optimise the length of the program, i.e. the most frequent registers have the smallest code_1 length.

Table 3

The instructions are coded by self-delimiting binary strings as follows:

1. `&R1,R2` is coded in two different ways depending on R2:[4]

$$01\text{code}_1(\text{R1})\text{code}_i(\text{R2}),$$

   where $i = 1$ if R2 is a register and $i = 2$ if R2 is an integer.
2. `+R1,R2` is coded in two different ways depending on R2:

$$111\text{code}_1(\text{R1})\text{code}_i(\text{R2}),$$

   where $i = 1$ if R2 is a register and $i = 2$ if R2 is a non-negative integer.
3. `=R1,R2,R3` is coded in four different ways depending on the data types of R2 and R3:

$$00\text{code}_1(\text{R1})\text{code}_i(\text{R2})\text{code}_j(\text{R3}),$$

   where $i = 1$ if R2 is a register and $i = 2$ if R2 is a non-negative integer, $j = 1$ if R3 is a register and $j = 2$ if R3 is a non-negative integer.
4. `!R1` is coded by

$$110\text{code}_1(\text{R1}).$$

5. `%` is coded by

$$100.$$

4 The Complexity of Fermat's Last Theorem

Fermat's last theorem is one of the most famous theorems in the history of mathematics. It states that there are no positive integers x, y, z satisfying the equation $x^n + y^n = z^n$, for any integer value $n > 2$. The result was conjectured by Pierre de Fermat in 1637, and it was proven only in 1995 by A. Wiles [17] (see also [1]). Many illustrious mathematicians failed to proved it, but their efforts stimulated the development of algebraic number theory.

The register machine program presented below uses the integer $B \geq 5$ to enumerate all 4-tuples of integers (x, y, z, n) with $z \leq B, x, y < z, n \leq B$ for which the equality $x^n + y^n = z^n$ is tested.

The register machine program for Fermat's last theorem is:

```
0. =a,a,14
1. &e,0     //===a^b
2. &d,1
3. +e,1
4. &f,0
5. &g,0
6. +f,1
```

[4] As $x\varepsilon = \varepsilon x = x$, for every string $x \in \{0,1\}^*$, in what follows we omit ε.

```
 7. +g,a
 8. =f,d,10
 9. =a,a,6
10. &d,g     //g = a*d
11. =e,b,13
12. =a,a,3
13. =a,a,c   //d = a^b
14. &B,4     //===main program
15. +B,1
16. &n,3
17. +n,1
18. =n,B,15
19. &z,3
20. +z,1
21. =z,B,17
22. &x,3
23. +x,1
24. =x,z,20
25. &y,3
26. +y,1
27. =y,z,23
28. &b,n
29. &a,x
30. &c,32
31. =a,a,1   //d = x^n
32. &E,d
33. &a,y
34. +c,4     //c = 36
35. =a,a,1   //d = y^n
36. +E,d     //E = x^n + y^n
37. &a,z
38. +c,4     //c = 40
39. =a,a,1   //d = z^n
40. =E,a,42  //x^n + y^n = z^n
41. =a,a,26  //x^n + y^n =/= z^n
42. %        //Fermat Theorem is false
```

The register machine program for Fermat's last theorem has 43 instructions. Its size is 597 bits[5], hence the Fermat's last theorem is in $\mathfrak{C}_{U,1}$. According to Theorem 1 we obtain:

Theorem 2 *Assume* ZFC *is arithmetically sound. Then, one can effectively construct in the formal language of* ZFC *the expression describing a two-dimensional Hamiltonian system* $\mathcal{H}$ *such that* ZFC *proves that* $\mathcal{H}$ *has a Smale horseshoe iff there exists a* Π_1*-statement* $\sigma \in \mathfrak{C}_{U,1}$ *such that* ZFC *proves* σ.

[5] We use: R_1 = a, R_2 = d, R_3 = z, R_4 = c, R_5 = B, R_6 = x, R_7 = n, R_8 = y, R_9 = e, R_{10} = f, R_{11} = g, R_{12} = E, R_{13} = b.

5 Conclusions

Using the computational method in [4, 2, 3] we have shown that the problem of proving the existence of a Smale horseshoe in a two-dimensional Hamiltonian system is in the class $\mathfrak{C}_{U,1}$, i.e. it has low complexity according to our complexity measure. The specific pair of two-dimensional Hamiltonians used in the proof of Theorem 1 plays no specific role: any pair of Hamiltonians, one for a dynamics displaying chaotic behaviour and one for a smooth dynamics, will be equally useful in Eq (**??**).

It will be interesting to investigate whether the results presented in this note for Fermat's last theorem can be generalised for any Π_1-statement (in [7] it is claimed that Theorem 1 is true for a couple of other Π_1-statements).

References

1. A. Aczel. *Fermat's Last Theorem: Unlocking the Secret of an Ancient Mathematical Problem*, Dell Publishing, New York, 1996.
2. C. S. Calude, E. Calude. Evaluating the Complexity of Mathematical Problems. Part 1 *Complex Systems*, 18 (2009), 267–285.
3. C.S. Calude, E. Calude. Evaluating the Complexity of Mathematical Problems. Part 2, *Complex Systems* 18 (2010), 387–401.
4. C. S. Calude, E. Calude, M. J. Dinneen. A new measure of the difficulty of problems, *Journal for Multiple-Valued Logic and Soft Computing* 12 (2006), 285–307.
5. B. F. Caviness. On canonical forms and simplification, *Journal of the Association for Computing Machinery* 17, 2 (1970), 385–396.
6. G. J. Chaitin. *Algorithmic Information Theory*, Cambridge University Press, Cambridge, 1987. (third printing 1990)
7. N. C. A. da Costa, F. A. Doria, A. F. Furtado do Amaral. Dynamical system where proving chaos is equivalent to proving Fermat's conjecture, *International Journal of Theoretical Physics* 32, 11 (1993), 2187–2206.
8. R. L. Devaney. *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Westview Press, 2003.
9. P. Gács, M. Hoyrup, C. Rojas. Randomness on computable probability spaces. A dynamical point of view, *Symposium on Theoretical Aspects of Computer Science 2009* (Freiburg), pp. 469–480.
10. J. Hartmanis. On effective speed-up and long proofs of trivial theorems in formal theories, *Informatique Théorique et Applications* 10 (1976), 29–38.
11. M. Hirsch. The chaos of dynamical systems, in P. Fisher, W. R. Smith (eds.). *Chaos, Fractals and Dynamics*, Marcel Dekker, 1985, 189–195.
12. P. J. Holmes, J. E. Marsden. Horseshoes in perturbations of Hamiltonian systems with two degrees of freedom, *Communications in Mathematical Physics* 82 (1982), 523–544.
13. S. H. Kellert. *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*, University of Chicago Press, 1993.
14. M. Laczkovich. The removal of π from some undecidable problems involving elementary functions, *Proceedings of the American Mathematical Society* 131, 7 (2002), 2235–2240.
15. D. Richardson. Some unsolvable problems involving elementary functions of a real variable, *Journal of Symbolic Logic* 33 (1968), 514–520.

16. P. Wang. The undecidability of the existence of zeros of real elementary functions, *Journal of the Association for Computing Machinery* 21 4, (1974), 586–589.
17. A. Wiles. Modular elliptic curves and Fermat's Last Theorem, *Annals of Mathematics* 141 (3) (1995), 443–551.

Quantum Algorithms with Continuous Variables for Black Box Problems

Nicolas J. Cerf[1], Peter Høyer[23], Loïck Magnin[14], and Barry C. Sanders[2]

[1] QuIC, École Polytechnique, CP 165, Université Libre de Bruxelles, 1050 Brussels, Belgium
[2] Institute for Quantum Information Science, University of Calgary, Calgary, Alberta, Canada, T2N 1N4.
[3] Department of Computer Science, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada, T2N 1N4.
[4] LRI, Univ Paris-Sud, CNRS; F-91405 Orsay, France

1 Introduction

Analog computation is a very powerful theoretical model of computation as it allows for real numbers to be manipulated directly. But in practice, the lack of efficient error correcting codes makes it far less robust that its digital counterpart. The situation is similar for quantum computing.

Computation with Continuous Variables (CV) is a quantum mechanical analogue of analog computation. CV has in particular achieved significant successes in quantum cryptography, including unconditionally secure key distribution [RC09] and teleportation. This is in parts achieved by offering two strengths: (1) the existence of a very efficient measurement scheme, the homodyne detection that measures the amplitude of an electromagnetic field, and (2) the Gaussian formalism which allows ability to manipulate infinite dimensional Hilbert spaces.

Gaussian states and Gaussian operations are not universal for quantum computation [BSBN02], but remain a main tool in CV information theory [CLP07]. One can wonder how much non-Gaussian operation one needs to perform universal quantum computation. Knill, Laflamme and Milburn [KLM01] show that we can obtain universality by allowing the creation of an initial non-Gaussian state and allow for non-Gaussian measurements.

In this paper we investigate a framework in which algorithms do the opposite: creation and measurements are Gaussian. Between them, there is only a call to a non-Gaussian oracle. We present here only the study of the Deutsch-Jozsa algorithm, but our results extend naturally to other black box problems such as Bernstein-Vazirani and Simon problems.

Problem DJ — Let $f : \{0,1\}^n \mapsto \{0,1\}$ be a function promised to be either constant, either balanced. Determine whether f is balanced or constant.

Classically, this problem needs an exponential number of queries to be solved exactly and k queries are needed by a randomized algorithm to solve the problem with an error exponentially small in k. In the discrete variable (DV) model, this problem has been solved by Deutsch and Jozsa [DJ92] deterministically with one

single call to the oracle whereas its classical counterpart needs an exponential number of queries.

We choose this problem because we can find two contradicting results in the Literature when using CV. The first result has been obtained by Pati and Braunstein [PB03] where they encode the input in only a single mode. Unfortunately they use position states that do not exist. Adcock, Høyer and Sanders [AHS08] developed this idea by giving a finite width to pulses so that they have only physical states. Not surprisingly they have a probabilistic algorithm where one should repeat $\mathcal{O}(m)$ times the algorithm in order to have an exponentially small error in m, thus giving the quantum algorithm no more advantage on the classical probabilistic one. The error done by one execution of the algorithm cannot be make as small as possible by reducing the width of the pulse. This effect is a consequence of encoding all the input state into one single mode.

The algorithm presented here mimics the usual Deutsch-Jozsa algorithm by encoding only one qubit into one physical system described by a continuous variable. This algorithm solves the DJ problem with only one single query with an error ε arbitrary small depending of the energy of the states $E = \mathcal{O}(\sqrt{\ln \frac{n}{\varepsilon}})$.

In the Literature, one can find three main encodings. Gottesman, Kitaev and Preskill [GKP01] have introduced a nice encoding whom main feature is to allow efficient error correcting scheme in the KLM setting, but they are not realizable in a lab. On a more practical side, the encoding studied in [RGM$^+$03] where the logical bits 0 and 1 are encoded in almost orthogonal coherent states $|\alpha\rangle$ and $|-\alpha\rangle$ have concentrated a lot of interest. The downside of this encoding is that unitaries should be replaced by probabilistic operations. We choose to use another encoding, in which the basis states are encoded into orthogonal optical cat states.

2 Our algorithm

Let us define the Gaussian state $|\alpha, s\rangle$ being the displaced squeezed vacuum with squeezing parameter $\frac{1}{s}$ (where $0 < s \leq 1$) and displacement $\alpha \in \mathbb{C}$. Without loss of generality, we consider only the case where $\alpha > 0$. In particular, the states $|\alpha, s = 1\rangle$ are coherent states $|\alpha\rangle$ that are the output of any (good laser).

The wave function of $|\alpha, s\rangle$ is:

$$\langle x|\alpha, s\rangle = \frac{1}{\pi^{1/4}\sqrt{s}} \exp\left\{-\frac{1}{s^2}\left(\frac{x}{\sqrt{2}} - \alpha\right)^2\right\}. \tag{1}$$

We note $\mathcal{H}_2 = \text{span}\{|\alpha, s\rangle, |-\alpha, s\rangle\}$ the Hilbert space associated to one qubit. Since $\langle \alpha, s|-\alpha, s\rangle = e^{-2(\frac{\alpha}{s})^2}$, $\{|\alpha, s\rangle, |-\alpha, s\rangle\}$ is not an orthonormal basis of $\mathcal{H}_2$ but we can define the cat states

$$|\mathbf{0}\rangle \equiv \frac{|\alpha, s\rangle + |-\alpha, s\rangle}{\sqrt{2(1 + e^{-2(\frac{\alpha}{s})^2})}} \quad \text{and} \quad |\mathbf{1}\rangle \equiv \frac{|\alpha, s\rangle - |-\alpha, s\rangle}{\sqrt{2(1 - e^{-2(\frac{\alpha}{s})^2})}}, \tag{2}$$

so that $\{|\mathbf{0}\rangle, |\mathbf{1}\rangle\}$ is an orthonormal basis of $\mathcal{H}_2$.

Writing $|\pm\alpha, s\rangle$ in the computational basis reads $|\pm\alpha, s\rangle = \sqrt{p}|\mathbf{0}\rangle \pm \sqrt{q}|\mathbf{1}\rangle$ with $p = \frac{1+e^{-2(\frac{\alpha}{s})^2}}{2}$ and $q = \frac{1-e^{-2(\frac{\alpha}{s})^2}}{2}$. (bold letters are used for logical qubits, ie. $|\boldsymbol{x}\rangle$ is string of qubits of the integer x and $|x\rangle$ is the x position state.)

The states $|\alpha, s\rangle$ and $|-\alpha, s\rangle$ are not orthogonal but the POVM $\{\pi_+, \pi_-\}$ with

$$\pi_+ = \int_{x=0}^{+\infty} |x\rangle\langle x| \mathrm{d}x \quad \text{and} \quad \pi_- = \int_{x=-\infty}^{0} |x\rangle\langle x| \mathrm{d}x \tag{3}$$

can distinguish them with an exponentially small error in $\frac{\alpha}{s}$:

$$\Pr(error) = \mathrm{Tr}[\pi_+ |-\alpha, s\rangle\langle -\alpha, s|] = \int_{x=0}^{+\infty} |\langle -\alpha, s|x\rangle|^2 \mathrm{d}x \tag{4}$$

$$= \frac{1}{2}\left(1 - \mathrm{erf}\left(\sqrt{2}\frac{\alpha}{s}\right)\right) \approx \frac{1}{2} e^{-2\left(\frac{\alpha}{s}\right)^2} \tag{5}$$

where erf denotes the error function.

We want to note that the Hilbert space $\mathcal{H}_2$ (when there is no squeezing) is the same Hilbert space engendered by the states of the [RGM+03] encoding.

The oracle U_f computing the function f is defined by:

$$\forall x \in \{0,1\}^n, y \in \{0,1\}, U_f|\boldsymbol{x}\rangle|\boldsymbol{y}\rangle = |\boldsymbol{x}\rangle|\boldsymbol{f(x)} \oplus \boldsymbol{y}\rangle. \tag{6}$$

We remark that the states $\frac{|\mathbf{0}\rangle \pm |\mathbf{1}\rangle}{\sqrt{2}}$ can be approximated by $|\pm\alpha, s\rangle$. As Eq. (5) suggests, a homodyne measurement in the x-basis followed by a post-selection on the sign of the outcome is an efficient way to distinguish between them. That is why we consider the following algorithm:

1. Create the input state $|\psi_{in}\rangle = |\alpha, s\rangle^{\otimes n}|-\alpha, s\rangle$.
2. Apply U_f to $|\psi_{in}\rangle$. Denote the result $|\psi_{out}\rangle$.
3. Measure each of the first n modes with an homodyne detection.
4. If all the results are positive output constant else output balanced.

The input state is $|\alpha, s\rangle^{\otimes n}|-\alpha, s\rangle$ which can be rewritten:

$$|\alpha, s\rangle^{\otimes n}|-\alpha, s\rangle = \sqrt{p}^n \left(\sum_{\boldsymbol{x}\in\{0,1\}^n} \sigma^{|\boldsymbol{x}|}|\boldsymbol{x}\rangle\right) \otimes \left(\sqrt{\varepsilon}\frac{|\mathbf{0}\rangle + |\mathbf{1}\rangle}{\sqrt{2}} + \sqrt{1-\varepsilon}\frac{|\mathbf{0}\rangle - |\mathbf{1}\rangle}{\sqrt{2}}\right)$$

with $\sigma = \sqrt{\frac{q}{p}}$ and $\varepsilon = \frac{1-\sqrt{1-e^{-4(\frac{\alpha}{s})^2}}}{2}$ and for all binary string s, $|s|$ denotes the Hamming weight of s.

After the oracle application, the state is:

$$|\psi_{out}\rangle = U_f|\psi_i\rangle = \underbrace{\sqrt{\varepsilon}\,\sqrt{p}^n \sum_{\boldsymbol{x}\in\{0,1\}^n} \sigma^{|\boldsymbol{x}|}|\boldsymbol{x}\rangle}_{|\psi_c\rangle} \frac{|\mathbf{0}\rangle + |\mathbf{1}\rangle}{\sqrt{2}} + \underbrace{\sqrt{1-\varepsilon}\,\sqrt{p}^n \sum_{\boldsymbol{x}\in\{0,1\}^n} \sigma^{|\boldsymbol{x}|}(-1)^{f(\boldsymbol{x})}|\boldsymbol{x}\rangle}_{|\psi_f\rangle} \frac{|\mathbf{0}\rangle - |\mathbf{1}\rangle}{\sqrt{2}}$$

The measurement consists on a homodyne measurement on each on the n first modes described by the state $\mathrm{Tr}_t|\psi_{out}\rangle\langle\psi_{out}| = \varepsilon|\psi_c\rangle\langle\psi_c| + (1-\varepsilon)|\psi_f\rangle\langle\psi_f|$ where Tr_t denotes the partial trace on the last mode.

The measurement is described b the POVM $\{\Pi_+, \Pi_-\}$ with

$$\Pi_+ = \bigotimes_{i=0}^{n} \int_{x_i=0}^{+\infty} |x_i\rangle\langle x_i| \mathrm{d}x_i \quad \text{and} \quad \Pi_- = \mathbb{1} - \Pi_+ = \Pi_- \tag{7}$$

Let $|\phi\rangle$ be a n mode state written in the computational basis $|\phi\rangle = \sum_{\boldsymbol{k}\in\{0,1\}^n} c_{\boldsymbol{k}}|\boldsymbol{k}\rangle$. We note $H(\phi) = \mathrm{Tr}[\Pi_+|\phi\rangle\langle\phi|]$ the probability that the result of each homodyne measurement are positive:

$$H(\phi) = \int_{x=x_1\cdots x_n=0}^{+\infty} |\langle x|\phi\rangle|^2 \,\mathrm{d}x = \sum_{\boldsymbol{k},\boldsymbol{l}\in\{0,1\}^n} c_{\boldsymbol{k}} c_{\boldsymbol{l}}^* \prod_{i=1}^{n} \int_{x_i=0}^{+\infty} \langle x_i|\boldsymbol{k}_i\rangle\langle \boldsymbol{l}_i|x_i\rangle \mathrm{d}x \tag{8}$$

and we have:

$$\int_{x_i=0}^{+\infty} \langle x_i|\boldsymbol{k}_i\rangle\langle \boldsymbol{l}_i|x_i\rangle \mathrm{d}x_i = \begin{cases} \frac{1}{2} & \text{if } \boldsymbol{k}_i = \boldsymbol{l}_i \\ \frac{\cos\theta}{2} & \text{otherwise} \end{cases} \quad \text{with} \quad \cos\theta = \frac{\mathrm{erf}\left(\sqrt{2}\frac{\alpha}{s}\right)}{\sqrt{1-e^{-4(\frac{\alpha}{s})^2}}} \tag{9}$$

so we get:

$$H(\phi) = \frac{1}{2^n} \sum_{k,l\in\{0,1\}^n} c_{\boldsymbol{k}} c_{\boldsymbol{l}}^* \cos^{|k\oplus l|}\theta. \tag{10}$$

where $\oplus$ is the bitwise `xor`, $|k\oplus l|$ is the Hamming distance between the binary strings k and l.

After a (long) derivation, we can show that by defining the function $g : \{0,1\}^n \to [-1,1]$ by $g(x) = \sigma^{|x|}(-1)^{f(x)}$, the Fourier coefficient $\hat{g}(\beta)$ appears naturally:

$$H(\psi_f) = \left(\frac{p}{2}\right)^n (1+\cos\theta)^n \sum_{\beta\in\{0,1\}^n} \left(\frac{1-\cos\theta}{1+\cos\theta}\right)^{|\beta|} \hat{g}(\beta)^2. \tag{11}$$

The probability that the result of the homodyne measurement on the first n modes is positive is given by:

$$\mathrm{Tr}[\Pi_+ \mathrm{Tr}_t|\phi_{out}\rangle\langle\phi_{out}|] = \varepsilon H(\psi_c) + (1-\varepsilon)H(\psi_f) \tag{12}$$

Using Fourier analysis on the Boolean cube we can show that the probability of success of the algorith is lower-bounded by $\left(\frac{1}{2} + \sqrt{pq}\cos\theta\right)^n$. This in fact corresponds to f being a constant function.

In both cases, the error is $\mathcal{O}(e^{-\left(\frac{\alpha}{s}\right)^2})$. This is in a deep contrast with the result obtained in [AHS08] where the error cannot be as small as desired by increasing a parameter.

3 Conclusion

We have shown how to solve the Deutsch-Jozsa problem when the information is encoded in CV states. Our algorithm is really simple — theoretically and experimentally: creation of Gaussian states, oracle application and single mode measurements, all the non-Gaussian part being done by the oracle.

Contrary to the usual Deutsch-Jozsa algorithm where the result is deterministic, we only have a probabilistic result but by choosing a high enough energy (*ie.* value of α) we can make this error exponentially small with only single call to the oracle. This is a consequence of the tensor product structure of the input state.

In this framework, the Bernstein-Vazirani and the Simon algorithms can be studied with the same tools and have comparable results.

We can also show that the energy of the input states should be $\mathcal{O}(\sqrt{\ln \frac{n}{\varepsilon}})$ in order to have an error upper-bounded by ε thus giving to the energy the role an important parameter of the problem.

Finally, all this work has been done in Hilbert spaces parameterized by α, but the measurements and the bit flip operation (which is a rotation of angle π in the phase space) do not depend on α. The stability of the problem to the value of α should be studied. In particular, if we use an oracle with a higher value of α than the one it has been design, what happens to the performance of the algorithm? On the one hand the oracle will introduce some noise, but on the other hand the creation and the measurements of the states should give more accurate results.

The bound on the probability of success of our algorithm is tight. Although we are not able to identify the worst-case for balanced function, we conjecture that there is a threshold on the values of α: for low values the function is a dictator, and for high values, f is majority. Resolving this problem would generalize "majority is the stablest" [MOO05] when you remove the condition of low-influence of variables.

References

[AHS08] Mark R. A. Adcock, Peter Høyer, and Barry C. Sanders. On the continuous variable quantum algorithm for oracle identification problems. arXiv:0812.3694v1, 2008.

[BSBN02] Stephen D. Bartlett, Barry C. Sanders, Samuel L. Braunstein, and Kae Nemoto. Efficient classical simulation of continuous variable quantum information processes. *Phys. Rev. Lett.*, 88(9):097904, Feb 2002.

[CLP07] Nicolas J. Cerf, Gerd Leuchs, and Eugene S. Polzik, editors. *Quantum information with continuous variables of atoms and light*. Imperial College Press, 2007.

[DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. In *Mathematical and Physical Sciences*, volume 439, pages 553–558, 1992.

[GKP01] Daniel Gottesman, Alexei Kitaev, and John Preskill. Encoding a qubit in an oscillator. *Phys. Rev. A*, 64(012310), 2001.

[KLM01] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409:46–52, January 2001.

[MOO05] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low in.uences invariance and optimality. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 21–30, Washington, DC, USA, 2005. IEEE Computer Society.

[PB03] A. K. Pati and Samuel L. Braunstein, editors. *Quantum Information with Continuous Variables*. Springer, 2003.

[RC09] Renato Renner and J. Ignacio Cirac. A de finetti representation theorem for infinite dimensional quantum systems and applications to quantum cryptography. *Phys. Rev. Lett.*, 102(11):110504, 2009.

[RGM+03] T. C. Ralph, A. Gilchrist, G. J. Milburn, W. J. Munro, and S. Glancy. Quantum computation with optical coherent states. *Phys. Rev. A*, 68(4):042319, Oct 2003.

A Proof of Eq. (11)

Let $|\psi\rangle = \sum_{\boldsymbol{k}\in\{0,1\}^n} c_{\boldsymbol{k}}|\boldsymbol{k}\rangle$ and note $H(\psi) = \mathrm{Tr}[\Pi_+|\psi\rangle\langle\psi|]$ the probability that the result of each homodyne measurement being positive:

$$H(\psi) = \int_{x=x_1\cdots x_n=0}^{+\infty} |\langle x|\psi\rangle|^2 \,\mathrm{d}x \tag{13}$$

$$= \int_{x=x_1\cdots x_n=0}^{+\infty} \left| \sum_{\boldsymbol{k}\in\{0,1\}^n} c_{\boldsymbol{k}}\langle x|\boldsymbol{k}\rangle \right|^2 \mathrm{d}x \tag{14}$$

$$= \int_{x=x_1\cdots x_n=0}^{+\infty} \sum_{\boldsymbol{k},\boldsymbol{l}\in\{0,1\}^n} c_{\boldsymbol{k}}c_{\boldsymbol{l}}^*\langle x|\boldsymbol{k}\rangle\langle\boldsymbol{l}|x\rangle \mathrm{d}x \tag{15}$$

$$= \sum_{\boldsymbol{k},\boldsymbol{l}\in\{0,1\}^n} c_{\boldsymbol{k}}c_{\boldsymbol{l}}^* \prod_{i=0}^{n-1} \int_{x_i=0}^{+\infty} \langle x_i|\boldsymbol{k}_i\rangle\langle\boldsymbol{l}_i|x_i\rangle \mathrm{d}x \tag{16}$$

and we have:

$$\int_{x_i=0}^{+\infty} \langle x_i|\boldsymbol{k}_i\rangle\langle\boldsymbol{l}_i|x_i\rangle \mathrm{d}x_i = \begin{cases} \frac{1}{2} & \text{if } \boldsymbol{k}_i = \boldsymbol{l}_i \\ \frac{\cos\theta}{2} & \text{otherwise} \end{cases} \quad \text{with} \quad \cos\theta = \frac{\mathrm{erf}\left(\sqrt{2}\frac{\alpha}{s}\right)}{\sqrt{1-e^{-4(\frac{\alpha}{s})^2}}} \tag{17}$$

Proof:
For all β, we first prove that $|\chi_\beta\rangle$ is an eigenvector of M. $\{|\chi_\beta\rangle\}_\beta$ being a basis of $\mathbb{C}^{2^n}$, we found all the eigenvectors of M.

$$M|\chi_\beta\rangle = \frac{1}{\sqrt{2}^n} \sum_{i,j,k\in\{0,1\}^n} \cos^{|i\oplus j|}\theta (-1)^{k\cdot\beta} |\boldsymbol{i}\rangle\langle\boldsymbol{j}|\boldsymbol{k}\rangle \tag{18}$$

$$= \frac{1}{\sqrt{2}^n} \sum_{i\in\{0,1\}^n} \left(\sum_{j\in\{0,1\}^n} \cos^{|i\oplus j|}\theta (-1)^{j\cdot\beta} \right) |\boldsymbol{i}\rangle \tag{19}$$

$$= \frac{1}{\sqrt{2}^n} \sum_{i\in\{0,1\}^n} \left(\sum_{m\in\{0,1\}^n} \cos^{|m|}\theta (-1)^{(m\oplus i)\cdot\beta} \right) |\boldsymbol{i}\rangle \tag{20}$$

$$= \sum_{m\in\{0,1\}^n} \cos^{|m|}\theta (-1)^{m\cdot\beta} |\chi_\beta\rangle \tag{21}$$

We remark that $|\chi_\beta\rangle$ is an eigenvector of M and its eigenvalue is a function of $|\beta|$.

$$\sum_{m\in\{0,1\}^n} \cos^{|m|}\theta (-1)^{m\cdot\beta} = \left(\sum_{m\in\{0,1\}^{n-|\beta|}} \cos^{|m|}\theta \right) \left(\sum_{i=0}^{|\beta|} \binom{|\beta|}{i} (-\cos\theta)^i \right) \tag{22}$$

$$= (1+\cos\theta)^{n-|\beta|} (1-\cos\theta)^{|\beta|} \tag{23}$$

So we get:

$$H(\psi) = \frac{1}{2^n} \sum_{k,l\in\{0,1\}^n} c_{\boldsymbol{k}} c_{\boldsymbol{l}}^* \cos^{|k\oplus l|}\theta. \tag{24}$$

$\oplus$ being the bitwise `xor`, $|k\oplus l|$ is the Hamming distance between the binary strings k and l.

Let us now give the meaning of $H(\psi_f)$:

$$H(\psi_f) = \left(\frac{p}{2}\right)^n \sum_{k,l\in\{0,1\}^n} \sigma^{|k|+|l|} (-1)^{f(k)+f(l)} \cos^{|\boldsymbol{k}\oplus\boldsymbol{l}|}\theta \tag{25}$$

$$= \left(\frac{p}{2}\right)^n \langle v|M|v\rangle \tag{26}$$

with $|v\rangle = \sum_{x\in\{0,1\}^n} \sigma^{|x|}(-1)^{f(x)}|x\rangle$ and $M = \sum_{k,l\in\{0,1\}^n} \cos^{|k\oplus l|}\theta |\boldsymbol{k}\rangle\langle\boldsymbol{l}|$. The matrix M can be diagonalized (proof in appendix **??**):

$$M = (1+\cos\theta)^n \sum_{\beta\in\{0,1\}^n} \rho^{|\beta|} |\chi_\beta\rangle\langle\chi_\beta| \tag{27}$$

where $\{\chi_\beta\}_\beta$ is the Fourier basis: $|\chi_\beta\rangle = H^{\otimes n}|\boldsymbol{\beta}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\cdot\beta}|\boldsymbol{x}\rangle$ and $\rho = \frac{1-\cos\theta}{1+\cos\theta}$. That gives:

$$H(\psi_f) = \left(\frac{p}{2}\right)^n (1+\cos\theta)^n \sum_{\beta\in\{0,1\}^n} \rho^{|\beta|} |\langle v|\chi_\beta\rangle|^2. \tag{28}$$

Defining the function $g : \{0,1\}^n \to [-1,1]$ by $g(x) = \sigma^{|x|}(-1)^{f(x)}$, the quantity $\langle v|\chi_\beta\rangle$ has an easy interpretation as the Fourier coefficient $\hat{g}(\beta)$.

$$H(\psi_f) = \left(\frac{p}{2}\right)^n (1+\cos\theta)^n \sum_{\beta\in\{0,1\}^n} \rho^{|\beta|} \hat{g}(\beta)^2. \tag{29}$$

Towards a Physical Implementation of P Systems: Photo-switching Molecules as Logic Gates and Registers

J. C. Chaplin[1,2], N. Krasnogor[1], and N. Russell[2]

[1] Automated Scheduling, Optimisation and Planning Research Group, School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB
jcc@cs.nott.ac.uk

[2] Institute of Biophysics, Imaging and Optical Science, School of Biology, University of Nottingham, University Park, Nottingham, NG7 2RD

Abstract. This paper explains progress towards the construction of a P System utilising these components, as well as the experimental set-up used to achieve these results. A photochromic spiropyran dye - NitroBIPS - has been used in conjunction with optical stimulation to produce NOR and NAND gates as well as numerical registers.

1 Introduction

Computing is all around us, from the largest supercomputers to predict weather patterns[1], to the smallest smart phones that keep us communicating. Unconventional methods of computing have been explored for sometime, including Chemical methods to produce solve combinatorial problems[2] or create logic gates[3], optical methods[4] or an approach in the middle that uses light to stimulate a substance[5]. This project aims to make inroads into a physical implementation of a Membrane Computer or 'P System' (Named for their creator, Gheorghe Păun[6]) using a similar combined approach of light stimulation and photo-switching molecules. P Systems are a model of computation initially inspired by the simulation of cellular modelling where multisets of symbols are evolved in parallel within membrane-enclosed compartments by a set of rules. Symbols can move between membranes, mimicking the flow of chemicals in biological cells and membranes can be created and destroyed. P Systems are well researched, with many variations and formulations[7–9], and have been shown to be flexible and universal in certain forms[10], and super-Turing in others; able to solve NP-Complete problems in linear time via membrane division[11, 12].

P-Systems consist of nested membranes and the associated regions contained within these membranes. Each region contains some objects in a multiset, and some rules on how these objects interact and change over time. This can be compared to cells as the membrane-separated compartments of cells containing various molecules, and these molecules being subject to chemical reactions that change and transport them. A more formal definition of a basic P System is thus:

A P-System P of degree m (where the degree is the number of discrete membranes) is a tuple:
$P = \{V, \mu, w_1, w_2, \ldots, w_m, R_1, R_2, \ldots, R_m, i_0\}$ where:
V is a finite and non-empty alphabet of objects representing the possible contents of cells.
μ is the nested structure of membranes, typically represented as either a tree or as brackets, with each membrane being labelled with a unique number. E.g.

$$[_1[_2[_3]_3]_2]_1$$

w_x are multi-sets of objects that begin in each membrane's region
R_x are the sets of rules in each membrane. E.g.

$$R_2 : \{aa \to b, b \to c, a \to cd\}$$

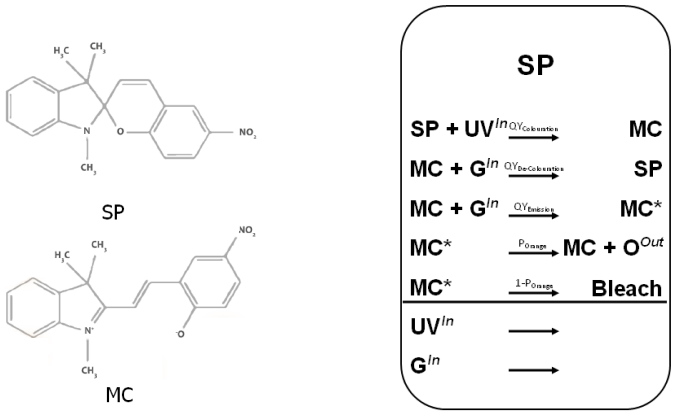
i_0 is the membrane label of the output region.

At the end of execution, the content of the output region is the result of the computation. Rules in a P-System are typically executed non-deterministically, in a maximally parallel fashion, or via a priority hierarchy. The latter requires that the rules are given an order of preference, and in a series of discrete steps, the rules are checked in order until one is found for which the necessary inputs are present, and is executed. The process then starts over. Non-deterministic execution randomly selects rules, checks if they are applicable, and chooses another if not. Maximally parallel execution will execute many rules simultaneously, to use up as much of the available objects as possible each time step. To our knowledge, no physical implementation of a P System has been completed, so we aim to produce the simplest possible physical P System implementation.

There exist many substances which react to light, from photosensitive Belousov-Zhabotinsky media[5] to neurons[13]. We sought a substance that would require minimal preparation work, ruling out neurons, and which could save a state for a reasonable time period. Though BZ-Reactions have been shown to be capable of data storage in the form of images[14], and indeed logical functions[5], we looked for a means to save integer numbers of objects as opposed to analogue images to better implement a P System. We are utilising a photochromic spiropyran dye[15, 16] - NitroBIPS or NBIPS - available from Sigma-Aldrich[3]. NitroBIPS possesses two stable states; Spiropyran (SP) and Merocyanine (MC), and can be switched between the two via the application of two wavelengths of light; Ultraviolet to switch from SP to MC (sometimes called 'colouration' or 'photocolouration'), and Green to switch from MC to SP ('de-colouration'). Additionally, a proportion of green light directed at MC state molecules will cause the molecule to enter a third 'MC*' state, followed by a return to the MC state and the emission of an orange photon in the 630nm range, or to enter a bleached state from which the NBIPS molecule is unresponsive. This feature allows us to

[3] More fully 1',3'-Dihydro-1',3',3'-trimethyl-6-nitrospiro[2H-1-benzopyran-2,2'-(2H)indole][17], Sigma-Aldrich product code 273619.

determine the proportion of SP and MC state molecules in a given sample of NBIPS, as only MC state molecules will emit orange light. Though not all MC state molecules will emit a photon, the quantum yield is fixed dependant on the solvent, allowing us to calculate the true amount.



(a) (b)

Fig. 1. In figure (a) we see the two states of NitroBIPS. On the top is the more stable Spiropyran form. Below is the Merocyanine form. Figure (b) is the mechanism of NBIPS expressed as a P System on a per-molecule basis, starting in this case with a single molecule in state SP and in which only photons absorbed by NBIPS molecule are considered. $SP, MC, MC*$ and $Bleach$ are objects representing NBIPS in different states, G^{In} and UV^{In} are green and ultraviolet photons respectively which enter the system from the environment, O^{Out} is an orange photon leaving the system. The rules have probabilities attached to them; not every absorbed photon will cause a change. $QY_{Colouration}$ is the proportion of absorbed UV photons that cause SP molecules to change to MC, $QY_{De-Colouration}$ is the opposite, $QY_{Emission}$ is the odds of a MC state molecule entering the MC* state in response to a green photon, and P_{Orange} is the proportion of MC* state molecules that emit orange photons instead of bleaching. The two rules below the line are of a lower priority to those above the line, and represent the fact that photons do not remain in the system if they are not used.

2 A NitroBIPS P System Implementation

2.1 Rule Application via Logic Gates

Any computer in general requires both processing elements and data storage elements. This section describes the first of these elements, a processor of some

sort. Specifically in this case, a set of logic gates. As shown, P System rules operate on a symbol consumption/production paradigm governed by rules. As our system operates with numerical registers (next section) we can operate on symbols via addition/subtraction mathematics.

There exists a wide variety of logic gates, but of the most commonly known class - two input/one output - only two are functionally complete in isolation; NAND and NOR - sometimes called being a sole sufficient operator - and hence are the minimum building blocks to produce a general-purpose computing device. Some sets of logic gates can also be logically complete; the most common being a two-element set of NOT and either AND or OR. A series of functionally complete logic gates is able to simulate any other logic gate of the same class. Both NAND and NOR gates (as well as others) are possible to implement with NitroBIPS and pulses of light. Different patterns of light create different results. Operation begins with the NitroBIPS in a fully SP state. For both NAND and NOR gates, some UV light is then used to prepare the gate, two pulses of green light are then used subject to the value of the inputs, and then a third pulse of green light is used to check the output and reset the gate. To describe these pulses, the following notation is used:

The pulses alter the number of molecules in SP and MC state. Pulses are measured in 'units' where one unit of light is the minimum number of photons required to cause a change in the state of the well that can be discerned by our system, a figure reliant primarily on system noise. Pulses are shown as $nL(\pm x)$, where n is the number of units of L light, and x is the change in the number of MC state molecules, which may be zero if there are no appropriate state molecules to convert, shown as -. As the number of MC and SP molecules are linked, +1 MC state molecules also corresponds to -1 SP state molecule, but this is omitted in the tables for clarity. This system is expressed more intuitively as a P System in figure 2 (b).

AND Gate

Input 1	Input 2	UV Input 1	UV Input 2	Green Moderator	Green Check/Reset	Orange Output
0	0	-	-	1G(-)	1G(-)	-
1	0	1UV(+1)	-	1G(-1)	1G(-)	-
0	1	-	1UV(+1)	1G(-1)	1G(-)	-
1	1	1UV(+1)	1UV(+1)	1G(-1)	1G(-1)	1

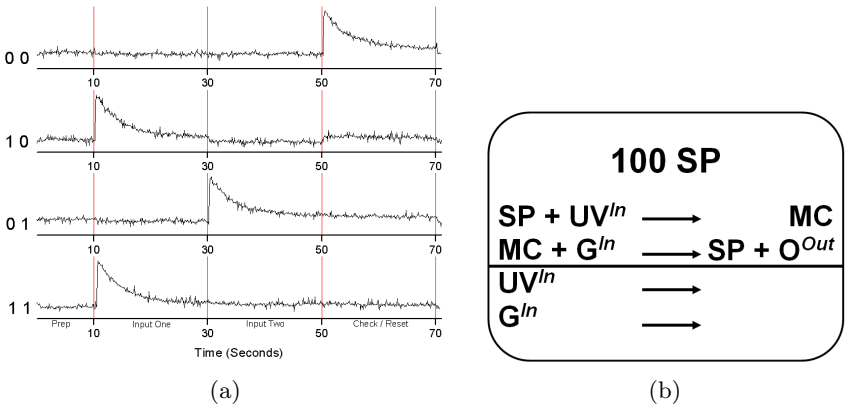
NAND Gate

Input 1	Input 2	UV Pulse	Green Input 1	Green Input 2	Green Check/Reset	Orange Output
0	0	2UV(+2)	-	-	2G(-2)	2
1	0	2UV(+2)	1G(-1)	-	2G(-1)	1
0	1	2UV(+2)	-	1G(-1)	2G(-1)	1
1	1	2UV(+2)	1G(-1)	1G(-1)	2G(-)	-

NOR Gate

Input 1	Input 2	UV Pulse	Green Input 1	Green Input 2	Green Check/Reset	Orange Output
0	0	1UV(+1)	-	-	1G(-1)	1
1	0	1UV(+1)	1G(-1)	-	1G(-)	-
0	1	1UV(+1)	-	1G(-1)	1G(-)	-
1	1	1UV(+1)	1G(-1)	1G(-)	1G(-)	-

AND is included here as an example of how other gates can be constructed, and how it operates on a different principal to the NAND and NOR gates. Similarly, OR and NOT can be produced; OR by omitting the moderator pulse in the AND pattern, and NOT by changing the NOR gate to have a single input. It should be noted that the gates do emit orange photons during the input section. For the output collected to be correct, only emission during the check/reset pulse should be counted. This may prove a problem depending on the set-up, the NAND gate would work as a NOR gate if only two units of orange light counts as a positive output, but with this method the orange emitted during the input phase when having both inputs as true might be incorrectly regarded as a positive output. It should also be noted that the NAND and OR gates actually have three different outputs; False, True and Double True. However, the NOR gate strains the NitroBIPS less per cycle and takes less time to run, so this is the gate we have moved forwards with.



(a) (b)

Fig. 2. (a) Four traces corresponding to the four combinations of inputs into the gate. (b) A modified version of figure 1 (b) representing the operations possible on a sample of NBIPS, where symbols represent large numbers of photons or molecules rather than individuals, and where the sample has 101 potential states (0 - 100 MC-state molecule units). The Gate Description tables utilise these rules.

2.2 Symbol and Rule Storage as Registers

As mentioned, data storage in a P System takes the form of symbols, and each membrane contains a multiset of symbols. If we assign one register per type of object, the register need only contain the quantity of that object. NBIPS registers operate as a proportion of SP to MC molecules. If a register has X molecules, and it takes a change in Y molecules to be able to distinguish state n from $n+1$, then a register can have $(X/Y)+1$ states. Change between states is via the application of UV light to increase the proportion of MC molecules (and decrease SP molecules), and green to increase the number of SP molecules (and decrease MC). Checking the state requires a short burst of green light, causing the emission of some orange light and also the de-colouration of some molecules. By keeping the pulse short, the number of molecules converted can be kept to a minimum. If necessary, a rectifying UV pulse can be used to undo the damage done by the check pulse.

The number of photons emitted will be proportional to the state of the register. If you know how many photons are emitted when the register is in a maximally-coloured state by an equal length and intensity green pulse, then the response of each register state is proportional.

$$S_n = (S_{Max}/P_{Max}) \times P_n$$

Where S is a state, and P is the number of photons emitted per pulse in that state. Current progress allows for 53 states in a register, a figure which will increase as we work to decrease system noise.

2.3 Logical Style

In contrast to the operation of electrical logic gates, whose inputs operate in a parallel fashion, the opto-chemical gates of this system operate in a serial manner, accepting inputs one after another. This results in several important differences to classical logic gates.

Firstly, the gate requires a preparation stage to ready the gate in the form of a pulse of UV light to convert some SP molecules to the MC state, as well as a check/reset pulse to extract the output and ready the gate for another run. In between these two bookend pulses lies the input pulses; typically two but could be any number; especially with NOR gates which scale to any number of inputs under this paradigm. Though our current gates are not even within the same order of magnitudes of speed as modern electronic logic gates, it stands to reason that the gate delay on opto-chemical gates will be much larger due to the extra work required to prepare and reset the gate.

However, the serial nature of the gate does add some interesting capabilities similar to that of sequential logic. The two (or more) inputs need not arrive at the same time, for example, nor in any particular order. Provided the inputs arrive within the 'clock cycle' of the preparation and check/reset pulse, the result will remain accurate. The output of the gate need not be read immediately either, but will remain stored in the gate until the check/reset pulse akin to a flip-flop.

A second difference is that the gates of the opto-chemical system have no physical form per-se, being as they are a region of NitroBIPS exposed to light. This means the pattern of light pulses can be altered, and the type of gate changed. NOR gates are an obvious choice under this system as they work well, provide true/false outputs and are functionally complete. Despite this, other gates can be made by altering the pattern of lights. Similarly, an interesting facet of the opto-chemical system is that both memory and logic gates are interchangeable. As previously described, NitroBIPS can function as both storage and as logic gates, but the difference is only the patterns of light shined upon a section of the material. Though conventional logic gates can function as storage when wired as flip-flops or latches, these can only store one bit per device, and need to be wired specially.

The third issue, and a clear negative is that the output of a gate is unsuitable to be used as a direct input to another, preventing the cascading of gates. Cascading is the feature of electronic logic gates where the input and output are of the same format, allowing outputs to be wired directly to the inputs of other logic gates to process functions without the need for an ancillary system to mediate between each gate. With the NBIPS logic gates, not only is the output light of the wrong wavelength, but far far too dim to be useful. Instead, the controlling computer is required to mediate and convert the output light back into an appropriate pulse. The computer is also necessary to regulate and time the preparation and check/reset pulses. Hence in its present form, the opto-chemical computer cannot be a stand-alone system, and will remain reliant on its parent conventional computer.

3 Experimental Section

3.1 NitroBIPS Characterisation

To use the NitroBIPS, we must first understand how it reacts. An optical setup and accompanying software control were created to expose samples of NitroBIPS to wavelengths of light and measure the photon output. NitroBIPS is stored in powdered form, and although it will react to light in its dry state, its opaque nature makes use difficult (light will only hit the exterior molecules) and the quantum yield is low. Instead, NitroBIPS should be dissolved into a solvent to allow control over concentration, and to allow optical access to all molecules. Testing of solvents showed that pure Ethanol is successful, cheap and safe. However, it also evaporates quickly, requiring a well-sealed well, and has a poor quantum yield. NitroBIPS would not dissolve in some other solvents with slower evaporation times, such as Glycerol. Dissolution in Methylcyclohexane or Toluene increases the quantum yield of colouration[18], but have not been explored yet.

Fluid NitroBIPS is useful, but for our purposes it was necessary to address different parts of NitroBIPS. With a liquid, this would require tiny wells of NitroBIPS to defeat diffusion, a difficult and complex enterprise. Instead, NitroBIPS was first mixed in a high concentration (2mM) with Methanol, and then

mixed into uncured Polydimethylsiloxane (PDMS) silicone rubber at a ratio of 5% NitroBIPS to 95% PDMS. Thorough mixing ensures that tiny bubbles of NitroBIPS are distributed evenly throughout the rubber, which subsequently cures into a transparent sheet[4]; we aim for a thickness of 0.5mm. Mixing with PDMS does not entirely eliminate the evaporation of the methanol from the mixture, but is simpler than fluid wells and will last 3-5 days out in the open depending on how heavily it is used (the heating effect of the light from the LEDs promotes evaporation).

The optical set-up features a 365nm UV LED and a 530nm Green LED to switch NitroBIPS between the two stable states. Light from these LEDs is combined with a dichroic mirror and the sample exposed with uniform Köhler illumination. Both LEDs are computer controlled in both on/off state and brightness, and can be manually adjusted to change the size of the projected disc. Emissions from the sample are separated from any reflected input light via a second dichroic mirror and are captured by a high sensitivity photodioide and the readings recorded on a computer. The system is controlled by a LabVIEW program

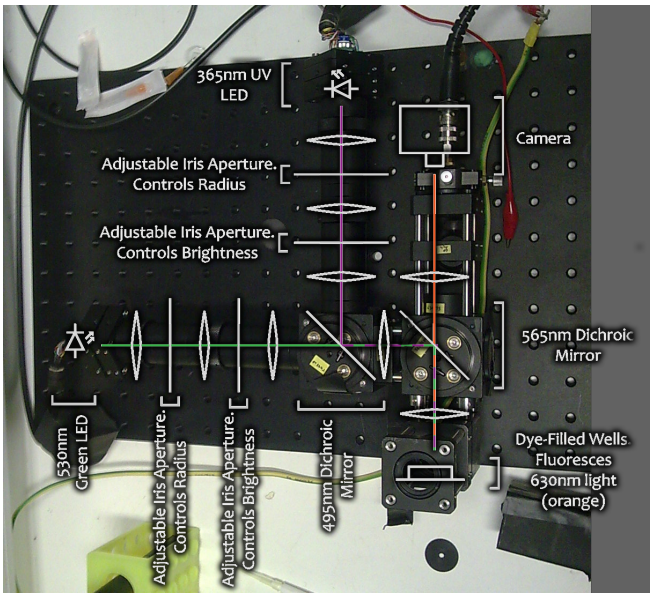


Fig. 3. The physical system. Two LED arms combine light via a dichroic to expose the sample. Orange light is filtered by a second dichroic and recorded by a photodiode or camera.

running on a computer coupled with a National Instruments Data Acquisition card, which allows precise control of all system inputs and recording of all system

[4] With thanks to Gerard Marriott for his suggestion

outputs. The low amount of orange light emitted by NitroBIPS coupled with a low collection rate to the diode necessitates heavy signal amplification and noise reduction/filtration. A hardware filter is used to remove most environmental noise, the system is grounded to an electrical earth, and experiments take place in a photography darkroom (without the photography red safety light).

A few issues have been identified with NitroBIPS. NitroBIPS changes states slowly in Methanol. It requires a large quantity of light for the system to function at any useful speed, though there exist methods to improve this. One method is placing a mirror on top of the sample. The mirror will reflect light which has not been absorbed back through the NitroBIPS, giving it a second chance to interact with NBIPS molecules. The mirror also directs more emitted orange light into the collection lens; orange photons from NBIPS fire in all directions with an equal probability, and only a small proportion of these are collected. The mirror directs photons which were fired upwards back down towards the lens under the NBIPS sample.

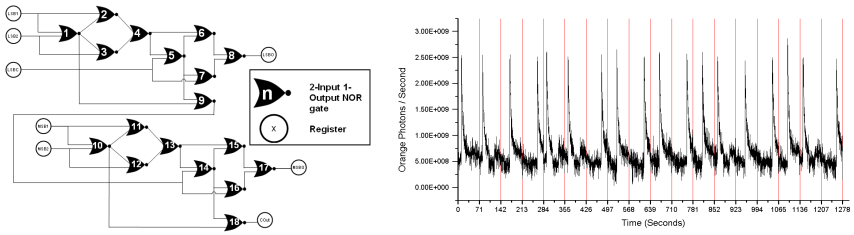
A second more complex method would be a change of solvent. Methanol is not the ideal solvent for NitroBIPS; others grant a higher quantum yield but the Methanol is required to mix with the PDMS. Other solvents either inhibit the curing of the PDMS, or evaporate before the PDMS is cured. PDMS also partially attenuates UV light, reducing the number of UV photons interacting with the NitroBIPS molecules. Using thin sheets of PDMS helps in this respect, as these is less material for photons to pass through. It is also necessary to strike a balance between high and low concentrations of NBIPS. Too high and the optical density of the material increases and photons are less likely to reach NitroBIPS molecules at the back of the sample. Too low and photons will pass through the sample without ever interacting with a NitroBIPS molecule.

NitroBIPS also bleaches over time. Bleaching occurs when green photons interact with MC-state molecules with a probability of approximately 0.17% per interaction, and leaves the molecule in a third unresponsive state which will not further react to light. NitroBIPS will also thermally decay at a rate dependant on ambient temperature, but at room temperature ($20°C$) has a decay constant of 0.000068, and a corresponding half-life of 10,193 seconds (2 hours 43 minutes). Thermal transitions take place in both colouration and de-colouration directions, but trending towards de-coloured as the de-colouration transition requires less energy. In general, this is slow enough to not be a concern unless data is being stored long term, in which case measures to decrease the temperature should be taken or by storing values on the computer if absolutely necessary. With our set-up, it is also not possible to place the system in a fully MC-state; the UV LED emits some light in the range of wavelengths that cause a MC to SP switch, causing some orange emission during UV pulses, and causing a proportion of NBIPS molecules to be in the SP state.

3.2 Results

Our present system operates with a single gate, exposing it to a UV preparation pulse, then green input pulses, a final green check/reset pulse and a single second

gap between logic gates to save the output of a gate and check the inputs to the next gate. The four traces of orange emission are shown in figure 3 (b), along with their corresponding truth table entries. Like all logic gates, NBIPS Logic Gates can be chained together to produce more complex devices, though in our current case, this actually means using a single gate repeatedly with the computer storing the outputs of gates and converting them to the inputs of subsequent gates. For example, we have used eighteen NOR gates to produce a 2-bit adder, as shown in figure 4.



(a) (b)

Fig. 4. (a) A two-bit NOR adder. The least significant bit (LSB) is added in the upper section, and the most significant bit (MSB) in the lower. The LSB is worth 1, the MSB worth 2, and the CarryOut - while technically an overflow - would be worth 4 if there was another full-adder for it to feed to. The numbers represent the order in which gates are executed. (b) Complete trace of the 2-bit NOR adder. Each interval represents one logic gate. The calculation taking place here is $3 + 1$, and we can see that LSB Output gate 8 (time 497-568) and MSB Output gate 17 (1136-1207) have output 0, where as the CarryOut gate 18 (1207-1278) has output 1. Hence by referring to (a), $3 + 1 = 4$ if we consider the CarryOut as a valid output. Total execution time was 21 minutes and 18 seconds.

4 Future Work

Two points limit the complexity of possible devices. Firstly, NBIPS bleaches, reducing the number of active molecules over time, causing the gates to produce smaller amounts of orange emission as they are used. This must be accounted for by lengthening the pulses of light as the system continues. Secondly, the present system takes a long time to run even before any pulses are lengthened to account for bleaching, adding a practical limit to how complex devices can be. Presently, the length of pulses is determined at the start of computation and kept constant. To increase the time available to process, the system needs to be updated to self-calibrate periodically to account for bleaching, altering pulse length to minimise gate execution time, but at the same time allowing for the decrease in orange response.

Creating a means to apply symbol rewriting rules using the minimal number of NOR gates is also current aim of the project. The logic system created would need to be able to interpret stored rules and instruct the computer in how to apply them. Additionally, we aim to use the flexible nature of NBIPS gates and registers to simulate membranes in the system, where the rules and registers of a membrane can be moved, erased or created by light, and arranged spatially on a device. The membranes are thus virtual constructs as opposed to having a physical form. Using NitroBIPS in liquid form utilising a microfluidics device may also be explored as an alternative to PDMS discs, as it more closely mimics the movement of symbols in a P System. The ability to delay the input and output of NBIPS gates could potentially be used to store partially computed functions, an idea we hope to explore. System noise limits the speed at which the system can run, and we plan on building a faraday cage to protect our system from electromagnetic interference, and have the laboratory's power supply smoothed or switch the system to battery power; both of which would dramatically reduce system noise. Finally, our current system can only expose the entire NBIPS sample. By utilising a spatial light modulator (SLM), we could direct light to specific parts of the sample, and address multiple locations in parallel.

5 Conclusion

We have made inroads into implementing a P system utilising a photo-switching molecule; NitroBIPS. Registers and many logical operators have been implemented and shown to work, including both 2-input/1-output sole sufficient operators. Current execution speeds are slow due to high system noise from electromagnetic fields and imperfections in the power supply, leading to the requirement of long light pulses to convert larger quantities of NitroBIPS. The amount of light required is dependent on system noise; very low system noise allows you to get away with smaller pulses of light, and hence quicker system operation. The system also remains heavily reliant on the controlling computer, with no obvious means by which to remove it. The system also cannot cascade, as unlike electronic logic gates whose input and outputs are of the same format, the NBIPS gates take two wavelengths of light as preparation and input and emit a third with a dramatically lower intensity without direction, requiring an intermediary converter. The need for a preparation pulse is not the problem here, as gates can be pre-prepared for inputs. The method however, would scale down to very small quantities of NBIPS; not quite one molecule as the emission of orange light is not definite and the detection of the single photon very hard, but small non-the-less. This method also has some interesting features such as the non-fixed nature of the registers and logic gates; they can be placed anywhere on a sample of NBIPS, and altered/erased at will. This could allow for a method of computation that can sacrifice storage space for increased parallel processing power, or reduce its processing power for more storage space. NBIPS logic gates can also accept inputs at different times, and retain state until the second input arrives, and save the output state until a check pulse is given, potentially allowing for partially executed functions to be stored.

References

1. T. Sato, "The earth simulator: Roles and impacts," *Nuclear Physics B - Proceedings Supplements*, vol. 129-130, pp. 102 – 108, 2004.
2. L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.
3. D. Margulies, G. Melman, and A. Shanzer, "A molecular full-adder and full-subtractor, an additional step toward a moleculator," *Journal of the American Chemical Society*, vol. 128, no. 14, pp. 4865–4871, 2006.
4. X. Li, Y. Wu, D. Steel, D. Gammon, T. Stievater, D. Katzer, D. Park, C. Piermarocchi, and L. Sham, "An all-optical quantum gate in a semiconductor quantum dot," *SCIENCE*, vol. 301, no. 5634, pp. 809–811, 2003.
5. R. Toth, C. Stone, A. Adamatzky, B. de Lacy Costello, and L. Bull, "Experimental validation of binary collisions between wave fragments in the photosensitive belousov-zhabotinsky reaction," *Chaos, Solitons and Fractals*, vol. 41, no. 4, pp. 1605–1615, 2009.
6. G. Paun, "Computing with membranes," *Journal of Computer and System Sciences*, vol. 61, pp. 108–143, 1998.
7. R. Freund, G. Paun, and M. J. Prez-Jimnez, "Tissue p systems with channel states," *Theoretical Computer Science*, vol. 330, no. 1, pp. 101 – 116, 2005.
8. F. Bernardini and M. Gheorghe, "Population p systems," *j-jucs*, vol. 10, no. 5, pp. 509–539, 2004.
9. R. Freund and M. Oswald, "Membrane systems with symport/antiport: Universality results," in *in Membrane Computing. Intern. Workshop WMC-CdeA2002, Revised Papers*, pp. 270–287, Springer-Verlag, 2002.
10. P. Frisco, H. Hoogeboom, and P. Sant, "A direct construction of a universal p system," *Fund. Inform.*, vol. 49, pp. 103–122, 2002.
11. G. Paun, "P systems with active membranes: attacking np compete problems," *Automata, Languages and Combinatorics*, vol. 6, no. 1, pp. 75–90, 2001.
12. C. Zandron, C. Ferretti, and G. Mauri, "Solving np complete problems using p systems with active membranes," in *Unconventional Models of Computation*, pp. 289–301, Springer, 2000.
13. H. Hirase, J. H. Goldberg, and R. Yuste, "Multiphoton stimulation of neurons," *Journal of Neurobiology*, vol. 51, no. 3, pp. 237–247, 2002.
14. A. Kaminaga, V. K. Vanag, and I. R. Epstein, "A reaction-diffusion memory device," *Angewandte Chemie*, vol. 118, no. 19, pp. 3159–3161, 2006.
15. I. Willner, "Photoswitchable biomaterials: En route to optobioelectronic systems," *Accounts of Chemical Research*, vol. 30, no. 9, pp. 347–356, 1997.
16. T. Sakata, Y. Yan, and G. Marriott, "Optical switching of dipolar interactions on proteins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 13, pp. 4759–4764, 2005.
17. G. Marriott, S. Mao, T. Sakata, J. Ran, D. K. Jackson, C. Petchprayoon, T. J. Gomez, E. Warp, O. Tulyathan, H. L. Aaron, E. Y. Isacoff, and Y. Yan, "Optical lock-in detection imaging microscopy for contrast-enhanced imaging in living cells," *Proceedings of the National Academy of Sciences*, 2008.
18. G. H and M. S, "Photochromism of nitrospiropyrans: Effects of structure, solvent and temperature," *Physical Chemistry Chemical Physics*, vol. 3, pp. 416–423, 2001.

Program-size versus Time complexity

Slowdown and speed-up phenomena in the micro-cosmos of small Turing machines

Joost J. Joosten[1], Fernando Soler-Toscano[1], and Hector Zenil[2,3]

[1] Grupo de Lógica, Lenguaje e Información
Departamento de Filosofía, Lógica, y Filosofía de la Ciencia
Universidad de Sevilla
{jjoosten,fsoler}@us.es

[2] Laboratoire d'Informatique Fondamentale de Lille
(CNRS), Université de Lille I
[3] Wolfram Research, Inc.
hectorz@wolfram.com

Abstract. The aim of this paper is to undertake an experimental investigation of the trade-offs between program-size and time computational complexity. The investigation proceeds by an exhaustive exploration and systematic study of the functions computed by the set of all 2-color Turing machines with 2, and 3 states with particular attention to the runtimes, space-usages and patterns corresponding to the computed functions when the machines have access to larger resources (more states).
We report that the average runtime of Turing machines computing a function almost surely increases as a function of the number of states, indicating that machines not terminating (almost) immediately tend to occupy all the resources at hand. We calculated all time complexity classes to which the algorithms computing the functions found in both (2,2) and (3,2) belong to, and made comparison among these classes.
Our study revealed various structures in the micro-cosmos of small Turing Machines. Most notably we observed "phase-transitions" in the halting-probability distribution.
Keywords: small Turing machines, Program-size complexity, Kolmogorov-Chaitin complexity, space/time complexity, computational complexity, algorithmic complexity.

1 Introduction

Among the several measures of computational complexity there are measures focusing on the minimal description of a program and others quantifying the resources (space, time, energy) used by a computation. This paper is a reflection of an ongoing project with the ultimate goal of contributing to the understanding of relationships between various measures of complexity by means of computational experiments.

1.1 Two measures of complexity

The long run aim of the project focuses on the relationship between complexity measures, particularly descriptional and computational complexity measures. In this subsection we shall briefly and informally introduce them.

In the literature there are results known to theoretically link some complexity notions. For example, in [6], runtime probabilities were estimated based on Chaitin's heuristic principle as formulated in [5]. Chaitin's principle is of descriptive theoretic nature and states that *the theorems of a finitely-specified theory cannot be significantly more complex than the theory itself.*

Bennett's concept of logical depth also combines the concept of time complexity and program-size complexity [1, 2] by means of the time that a decompression algorithm takes to decompress an object from its shortest description.

Recent work by Neary and Woods [14] has shown that the simulation of cyclic tag systems by cellular automata is effected with a polynomial slow-down, setting a very low threshold of possible non-polynomial tradeoffs between program-size and computational time complexity.

Computational Complexity Computational complexity [4, 9] analyzes the difficulty of computational problems in terms of computational resources. The computational time complexity of a problem is the number of steps that it takes to solve an instance of the problem using the most efficient algorithm, as a function of the size of the representation of this instance.

As widely known, the main open problem with regard to this measure of complexity is the question of whether problems that can be solved in non-deterministic polynomial time can be solved in deterministic polynomial time, aka the P versus NP problem. Since P is a subset of NP the question is whether NP is contained in P. If it is, the problem may be translated as, for every Turing machine computing an NP function there is (possibly) another Turing machine that does so in P time. In principle one may think that if in a space of all Turing machines with a certain fixed size there is no such a P time solving machine for the given problem (and because a space of smaller Turing machines is always contained in the larger) only by adding more resources a more efficient algorithm, perhaps in P, might be found.

Descriptional Complexity The algorithmic or program-size complexity [8, 5] of a binary string is informally defined as the shortest program that can produce the string. There is no algorithmic way of finding the shortest algorithm that outputs a given string

The complexity of a bit string s is the length of the string's shortest program in binary on a fixed universal Turing machine. A string is said to be complex or random if its shortest description cannot be much more shorter than the length of the string itself. And it is said to be simple if it can be highly compressed. There are several related variants of algorithmic complexity or algorithmic information.

In terms of Turing machines, if M is a Turing machine which on input i outputs string s, then the concatenated string $\langle M, i\rangle$ is a description of s. The

size of a Turing machine in terms of the number of states (s) and colors (k) (aka known as symbols) is determined by the product $s \cdot k$. Since we are fixing the number of colors to $k = 2$ in our study, we increase the number of states s as a mean for increasing the program-size (descriptional) complexity of the Turing machines in order to study any possible tradeoffs with any of the other complexity measures in question, particularly computational (time) complexity.

1.2 Turing machines

Throughout this project the computational model will be that of Turing machines. Turing machines are well-known models for universal computation. This means, that anything that can be computed at all, can be computed on a Turing machine.

In its simplest form, a Turing machine consists of a two-way infinite tape that is divided in adjacent cells. Each cell can be either blank or contain a non-blank color (symbol). The Turing machine comes with a "head" that can move over the cells of the tape. Moreover, the machine can be in a different state. At each step in time, the machine reads what color is under the head, and then, depending on in what state it is writes a (possibly) new color in the cell under the head, goes to a (possibly) new state and have the head move either left or right. A specific Turing machine is completely determined by its behavior at these time steps. One often speaks of a transition rule, or a transition table. Figure 1 depicts graphically such a transition rule when we only allow for 2 colors, black and white.

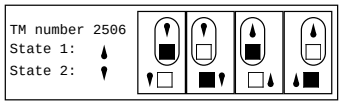


Fig. 1. Transition table of a 2-color 2-state Turing machine with rule 2506 according to Wolfram's enumeration and Wolfram's visual representation style [12].

For example, the head of this machine will only move to the right, write a black color and go to state 2 whenever the machine was in state 2 and it read a blank symbol.

1.3 Relating notions of complexity

We relate and explore throughout the experiment the connections between descriptional complexity and time computational complexity. One way to increase the descriptional complexity of a Turing machine is enlarging its transition table description by adding a new state. Our current findings suggest that even if a more efficient Turing machine algorithm solving a problem instance may exist, the probability of picking a machine algorithm at random solving the problem

in a faster time has probability close to 0 because the number of slower Turing machines computing a function outnumbers the number of possible Turing machines speeding it up by a fast growing function.

This suggests that the theoretical problem of P versus NP might be disconnected to the question in practice when using brute force techniques. Disregarding the answer to the P versus NP as a theoretical problem, efficient heuristics to search for the P time algorithm may be required, other than picking it at random or searching it by exhaustive means, for otherwise the question in practice may have a different answer in the negative independent of the theoretical solution. We think our approach provides insights in this regard.

1.4 Investigating the micro-cosmos of small Turing machines

We know that small programs are capable of great complexity. For example, computational universality occurs in cellular automata with just 2 colors and nearest neighborhood (Rule 110) [12, 3] and also (weak) universality in Turing machines with only 2-states and 3-colors [13].

For all practical purposes one is restricted to perform experiments with small Turing machines (TMs) if one pursuits a thorough investigation of complete spaces for a certain size. Yet the space of these machines is rich and large enough to allow for interesting and insightful comparison, draw some preliminary conclusions and shed light on the relations between measures of complexity.

To be more concrete, in this paper, we look at TMs with 2 states and 2 colors and compare them to TMs with 3 states and 2 colors. The main focus is on the functions they compute and the runtimes for these functions[4]. Some of the questions we try to answer include what kind of, and how many functions are computed in each space? What kind of runtimes and space-usage do we typically see and how are they arranged over the TM space?

2 Methodology

From now on, we shall write (2,2) for the space of TMs with 2 states and 2 colors, and (3,2) for the space of TMs with 3 states and 2 colors. Let us briefly restate the set-up of our experiment.

2.1 Methodology in short

We look at TMs in (2,2) and compare them to TMs in (3,2). In particular we shall study the functions they compute[5] and the time they take to compute in each space.

[4] We shall often refer to the collection of TMs with k colors and s states as a TM space.

[5] It is not hard to see that any function that is computed in (2,2) is also present in (3,2).

The way we proceeded is as follows. We ran all the TMs in (2,2) and (3,2) for 1000 steps for the first 21 input values $0, 1, \ldots, 20$. If a TM does not halt by 1000 steps we simply say that it diverges. Thus, we collect all the functions on the domain $[0, 20]$ computed in (2,2) and (3,2) and investigate and compare them in terms of run-time, complexity and space-usage.

Clearly, at the outset of this project we needed to decide on at least the following issues:

1. How to represent numbers on a TM?
2. How to decide which function is computed by a particular TM.
3. Decide when a computation is considered finished.

The next subsections will fill out the details of the technical choices made and provide motivations for these choices. Our set-up is reminiscent of and surely motivated by a similar investigation in Stephan Wolfram's book [12], Chapter 12, Section 8.

2.2 Resources

There are $(2sk)^{sk}$ s-state k-color Turing machines. That means 4 096 in (2,2) and 2 985 984 TMs in (3,2). In short, the number of TMs grows exponentially in the amount of resources. Thus, in representing our data and conventions we should be as economical as possible in using our resources so that exhaustive search in the spaces still remains feasible. For example, an additional halting state will immediately increase the search space[6].

2.3 One-sided Turing Machines

In our experiment we have chosen to work with one-sided TMs. That is to say, we work with TMs with a tape that is unlimited to the left but limited to the right-hand side. One sided TMs are a common convention in the literature just perhaps after the more common two sided convention. The following considerations led us to work with one-sided TMs.

- Efficient (that is, non-unary) number representations are place sensitive. That is to say, the interpretation of a digit depends on the position where the digit is in the number. Like in the decimal number 121, the leftmost 1 corresponds to the centenaries, the 2 to the decades and the rightmost 1 to the units. On a one-sided tape which is unlimited to the left, but limited on the right, it is straight-forward how to interpret a tape content that is almost everywhere zero. For example, the tape ...00101 could be interpreted as a binary string giving rise to the decimal number 5. For a two-sided infinite tape one can think of ways to come to a number notation, but all seem rather arbitrary.

[6] Although in this case not exponentially so as halting states define no transitions.

- With a one-sided tape there is no need for an extra halting state. We say that a computation simply halts whenever the head "drops off" the tape from the right hand side. That is, when the head is on the extremal cell on the right hand side and receives the instruction to moves right. A two-way unbounded tape would require an extra halting state which, in the light of considerations in 2.2 is undesirable.

On the basis of these considerations, and the fact that some work has been done before in the lines of this experiment [12] that also contributed to motivate our own investigation, we decided to fix the TM formalism and choose the one-way tape model.

2.4 Unary input representation

Once we had chosen to work with TMs with a one-way infinite tape, the next choice is how to represent the input values of the function. When working with two colors, there are basically two choices to be made: unary or binary. However, there is a very subtle point if the input is represented in binary. If we choose for a binary representation of the input, the class of functions that can be computed is rather unnatural and very limited.

The main reason is as follows. Suppose that a TM on input x performs some computation. Then the TM will perform the very same computation for any input that is the same as x on all the cells that were visited by the computation. That is, the computation will be the same for an infinitude of other inputs thus limiting the class of functions very severely. Thus, it will be unlikely that some universal function can be computed for any natural notion of universality.

On the basis of these considerations we decided to represent the input in unary. Moreover, from a theoretical viewpoint it is desirable to have the empty tape input different from the input zero, thus the final choice for our input representation is to represent the number x by $x + 1$ consecutive 1's.

The way of representing the input in this way has two serious draw-backs:

1. The input is very homogeneous. Thus, it can be the case that TMs that expose otherwise very rich and interesting behavior, do not do so when the input consists of a consecutive block of 1's.
2. The input is lengthy so that runtimes can grow seriously out of hand. See also our remarks on the cleansing process below.

2.5 Binary output convention

None of the considerations for the input conventions applies to the output convention. Thus, it is wise to adhere to an output convention that reflects as much information about the final tape-configuration as possible. Clearly, by interpreting the output as a binary string, from the output value the output tape

configuration can be reconstructed. Hence, our outputs, if interpreted, will be so as binary numbers.

The output representation can be seen as a simple operation between systems, taking one representation to another. The main issue is, how does one keep the structure of a system when represented in another system, such that, moreover, no additional complexity is introduced.

For the tape identity (see Definition 2), for example, one may think of representations that, when translated from one to another system, preserve the simplicity of the function. Some will do so such as taking the output in unary. If one uses a unary representation to feed the *Mathematica* function FindSequenceFunction[7] that will find out, by looking at the sequence of outputs in the chosen representation, that it is about the identity function as one would immediately tell upon looking at the pictogram of the Turing machine. But unary does not work for all other Turing machine evolutions.

For example, when taking the output tape configuration as written in binary, many functions expose (at least) exponential growth. For the tape-identity, that is a TM that outputs the same tape configuration as the input tape configuration, the function $TM(x) = 2^{x+1} - 1$ is the sequence generator under this output representation, rather than $TM(x) = x$. In particular, the TM that halts immediately by running off the tape while leaving the first cell black also computes the function $2^{x+1} - 1$.

These concerns, although legitimate and rich in discussion are undesirable, but as we shall see, in our current set-up there will be few occasions where we actually do interpret the output as a number other than for representational purposes.

2.6 The halting problem and Rice's theorem

By the halting problem and Rice's theorem we know that it is in general undecidable to know wether a function is computed by a particular TM and whether two TMs define the same function. The latter is the problem of extensionality (do two TMs define the same function?) known to be undecidable by Rice's theorem. It can be the case that for TMs of the size considered in this paper, universality is not yet attained[8], that the halting problem is actually decidable in these small spaces and likewise for extensionallity.

[7] This function in *Mathematica* may be seen as a specific purpose Turing machine for which a compiler is needed so that one can provide as input to this function the output of one of our Turing machines. FindSequenceFunction will then attempt to find a simple function that yields the sequence when given successive integer arguments.

[8] Recent work by [15] have shown some small two-way infinite tape universal TMs. It is known that there is no universal machine in the space of two-way unbounded tape (2,2) Turing machines but there is known at least one weak universal Turing machine in (2,3)[12] and it may be (although unlikely) the case that a weak universal Turing machine in (3,2) exists.

As to the halting problem, we simply say that if a function does not halt after 1000 steps, it diverges. Theory tells that the error thus obtained actually drops exponentially with the size of the computation bound [6] and we re-affirmed this in our experiments too as is shown in Figure 2. After proceeding this way, we see that certain functions grow rather fast and very regular up to a certain point where they start to diverge. These obviously needed more than 1000 steps to terminate. We decided to complete these obvious non-genuine divergers manually. This process is referred to as *cleansing*, Of course some checks were performed as to give more grounds for doing so. We are fully aware that errors can have occurred in the cleansing. For example, a progression of a TM is guessed and checked for two values. However, it can be the case that for the third value our guess was wrong: the Halting Problem is undecidable and our approximation is better than doing nothing.

As to the problem of extensionality, we simply state that two TMs calculate the same function when they compute (after cleansing) the same outputs on the first 21 inputs 0 through 20 with a computation bound of 1000 steps. We found some very interesting observations that support this approach: for the (2,2) space the computable functions are completely determined by their behavior on the first 3 input values 0,1,2. For the $(3, 2)$ space the first 8 inputs were found to be sufficient to determine the function entirely.

2.7 Running the experiment

To explore the different spaces of TMs we have programmed in C language a TM simulator. We tested this C language simulator against the `TuringMachine` function in *Mathematica* as it used the same encoding for TMs. It was checked and found in concordance for the whole (2,2) space and a sample of the (3,2) space.

We have run the simulator in the cluster of the CICA (Centro de Informática Científica de Andalucía[9]). To explore the (2,2) space we used only one node of the cluster and it took 25 minutes. The output was a file of 2 MB. For (3,2) we used 25 nodes (50 microprocessors) and took a mean of three hours in each node. All the output files together fill around 900 MB.

3 Results

Definition 1. *In our context and in the rest of this paper, an* algorithm *computing a function is one particular set of 21 quadruples of the form*

$$\langle \textit{input value}, \textit{output value}, \textit{runtime}, \textit{space usage} \rangle$$

where the output, runtime and space-usage correspond to that particular input.

Definition 2. *We say that a TM computes the* tape identity *when the tape configuration at the end of a computation is identical to the tape configuration at the start of the computation.*

[9] Andalusian Centre for Scientific Computing.

3.1 Investigating the space of 2-states, 2-colors Turing machines

In the cleansed data of (2,2) we found 74 functions and a total of 253 different algorithms computing them.

Determinant initial segments An indication of the complexity of the (2,2) space is the number of outputs needed to determine a function. In the case of (2,2) this number of outputs is only 3. For the first output there are 11 different outputs. The following list shows these different outputs (first value in each pair) and the frequency they appear with (second value in each pair). Output `-1` represents the divergent one:

```
{{3, 13}, {2, 12}, {-1, 10}, {0, 10}, {1, 10}, {7, 6}, {6, 4},
 {15, 4}, {4, 2}, {5, 2}, {31, 1}}
```

For two outputs there are 55 different combinations and for three we find the full 74 functions. The first output is most significant; without it, the other outputs only appear in 45 different combinations. This is because there are many functions with different behavior for the first input than for the rest.

We find it interesting that only 3 values of a TM are needed to fully determine its behavior in the full (2,2) space that consists of 4 096 different TMs. Just as a matter of analogy we bring the $\mathbf{C}^\infty$ functions to mind. These infinitely often differentiable continuous functions are fully determined by the outputs on a countable set of input values. It is an interesting question how the minimal number of output values needed to determine a TM grows relative to the total number of $(2 \cdot s \cdot k)^{s \cdot k}$ many different TMs in (s,k) space.

Halting probability In the cumulative version of Figure 2 we see that more than 63% of executions stop after 50 steps, and little growth is obtained after more steps. Considering that there is an amount of TMs that never halt, it is consistent with the theoretical result in [6] that most TMs stop quickly or never halt.

We find it interesting that Figure 2 shows features reminiscent of phase transitions. Completely contrary to what we would have expected, these "phase transitions" were even more pronounced in $(3, 2)$ as one can see in Figure 10.

Runtimes There is a total of 49 different sequences of runtimes in (2,2). This number is 35 when we only consider total functions. Most of the runtimes grow linear with the size of the input. A couple of them grow quadratically and just two grow exponentially. The longest halting runtime occurs in TM numbers 378 and 1351, that run for 8 388 605 steps on the last input, that is on input 20.

Below follows the sequence of {`input, output, runtime, space`} for TM number 378:

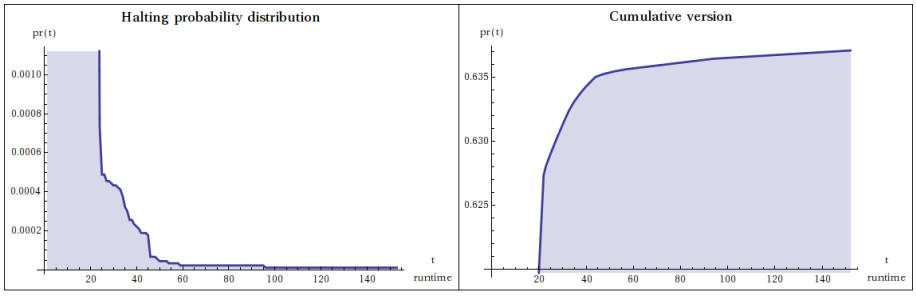


Fig. 2. Halting times in (2,2).

```
{{0, 1, 5, 1}, {1, 3, 13, 2}, {2, 7, 29, 3}, {3, 15, 61, 4},
{4, 31, 125, 5}, {5, 63, 253, 6}, {6, 127, 509, 7}, {7, 255,
1021, 8}, {8, 511, 2045, 9}, {9, 1023, 4093, 10}, {10, 2047,
8189, 11}, {11, 4095, 16381, 12}, {12, 8191, 32765, 13},
{13, 16383, 65533, 14}, {14, 32767, 131069, 15}, {15, 65535,
262141, 16}, {16, 131071, 524285, 17}, {17, 262143, 1048573,
18}, {18, 524287, 2097149, 19}, {19, 1048575, 4194301, 20},
{20, 2097151, 8388605, 21}}
```

Rather than exposing lists of values we shall prefer to graphically present our data. The output values are graphically represented as follows. On the fist line we depict the tape output on input zero (that is, the input consisted of just one black cell). On the second line we depict the tape output on input one (that is, the input consisted of two black cells), etc. By doing so, we see that the function computed by 378 is just the tape identity.

Let us focus on all the (2,2) TMs that compute that tape identity. We will depict most of the important information in one overview diagram. This diagram as shown in figure 3 contains at the top a graphical representation of the function computed as described above.

Below the representation of the function, there are six graphs. On each horizontal axis of these graphs, the input is plotted. The τ_i is a diagram that contains plots for all the runtimes of all the different algorithms computing the function in question. Likewise, σ_i depicts all the space-usages occurring. The $<\tau>$ and $<\sigma>$ refer to the (arithmetical) average of time and space usage. The subscript h indicates that the harmonic average is calculated. As the harmonic average is only defined for non-zero numbers, for technical reasons we depict the harmonic average of $\sigma_i + 2$ rather than for σ_i.

The harmonic mean of the runtimes can be interpreted as follows. Each TM computes the same function. Thus, the total information in the end computed by each TM per entry is the same although runtimes may be different. Hence the runtime of one particular TM on one particular input can be interpreted as time/information. If we consider the following situation:

Let the TMs computing a function be $\{TM_1, \ldots, TM_n$ with runtimes t_1, $\ldots, t_n\}$.

If we let TM_1 run for 1 time unit, next TM_2 for 1 time unit and finally TM_n for 1 time unit, then the amount of information of the output computed is $1/t_1 + ... + 1/t_n$. The corresponding average of this impact function is exactly the harmonic mean, hence the introduction of the harmonic mean as an interpretation of the typical amount of information computed by a random TM in a time unit.

The image provides the basic information of the TM outputs depicted by a diagram with each row the output of each of the 21 inputs, followed by the plot figures of the average resources taken to compute the function, preceded by the time and space plot for each of the algorithm computing the function. For example, this info box tells us that there are 1 055 TMs computing the identity function, and that these TMs are distributed over just 12 different algorithms (i.e. TMs that take different space/time resources). Notice that at first glance at the runtimes τ_i, they seem to follow just an exponential sequence while space grows linearly. However, from the other diagrams we learn that actually most TMs run in constant time and space. Note that all TMs that run out of the tape in the first step without changing the cell value (the 25% of the total space) compute this function.

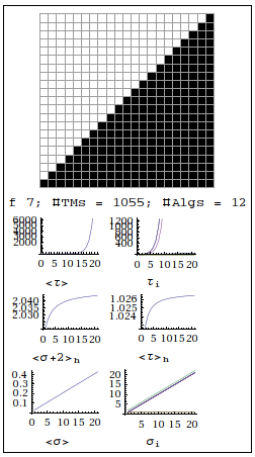


Fig. 3. Overview diagram of the tape identity.

Runtimes and space-usages Observe the two graphics in Figure 4. The left one shows all the runtime sequences in (2,2) and the right one the used-space sequences. Divergences are represented by -1, so they explain the values below the horizontal axis. We find some exponential runtimes but most of them and space-usage remain linear.

An interesting feature of Figure 4 is the clustering. For example, we see that the space usage comes in three different clusters. The clusters are also present in the time graphs. Here the clusters are less prominent as there are more runtimes and the clusters seem to overlap. It is tempting to think of this clustering as rudimentary manifestations of the computational complexity classes.

Another interesting phenomenon is observed in these graphics. It is that of alternating divergence, detected in those cases where value -1 alternates with

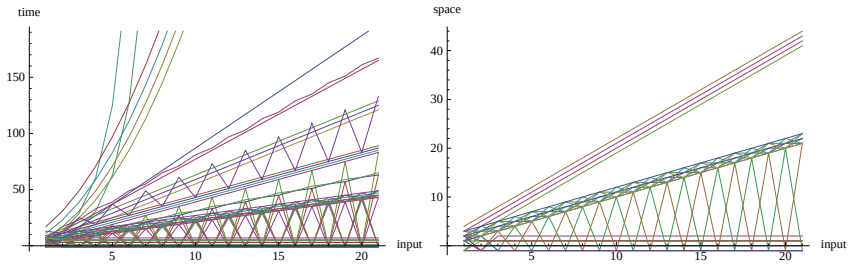


Fig. 4. Runtime and space distribution in (2,2).

the other outputs, spaces or runtimes. The phenomena of alternating divergence is also manifest in the study of definable sets.

Definable sets Like in classical recursion theory, we say that a set W is definable by a (2,2) TM if there is some machine M such that $W = W_M$ where W_M is defined as usual as

$$W_M := \{x | M(x) \downarrow\}.$$

Below follows an enumeration of the definable sets in (2,2).

```
{{}, {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20}, {0}, {0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20},
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20}, {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20}, {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}, {0, 1}}
```

It is easy to see that the definable sets are closed under complements.

Clustering per function We have seen that all runtime sequences in (2,2) come in clusters and likewise for the space usage. It is an interesting observation that this clustering also occurs on the level of single functions. Some examples are reflected in Figure 5.

Computational figures reflecting the number of available resources Certain functions clearly reflect the fact that there are only two available states. This is particularly noticeable from the period of alternating converging and non-converging values and in the offset of the growth of the output, and in the alternation period of black and white cells. Some examples are included in Figure 6.

Computations in (2,2) Let us finish this analysis with some comments about the computations that we can find in (2,2). Most of the TMs perform very simple computations. Apart from the 50% that in every space finish the computations

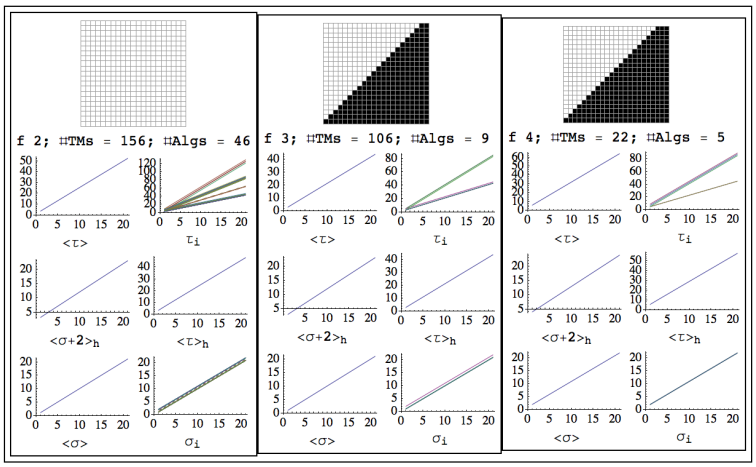


Fig. 5. Clustering of runtimes and space-usage per function.

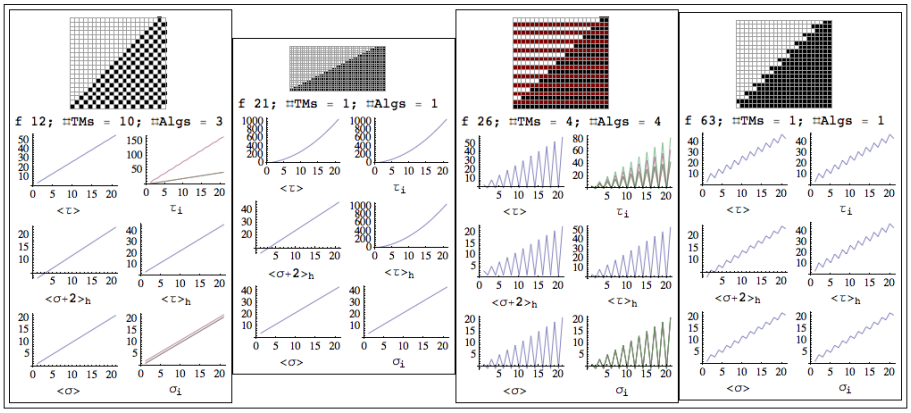


Fig. 6. Computational figures reflecting the number of available resources.

in just one step (those that move to the right from the initial state), the general pattern is to make just one round through the tape and back. It is the case for TM number 2240 with the sequence of runtimes:

```
{5, 5, 9, 9, 13, 13, 17, 17, 21, 21, ..}
```

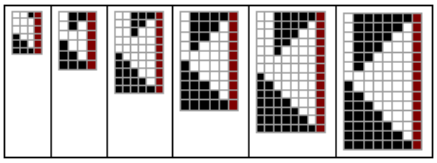


Fig. 7. Turing machine tape evolution for rule 2240.

TM 2205 however is interesting in that it shows a clearly localized and propagating pattern that contains the essential computation. Most TMs that cross the tape just once and then go back to the beginning of the tape expose behavior that is a lot simpler and only visit each cell twice.

Figure 7 shows the sequences of tape configurations for inputs 0 to 5. The walk around the tape can be more complicated. This is the case for TM number 2205 with the runtime sequence:

```
{3, 7, 17, 27, 37, 47, 57, 67, 77, ...}
```

it has a greater runtime but it only uses that part of the tape that was given as input, as we can see in the computations (figure 8, left). In this case the pattern is generated by a genuine recursive process thus explaining the exponential runtime.

The case of TM 1351 is one of the few that escapes from this simple behavior. As we saw, it has the greatest runtimes in (2,2). Figure 8 (right) shows its tape evolution. Note that it is computing the tape identity. Many other TMs in (2,2) compute this function in linear or constant time.

In (2,2) we also witnessed TMs performing iterative computations that gave rise to mainly quadratic runtimes.

As most of the TMs in (2,2) compute their functions in the easiest possible way (just one crossing of the tape), no significant speed-up can be expected. Only slowdown is possible in most cases.

3.2 Investigating the space of 3-state, 2-color Turing machines

In the cleansed data of (3,2) we found 3886 functions and a total of 12824 different algorithms that computed them.

Determinant initial segments As these machines are more complex than those of (2,2), more outputs are needed to characterize a function. From 3 required in (2,2) we need now 8, see Figure 9.

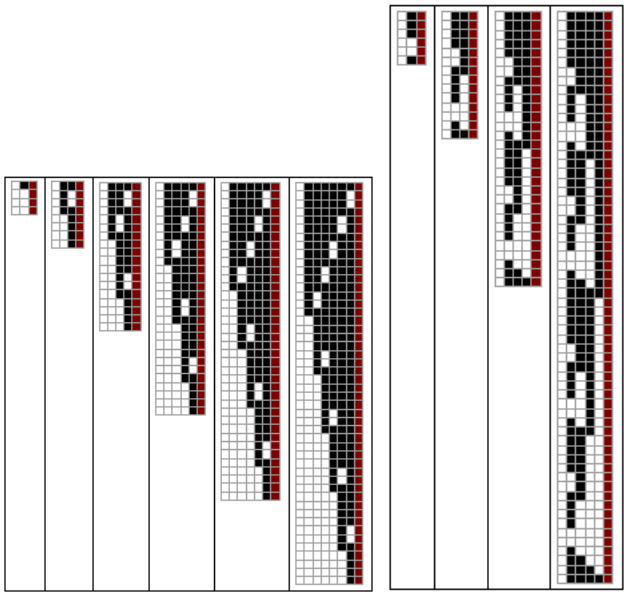


Fig. 8. Tape evolution for rules 2205 (left) and 1351 (right).

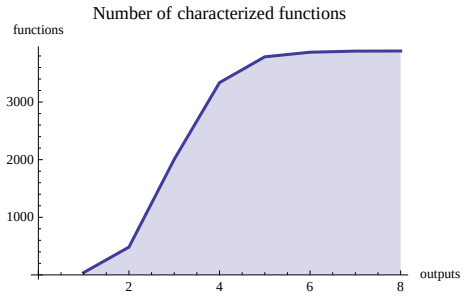


Fig. 9. Number of outputs required to characterize a function in (3,2).

Halting probability Figure 10 shows the runtime probability distributions in (3,2). The same behavior that we commented for (2,2) is also observed. Note that the "phase transitions" in (3,2) are even more pronounced than in (2,2). It is tempting to think as those phase transitions as rudimentary manifestations of computational complexity classes. Further investigation should show whether the distinct regions correspond to particular methods employed by the TMs in that region. Amongst those method we see as most prominent modes of computing the following: running off the tape (almost) immediately; going from the initial head position to the end of the input and back to the beginning again; iterating a particular process several times; recursion.

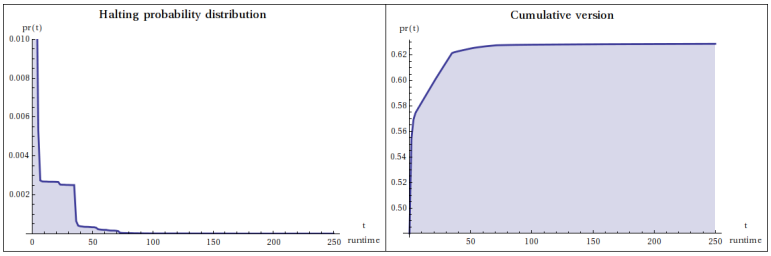


Fig. 10. Runtime proprobability distributions in (3,2).

Runtimes and space-usages In (3,2) the number of different runtimes and space usage sequences is the same: 3676. Plotting them all as we did for (2,2) would not be too informative in this case. So, Figure 11 shows samples of 50 sequences of space and runtime sequences. Divergent values are omitted as to avoid big sweeps in the graphs caused by the alternating divergers. As in (2,2) we observe the same phenomenon of clustering.

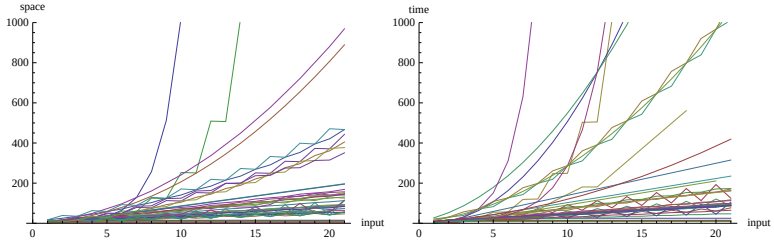


Fig. 11. Sampling of 50 space (left) and runtime (right) sequences in (3,2).

Definable sets Now we have found 100 definable sets. Recall that in (2,2) definable sets were closed under taking complements. It does not happens now. There are 46 definable sets, as

```
{{}, {0}, {1}, {2}, {0, 1}, {0, 2}, {1, 2}, {0, 1, 2}, ...}
```

that coexist with their complements, but another 54, as

```
{{0, 3}, {1, 3}, {1, 4}, {0, 1, 4}, {0, 2, 3}, {0, 2, 4}, ...}
```

are definable sets but their complements are not.

Clustering per function In (3,2) the same phenomenon of the clustering of runtime and space usage in single functions also happens. Moreover, as Figure 12 shows, exponential runtime sequences may occur in a (3,2) function (left) with

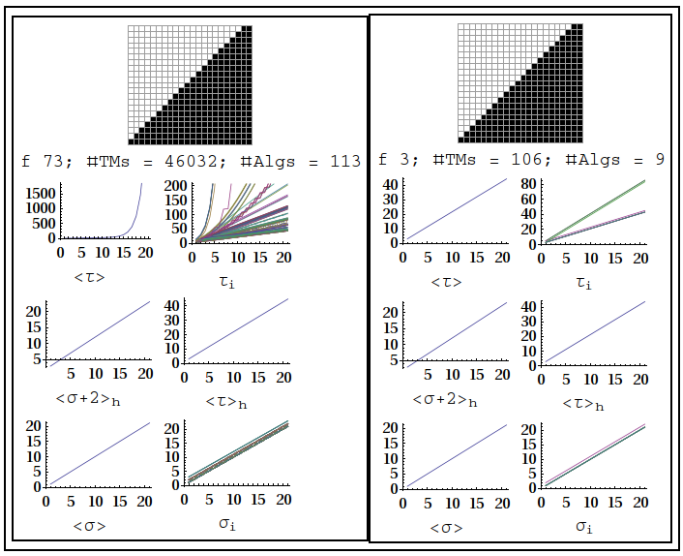


Fig. 12. Clustering per function in (3,2).

other linear behaviors, some of them already present in the (2,2) computations of the function (right).

Exponential behavior in (3,2) computations Recall that in (2,2) most convergent TMs complete their computations in linear time. Now (3,2) presents more interesting exponential behavior, not only in runtime but also in used space.

The max runtime in (3,2) is 894 481 409 steps found in the TMs number 599063 and 666364 (a pair of twin rules[10]) at input 20. The values of this function are double exponential. All of them are a power of 2 minus 2. Look at the first outputs:

```
{14, 254, 16382, 8388606, 137438953470, ... }
```

Adding 2 to each value, the logarithm to base 2 of the output sequence is:

```
{4, 8, 14, 23, 37, 58, 89, 136, 206, 311, 469, 706, 1061,
 1594, 2393, 3592, 5390, 8087, 12133, 18202, 27305}
```

Figure 13 displays these logarithms, and the runtime and space sequences.

Finally, Figure 14 shows the tape evolution with inputs 0 and 1. The pattern observed on the right repeats itself.

[10] We call two rules in (3,2) *twin rules* whenever they are exactly the same after switching the role of State 2 and State 3.

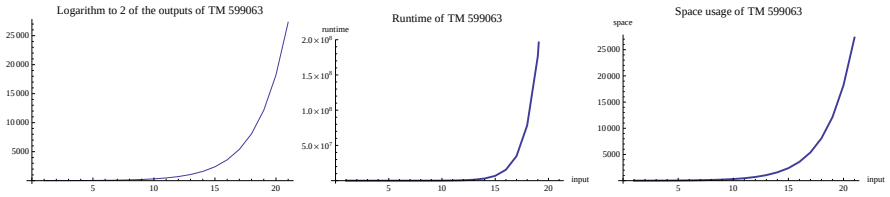


Fig. 13. Rule number 599063. Logarithm to base 2 of the outputs (left), runtime (center) and space usage (right).

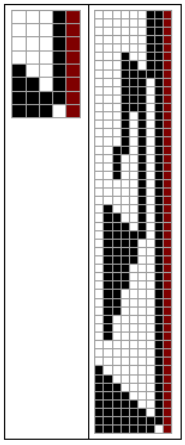


Fig. 14. Tape evolution for rule 599063.

4 Comparison between (2,2) and (3,2)

The most prominent conclusion from this section is that slow-down of a computation is more likely than speed-up.

4.1 Runtimes comparison

In this section we compare the types of runtime progressions we encountered in our experiment. We use the big $\mathcal{O}$ notation to classify the different types of runtimes. Again, it is clear to bear in mind that our findings are based on just 21 different inputs.

As shown no essentially (different asymptotic behavior) faster runtime was found in (3,2), no speed up was found other than by a linear factor as reported in the next section (4.2). That is, no algorithm in (3,2) computing a function in (2,2) was faster than the fastest algorithm computing the same function in (2,2). Obviously (3,2) computes a larger set of functions and they shall be compared to the next larger (4,2) space of TMs. Amusing findings were Turing machines both in (2,2) and (3,2) computing the identify function in as much as exponential time,

as an example of machines spending all resources to compute a simple function. Another example is the constant function $f(n) = 0$ computed in n^9 or n^19, and $f(n) = 1$ computed in as much as exponential time as well, these in (3,2).

In the table, the first column is the function index from 1 to 74 occurred in both (2,2) and (3,2). Under (2,2) is the distribution of time complexity classes for the function in that row in (2,2), followed by the distribution of time complexity classes computing the same function in (3,2). Each time complexity class is followed by the number of occurrences among the algorithms in that TM space and for each function, sorted from greater to lower. No complexity class is estimated if the sequence is divergent, such as for function 1.

Function #	$(\mathbf{2},\mathbf{2})$	$(\mathbf{3},\mathbf{2})$
1	$None$	$None$
2	$O(n), 46$	$O(n), 1084$; $O(1), 129$; $O(n^{19}), 46$ $O(n^3), 8$; $O(n^2), 6$
3	$O(n), 9$	$O(n), 93$; $O(n^2), 12$; $O(Exp), 5$ $O(1), 2$; $O(n^{19}), 1$
4	$O(n), 5$	$O(n), 60$; $O(n^2), 9$; $O(Exp), 4$ $O(n^{19}), 1$
5	$O(n), 2$	$O(n), 133$; $O(n^2), 2$
6	$O(n), 3$	$O(n), 61$; $O(1), 7$; $O(n^3), 1$
7	$O(n), 5$; $O(1), 4$; $O(Exp), 3$	$O(1), 46$; $O(n), 32$; $O(Exp), 17$ $O(n^2), 6$
8	$O(n), 2$	$O(n), 34$
9	$O(n), 1$	$O(n), 34$
10	$O(n), 1$	$O(n), 12$; $O(n^2), 1$
11	$O(n), 2$	$O(n), 25$; $O(n^2), 4$; $O(Exp), 2$
12	$O(n), 3$	$O(n), 70$; $O(n^2), 1$
13	$O(1), 2$	$O(1), 12$
14	$O(1), 5$	$O(1), 23$; $O(n), 8$
15	$O(1), 3$	$O(1), 11$
16	$O(1), 3$	$O(1), 9$
17	$O(n^2), 1$	$O(n^2), 13$
18	$O(n), 1$	$O(n), 12$
19	$O(n), 2$	$O(n), 54$; $O(n^2), 4$
20	$O(n^2), 1$	$O(n^2), 11$
21	$O(n^2), 1$	$O(n^2), 11$
22	$O(n), 1$	$O(n), 14$
23	$O(1), 3$	$O(1), 9$
24	$O(n^2), 1$	$O(n^2), 12$
25	$O(n), 5$	$O(n), 38$; $O(n^9), 2$; $O(n^2), 1$
26	$O(n), 4$	$O(n), 14$
27	$O(1), 1$	$O(1), 6$
28	$O(1), 1$	$O(1), 7$

Function #	$(\mathbf{2},\mathbf{2})$	$(\mathbf{3},\mathbf{2})$
29	$O(1), 39$	$O(1), 107$
30	$O(1), 1$	$O(1), 7$
31	$O(1), 3$	$O(1), 25$
32	$O(1), 1$	$O(1), 5$; $O(n), 1$
33	$O(1), 9$	$O(1), 9$; $O(n), 7$; $O(Exp), 3$
34	$O(1), 23$	$O(1), 58$; $O(n), 13$; $O(Exp), 1$
35	$O(n), 2$	$O(n), 31$; $O(n^2), 2$
36	$O(n), 1$	$O(n), 19$; $O(1), 3$
37	$O(n), 1$	$O(n), 12$
38	$O(1), 1$	$O(1), 23$; $O(n), 1$
39	$O(1), 1$	$O(1), 16$
40	$O(n), 1$	$O(n), 6$; $O(1), 3$
41	$O(1), 1$	$O(1), 23$
42	$O(1), 4$	$O(1), 42$; $O(n), 1$
43	$O(1), 2$	$O(1), 16$
44	$O(1), 1$	$O(1), 22$; $O(n), 1$
45	$O(1), 1$	$O(1), 8$
46	$O(1), 1$	$O(1), 14$; $O(n), 2$
47	$O(n), 1$	$O(n), 57$; $O(1), 26$
48	$O(n), 1$	$O(n), 32$
49	$O(n), 1$	$O(n), 17$; $O(1), 14$
50	$O(n), 1$	$O(n), 15$
51	$O(n), 1$	$O(n), 15$
52	$O(n), 1$	$O(n), 12$
53	$O(1), 1$	$O(1), 10$
54	$O(1), 3$	$O(1), 70$
55	$O(1), 3$	$O(1), 17$; $O(n), 1$
56	$O(1), 6$	$O(1), 35$; $O(n), 7$
57	$O(1), 1$	$O(1), 21$; $O(n), 4$; $O(Exp), 2$
58	$O(1), 1$	$O(1), 22$
59	$O(1), 1$	$O(1), 15$; $O(n), 7$
60	$O(n), 1$	$O(n), 37$; $O(1), 1$
61	$O(n), 1$	$O(n), 45$; $O(1), 3$
62	$O(n), 1$	$O(n), 20$; $O(1), 15$
63	$O(n), 1$	$O(n), 11$
64	$O(n), 1$	$O(n), 31$; $O(1), 2$
65	$O(n), 1$	$O(n), 21$
66	$O(1), 1$	$O(1), 20$
67	$O(1), 1$	$O(1), 25$
68	$O(1), 1$	$O(1), 11$
69	$O(1), 2$	$O(1), 16$
70	$O(n), 1$	$O(n), 4$; $O(1), 3$
71	$O(n), 1$	$O(n), 20$; $O(1), 1$
72	$O(1), 1$	$O(1), 4$
73	$O(n), 1$	$O(n), 10$
74	$O(n), 1$	$O(n), 12$; $O(1), 2$

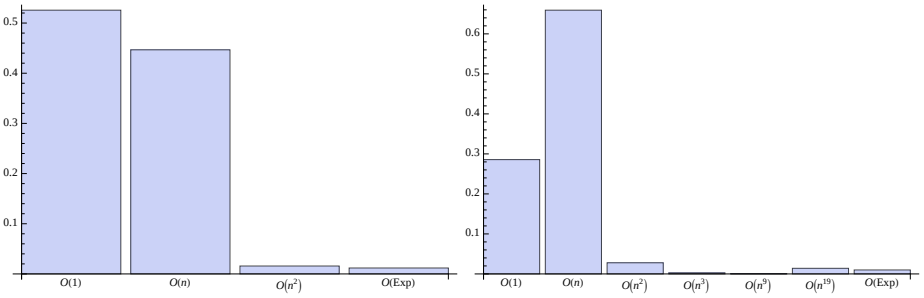


Fig. 15. Time complexity distributions of (2,2) (left) and (3,2) (right).

As shown in this time complexity table comparing runtimes between (2,2) and (3,2), no speed up was found other than by a linear factor as reported in the next subsection (4.2). That is, no algorithm in (3,2) computing a function in (2,2) was faster than the fastest algorithm computing the same function in (2,2). Obviously (3,2) computes a larger set of functions and they shall be compared to the next larger (4,2) space of TMs. An amusing finding were Turing machines both in (2,2) and (3,2) computing the identify function in as much as exponential time, as an example of a machine spending all resources to compute a simple function.

4.2 Quantifying the linear speed-up factor

For obvious reasons all functions computed in (2,2) are computed in (3,2). The most salient feature in the comparison of the (2,2) and (3,2) spaces is the prominent slowdown indicated by both the arithmetic and the harmonic averages. (3,2) spans a larger number of runtime classes. Figures 16 and 17 are examples of two functions computed in both spaces in a side by side comparison with the information of the function computed in (3,2) on the left side and the function computed by (2,2) on the right side. Notice that the numbering scheme of the functions indicated by the letter f followed by a number may not be the same because they occur in different order in each of the (2,2) and (3,2) spaces but they are presented side by side for comparison with the corresponding function number in each space.

One important calculation experimentally relating descriptional (program-size) complexity and (time resources) computational complexity is the comparison of maximum of the averages on inputs 0,...,20, and the estimation of the speed-ups and slowdowns factors found in (3,2) with respect to (2,2).

It turns out that 19 functions out of the 74 computed in (2,2) and (3,2) had at least one fastest computing algorithm in (3,2). That is 0.256 of the 74 functions in (2,2). A further inspection reveals that among the 3 414 algorithms in (3,2), computing one of the functions in (2,2), only 122 were faster. If we supposed that "chances" of speed-up versus slow-down on the level of algorithms were fifty-fifty, then the probability that we observed at most 122 instantiations of

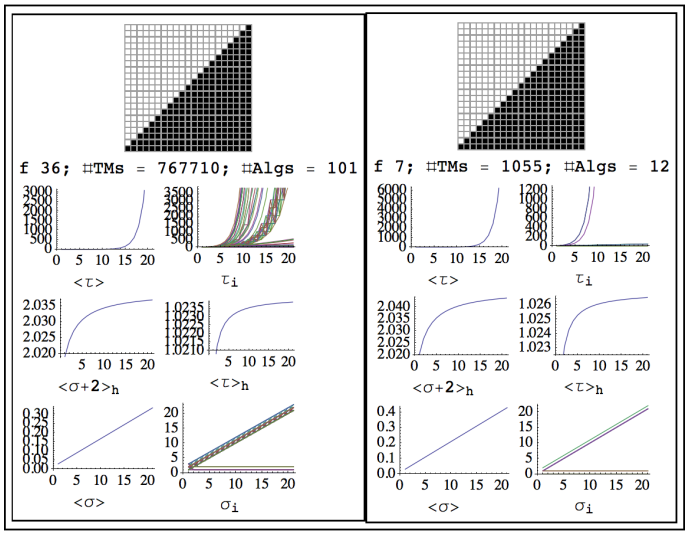


Fig. 16. Side by side comparison of an example computation of a function in (2,2) and (3,2) (the identity function).

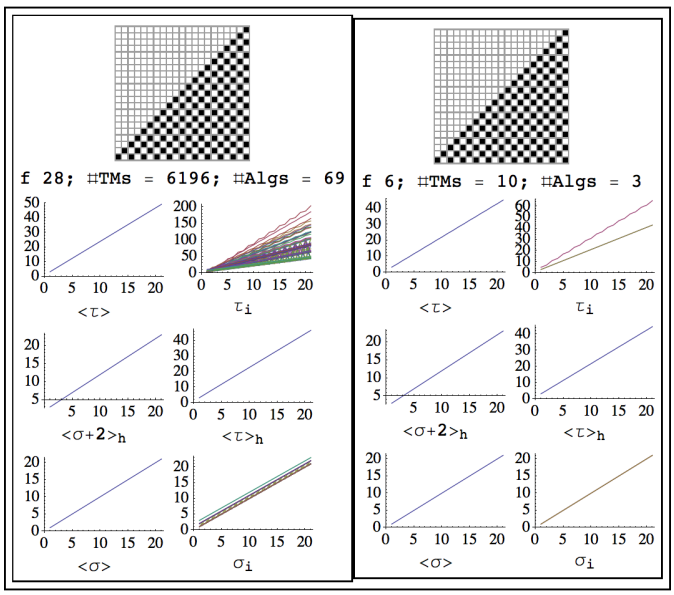


Fig. 17. Side by side comparison of the computation of a function in (2,2) and (3,2).

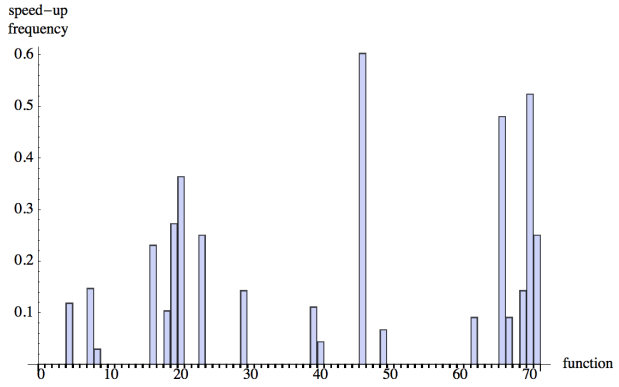


Fig. 18. Distribution of speed-up probabilities per function. Interpreted as the probability of picking a an algorithm in (3,2) computing faster an function in (2,2).

speed-up would be in the order of 10^{-108}. Thus we can safely state that the phenomena of slow-down at the level of algorithms is significant.

Figure 18 shows the scarceness of the speed-up and the magnitudes of such probabilities. Figures 19 quantify the linear factors of speed-up showing the average and maximum. The typical average speed-up was 1.23 times faster for an algorithm found when there was a faster algorithm in (3,2) computing a function in (2,2).

In contrast, slowdown was generalized, with no speed-up for 0.743 of the functions. Slowdown was not only the rule but the significance of the slowdown much larger than the scarce speed-up phenomenon. The average algorithm in (3,2) took 2 379.75 longer and the maximum slowdown was of the order of 1.19837×10^6 times slower than the slowest algorithm computing the same function in (2,2).

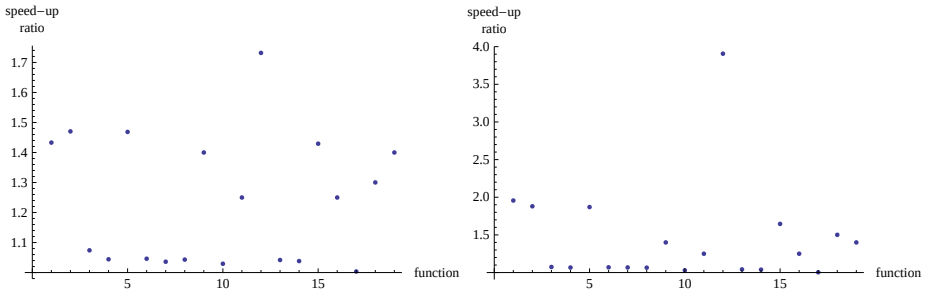


Fig. 19. Speed up significance: on the left average and on the right maximum speed-ups.

5 Concluding

We have undertaken a systematic and exhaustive study of small Turing machine with 2 colors and 2 and 3 states. For larger number of states, sampling was unavoidable and results are yet to be interpreted. The *Halting Problem* and other undecidable concerns for an experimental procedure such as the presented herein, including the problem of extensionality, were overcome by taking a finite and pragmatic approach (theory tells us that in various cases the corresponding error drops exponentially with the size of the approximation[6]). Analyzing the data gave us interesting functions with their geometrical patterns for which average and best case computations in terms of time steps were compared against descriptional complexity (the size of the machines in number of states).

Because picking an algorithm at random with uniform probability among the algorithms computing a function in an increasingly larger Turing machine space according to the maximum allowed number of states leads to increasingly greater chances to pick a slow algorithm compared to the number of fastest algorithms in the same space, one may say that an additional effort has to be made, or additional knowledge has to be known, in order to pick a faster algorithm without having to spend larger and larger resources in the search of an efficient algorithm itself. One can say that this is a *No free lunch*-type metaphor saying that speeding-up *is not for free*.

Exact evaluations with regard to runtimes and space-usages were provided shedding light onto the micro-cosmos of small Turing machines, providing figures of the halting times, the functions computed in (2,2) and (3,2) and the density of converging versus diverging computations. We found that increasing the descriptional complexity (viz. the number of *states*), the number of algorithms computing less *efficiently*, relative to the previous found runtimes in (2,2), computing a function grows faster than the number of machines more *efficiently* computing it. In other words, given a function, the set of average runtimes in (2,2) *slows down* in (3,2) with high probability.

Acknowledgements

The initial motivation and first approximation of this project was developed during the NKS Summer School 2009 held at the Istituto di Scienza e Tecnologie dell'Informazione, CNR in Pisa, Italy. We wish to thank Stephen Wolfram for interesting suggestions and guiding questions. Furthermore, we wish to thank the CICA center and its staff for providing access to their supercomputing resources. Hector Zenil also wants to thank the CONACyT for providing financial support, as well as Wolfram Research.

References

1. C.H. Bennett. Logical Depth and Physical Complexity in Rolf Herken (ed) *The Universal Turing Machine–a Half-Century Survey,* Oxford University Press 227-257, 1988.

2. C.H. Bennett. How to define complexity in physics and why. In *Complexity, entropy and the physics of information.* Zurek, W. H.; Addison-Wesley, Eds.; SFI studies in the sciences of complexity, p 137-148, 1990.
3. M. Cook. *Universality in Elementary Cellular Automata.* Complex Systems, 2004.
4. S. Cook. *The complexity of theorem proving procedures.* Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151-158, 1971.
5. G.J. Chaitin. *Gödel's theorem and information,* Int. J. Theoret. Phys. 21, pp. 941-954, 1982.
6. C.S. Calude, M.A. Stay, *Most programs stop quickly or never halt,* Advances in Applied Mathematics, 40 295-308, 2005.
7. E. Fredkin. *Digital Mechanics,* Physica D: 254-70, 1990.
8. A. N. Kolmogorov. *Three approaches to the quantitative definition of information.* Problems of Information and Transmission, 1(1):1–7, 1965.
9. L. Levin. *Universal search problems.* Problems of Information Transmission 9 (3): 265-266. 1973
10. S. Lin & T. Rado. *Computer Studies of Turing Machine Problems.* J. ACM. 12, 196-212, 1965.
11. S. Lloyd, *Programming the Universe,* Random House, 2006.
12. S. Wolfram, *A New Kind of Science.*, Wolfram Media, 2002.
13. *Wolfram's 2, 3 Turing Machine Research Prize.*, http://www.wolframscience.com/prizes/tm23/ Accessed on June, 24, 2010.
14. R. Neary and D. Woods. *On the time complexity of 2-tag systems and small universal turing machines.* In FOCS, pages 439-448. IEEE Computer Society, 2006.
15. D. Woods & T. Neary. *Small semi-weakly universal Turing machines.* Fundamenta Informaticae. 91:161-177 (2009).

Through the Looking Glass: What Computation Found There

Rossella Lupacchini

Dipartimento di Filosofia, Università di Bologna (Italy)
rossella.lupacchini@unibo.it
OUTLINE

Abstract. Commenting about the fact that *two* matrices, $\mathbf{1}$ and $-\mathbf{1}$, within the 2-dimensional *complex* space, can be mapped onto the *identity* matrix $\mathbf{1}$, in the 3-dimensional real space, Goldstein (1950) remarks that this apparently strange fact has no "physical" meaning. The [complex] space is a purely mathematical construction, conceived to establish a correspondence between 2x2 and 3x3 matrices of a particular kind. Such a space can not possess the same properties of the 3-dimensional physical space. This paper is an attempt to endow the *complex space* with physical and computational meaning.

1 Observability in Computation and Physics

The following quotation from Hermann Weyl's 1949 book reveals both the intimate link between foundational issues in physics and mathematics as well as their common root in Hilbert's work.

> The "physical process" undisturbed by observation is represented by a mathematical formalism without intuitive [*anschauliche*] interpretation; only the concrete experiment, the measurement by means of a grating, can be described in intuitive terms. This contrast of physical process and measurement has its analogue in the contrast of formalism and meaningful thinking in Hilbert's system of mathematics. (Weyl 1949, p. 261)

Hilbert's finitist proof theory was intended to solve this contrast by "projecting" continuous mathematics into *discreteness*. It failed. Gödel's incompleteness theorems and Turing's negative solution of the *Entscheidungsproblem* took Hilbert's proof theory to its limits as they showed that mathematical procedures cannot be completely included in one "formal system". Those limits, set by a *Turing machine*, overlap with classical physics'.

However, at first in the 1930s, Gödel considered his incompleteness results not as a failure Hilbert's program, but rather as a claim "that through the transition from evidence to formalism something is lost." In his view, "questions which are undecidable in a given formalism are always decidable by evident inferences not expressible in the given formalism." (Gödel 193?, p. 164) Therefore, his point was that it is not possible to include mathematics in one formal system, but Hilbert's

conviction remained entirely untouched. But what is lost through the transition from evidence to formalism? Is it possible to fill the gap between mathematical evidence and logical formalism?

In Königsberg, in the fall of 1930, at almost the same time that Gödel was sketching his incompleness results, at the Congress of Scientific Epistemology, Hilbert addressed such questions in a lecture on *Naturerkennen und Logik*, delivered at the meeting of the Society of German Scientists and Physicists. The issue, which came into focus with Kants transcendental philosophy, was the part played in our understanding by logic on the one side and experience on the other. Hilbert's answer emerges from mathematics and the axiomatic method. Kant's a priori theory contains anthropomorphic dross from which it must be freed: "After we remove that, only that a priori will remain which also is the foundation of pure mathematical knowledge". Nevertheless, Hilbert believes "that, in the end, mathematical knowledge rests on a kind of *intuitive insight* [*anschaulicher Einsicht*], and that even for building up the theory of numbers a certain a priori intuitive view is necessary. With this, the most general, basic idea of Kantian epistemology retains its significance, namely, the philosophical problem of characterizing that intuitive view and thus investigating the conditions of possibility of all conceptual knowledge and, at the same time, of every experience." In his investigations of the foundations of mathematics, Hilbert tackled this problem on the way of finitism, *i.e.* of arithmetic. And yet, on reflection, Hilbert's vision of geometry could provide a more comprehensive general frame:

> At the time of Kant, one could well think that *geometry* was, like arithmetic, something which precedes all natural knowledge. This Kantian view was abandoned, since geometry is nothing but that part of the whole conceptual framework of physics which represents the possible position relations among rigid bodies in the world of real objects. (Hilbert 1923)

While Gödel showed that it is impossible to carry a finitist analysis to such a point that all the intuitive judgements of mathematics could be replaced by a finite number of mechanical rules, Turing made clear what a mechanical rule is, by imposing precise *finiteness conditions* on mathematical procedures. Thus, he made clear how to distinguish when a step is purely formal and when a step makes use of intuition. By grounding those conditions on physical processes, the very necessity for intuition may then be characterized as emerging out of some physical constrains.[1]

Computability limits set by Turing (1936) are motivated as boundedness conditions on the configurations of symbols which are operated on by a "computer". All such configurations must be "immediately recognisable" *by* the computer. As he understood calculations as symbolic processes carried out by a computer, Turing was able to impose restrictions on the operations permitted and justify them through an analysis of the idealized capacities of the computer available

[1] This line of research has been envisaged, in different ways, by Turing (1948), Gandy (1980) and von Neumann (1954).

for their execution. Therefore, he was able to conceive a *machine* playing the role of the computer: its lay-out was modelled in an abstract symbolic structure, *i.e.*, a Turing machine. In arguing for the adequacy of his notion, Turing focused on the essential (human) capacity involved in computing, namely *distinguishing* symbols, and formulated it in terms of finiteness conditions on the symbols scanned by the machine. In accordance with experience, "if we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent." (Turing 1936, p. 75)

By stretching the ideal of formalism and finitism - *i.e.*, "to atomize mathematical reasoning into such tiny steps that nothing is left to the imagination, nothing is left out!" (Chaitin 2002) - a Turing machine, on one hand, guarantees to mathematics its "existence" *via* undecidability (von Neumann 1927); on the other, it demands for indeterminism to be capable of "meaningful thinking". By connecting the *effectiveness* of computability to the "resolution power" of the *computer* involved, Turing's computability shows that, beside any "concrete" physical process, any "effective" process of computation rests on observation. No adequate understanding of effective procedures can dispense with the medium of the agent (computer, observer or measurer) working out the operations involved. Computational and physical theories are bound to observability constraints. But quantum theory demands more, it demands to refine the very notion of "observability".

2 Quantum Observability

Any physical theory is about observables, namely physical quantities which can be measured on a system, but the classical presupposition that the measured values correspond to objective properties of the system - "*be*ables" - is not tenable in quantum theory because its observables can be *incompatible*. A quantum state does not describe how things are but how their probabilities are weaved. Quantum physics differs from classical physics as to the impossibility of performing certain measurements simultaneously with accuracy: a measurement is not datum "copiative contemplation", it is an *inter-action* between the system-to-be-observed and the observer-system; hence, it establishes a connection between the two parts. As far as measurement is viewed as a subject-object interaction, with the twin requirement of freedom in choosing the observable to be questioned and capability of distinguishing "incompatible" outcomes, quantum theory demands to sharpen the probability relations associated with its possible states and, consequently, to refine their mathematical representation. According to the Heisenberg principle, the uncertainty in the value of one observable has to be rigorously distinct from, but not independent of, the uncertainty in the values of the others incompatible observables. Thus "incompatible" does not mean "not able to coexist", quite the contrary. Incompatible observables live *within the same* representation space and are represented by operators which are *mutually transformable* and *do not commute*.

Physical quantities, which have no classical analogue, are the intrinsic spins of quantum particles. Consider a triad S_α S_β, S_γ representing the spin components of an electron. Each of these observables is assumed to have two values, '+' and '-'. Any "pure" state of the electron assigns probability 1 to exactly one value of one observable, say $(S_\alpha, +)$, and probability 0 to the opposite value $(S_\alpha, -)$, and the same probability 0.5 to the values of the incompatible observables $(S_\beta, +)$ and $(S_\beta, -)$, $(S_\gamma, +)$ and $(S_\gamma, -)$. Accordingly, any pure state of one observable is equidistant from the pure states of the other observables.

A unit sphere is a convenient way to visualize the *symmetry and continuity* constraints on probabilities associated with such incompatible observables, keeping in mind that the angular separation between "orthogonal" pure states of the same observable *doubles* $\frac{\pi}{2}$. If a pure state ψ of one observable is represented by the point $\sigma = (\phi, \theta)$ on the sphere,[2] the second pure state of the same observable, orthogonal to the first, is represented by the antipode $\sigma^* = (\pi \pm \phi, \theta)$. As an "observer-subject", the state ψ assigns probabilities to each *experimental question* concerning the value of each observable S_σ over the sphere, i.e. concerning the "object" $\sigma^+ \equiv (S_\sigma, +)$. The probability is a symmetrical and continuos function f of the angular separation δ between any pair of states corresponding to the points σ and ϑ on the sphere: $p_\sigma(\vartheta) = f(\delta_{\sigma,\vartheta}) = p_\vartheta(\sigma)$; hence ψ assigns probability 1 to exactly one point, that that coincides with its own "point of view", namely when $\delta = 0$, and the same probability $p_\psi(\sigma)$ to all points on the same 'latitude' as σ.

The point at issue is that no pure state of one observable can coincide with a pure state of another observable, for all pure states must be *distinguishable*. Here is the reason to require, beside orthogonality between pure states of one and the same observable, that the operators representing incompatible observables do not commute. However, by *rotating* the sphere, the diagram of S_σ–results can be transformed into the diagram of S_ϑ-results as the operators are mutually transformable. Thus pure states of the same observable are *invariantly* mutually orthogonal, while pure states of incompatible observables are mutually "oblique". That is how probabilities are assigned to quantum states over a unit sphere according to the uncertainty principle.

Now we must distinguish between the visual three dimensional space containing the points σ and the *representation* space from which the usual algorithm generates probability assignments (Hughes 1989). Since the possible outcomes of each measurement are two, the representation space has to be two-dimensional. How to render in two dimensions the network of probability relations amongst the values of three incompatible observables? The symmetry group of unit sphere, which is the set of all its rotations about its centre, has no representation in the two-dimensional real space. A subtle invention is needed.

[2] The azimuthal angle ϕ can vary as $-\pi < \phi \leq \pi$ and the longitude θ as $-\frac{\pi}{2} < \theta \leq \frac{\pi}{2}$.

3 Alberti's Veil *vs* Einstein's

Reflecting about the significance of quantum theory, John Bell (1973) underlined that quantum theory is fundamentally about the results of "measurements", and therefore presupposes a "measurer" (or subject) in addition to the "system" (or object). But a theory about "measurement" implies incompleteness of the system and unanalyzed interventions from outside. Here is why the *subject-object distinction* is viewed as an issue "at the very root of the unease that many people still feel in connection with quantum mechanics." Bell raised the question as to how it can again become possible "to say of a system not that such and such may be *observed* to be so but that such and such *be* so". Can "Einstein's veil" be removed?

As mentioned above, the probabilities distribution over the unit sphere is a uniform map of points whose angular separation is *twice* the angular separation between the corresponding quantum states. This doubling of angles recalls Hamilton's mistake in his attempt to give a meaning to *imaginary units* through rotations. The fascinating story of the invention of "quaternions" is masterfully told by Altmann (1992). Here it is worth recalling that Hamilton's "original sin" lies in interpreting a *pure normalized* quaternion $Q = \left[\cos\frac{\pi}{2}, \sin\frac{\pi}{2}\mathbf{r}\right] = [0, \mathbf{r}]$ as a vector $\mathbf{r}$. But Q is not a vector, it is a rotation by π about the axis $\mathbf{r}$, namely a *binary rotation*. In two dimensions, such a rotation requires a *reflection* operator.

Almost four centuries before the invention of quaternions, we can recognize a reflection operator in the "Alberti's veil", the most eloquent icon of the invention of *perspectiva pingendi*. It is wellknown that carrying over concepts and methods of the medieval *natural* perspective into a flat surface, the Renaissance artists bring about the *artificial* perspective. This *inventio* asks the light to get rid of any "substantial" character and its rays to challenge the rules of Euclidean geometry: traveling in parallel they meet in one point and give rise to a *pictorial* space. The result is a painted or drawn scene, which is supposed to be indistinguishable from the image transmitted by a glass or reflected by a mirror. It is achieved by projecting the three-dimensional vision on a plane, letting the flight lines converge in a *central* point specularly symmetrical to the unmoving eye of the painter-observer. In this representation space, every image is anchored to its author-creator through Alberti's veil acting as a "beam-splitter".

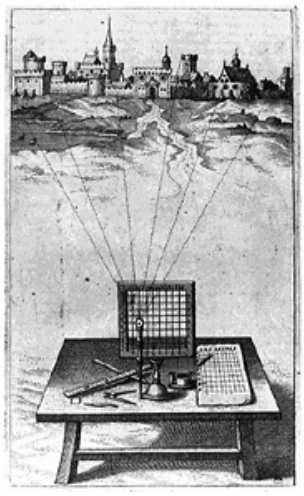


Fig. 1. Alberti's veil

A superb illustration of the epistemological value of artificial perspective is provided by the subject-object specular symmetry emerging from the Arnolfinis portrait by van Eyck. The clear signature of the artist is on the wall behind the Arnolfinis: "Johannes de Eyck fuit hic 1434". However, that "hic" set the

painter not only in the *historical* space-time of the portrait, as a *faber*, but also in the *symbolic* representation space of the painting, together with the Arnolfinis. Approximately at the point where, according to the correct rules of perspective, the flight lines would converge the outline of van Eyck is reflected in a mirror.

In this painting, the auto-portrait of the painter and the ancillary role of writing (Johannes de Eyck fuit hic) make evident what, some decades later, the artificial perspective would set up as a "formal system": the displaying of the representation space through an *imaginary* dimension traced to the painter's eye. To allow the picture to take shape a third dimension must be added: then the Arnolfinis' scene is unfolded to the rear, while the painter is projected to the front by its mirror image. The painter, as well as the observer, can benefit from two "complementary" points of view: one "real" - within the sensitive reality of the person who watches the painting and sees the frontal scene; the other "reflected" - within the reality constructed by art, beyond the plane of the representation, where the other side of the scene is imagined. The *perspectival* representation space enables the painter to be inside and outside the representation, alternatively observer-subject and observed-object, because the two conditions - observing and being observed - are symmetrical, mutually transformable, thanks to the overturning in the painting.

Coming back to Hamilton's concern about imaginary units, the first step is to understand the meaning of the multiplication rule

$$\mathbf{i}^2 = -1$$

at the origins of complex algebra. On the Argand plane, one can easily see that $\mathbf{i}$ rotates all objects of the form $a + b\mathbf{i}$ by $\frac{\pi}{2}$.[3] By repeating this operation, α is rotated by π and changes sign.[4] Accordingly, $\mathbf{i}^2 = -1$ is understood as a binary rotation.

Quaternions arise by continuing the "doubling" process that gives us complex numbers from real numbers. As an extension of complex numbers, Hamilton sees them related to rotations. How? Quaternions are objects of the form $a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, where a, b, c, d are real, and $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are multiplied according to the rules

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \qquad \mathbf{ij} = -\mathbf{ji} = \mathbf{k}.$$

A quaternion $A = [a, \mathbf{A}]$, where $\mathbf{A} = b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, will be *normalized* when

[3] $\alpha = a + b\mathbf{i} \Rightarrow i\alpha = -b + a\mathbf{i} = \alpha_{\perp}$.
[4] $\mathbf{i}\alpha = \alpha_{\perp} \Rightarrow \mathbf{i}\alpha_{\perp} = -(a + b\mathbf{i}) = -\alpha$.

$$|A|^2 = a^2 + |\mathbf{A}|^2 = 1.$$

So, all quaternions of the form $[\cos\alpha, \sin\alpha\mathbf{n}]$ with $|\mathbf{n}|^2 = 1$ are normalized; if $\alpha = \frac{\pi}{2}$, then such quaternions are also *pure*. What is the meaning of a pure normalized quaternion? What is the action of a normalized quaternion on a pure normalized quaternion?

Following Hamilton, since quaternions are related to rotations and a pure normalized quaternion is to identify with a unit vector, a normalized quaternion acting on a unit vector rotates the vector, *i.e.* it produces another pure normalized quaternion. Thus, if $\rho = [0, \mathbf{r}]$ with $|\mathbf{r}| = 1$,

$$A\rho = [\cos\alpha, \sin\alpha\mathbf{n}]\,[0, \mathbf{r}] = [0 - \sin\alpha\mathbf{n}\cdot\mathbf{r}, \cos\alpha\mathbf{r} + \sin\alpha\,(\mathbf{n}\times\mathbf{r})].$$

If $\mathbf{n}$ and $\mathbf{r}$ are orthogonal, the scalar product $\mathbf{n}\cdot\mathbf{r}$ will be null, and the outcome is another pure normalized quaternion:

$$A\rho = [0, \cos\alpha\mathbf{r} + \sin\alpha\,(\mathbf{n}\times\mathbf{r})] = [0, \mathbf{r}'] = \rho'\ .$$

As to the geometrical interpretation, for Hamilton the action of the quaternion A is the rotation $R\,(\alpha\mathbf{n})$, by the angle α about the axis $\mathbf{n}$,

$$A = [\cos\alpha, \sin\alpha\mathbf{n}] \Rightarrow R\,(\alpha\mathbf{n}),$$

which transforms the vector $\mathbf{r}$ into $\mathbf{r}'$ (see Fig. 2A). However, the correct meaning of $A\rho = \rho'$ is given by Fig. 2B: the result of a binary rotation about $\mathbf{r}$ followed by a rotation by 2α about the axis $\mathbf{n}$, with $\mathbf{r}\perp\mathbf{n}$, is a binary rotation about $\mathbf{r}'$, with $\mathbf{r}'\perp\mathbf{n}$, at an angle α from $\mathbf{r}$. (Simon 1992, p. 58)

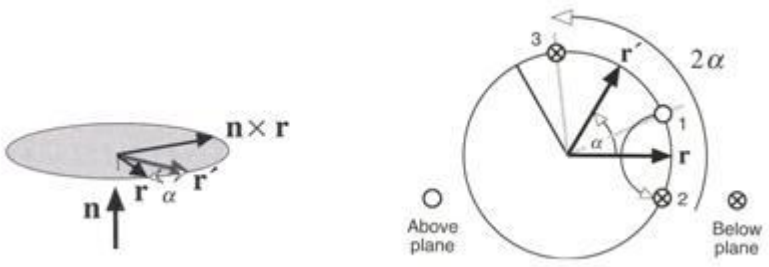


Fig. 2. A: Hamilton's rotation B: Binary rotation about $\mathbf{r}'$

Therefore, if the quaternion A is interpreted as a rotation $R\,(2\alpha\mathbf{n})$, a pure normalized quaternion, such as ρ, cannot be identified with a vector $\mathbf{r}$, but rather with a binary rotation $R\,(\pi\mathbf{r})$:

$$A = [\cos\alpha, \sin\alpha\mathbf{n}] \Rightarrow R(2\alpha\mathbf{n})$$

$$\rho = [0, \mathbf{r}] = \left[\cos\tfrac{\pi}{2}, \sin\tfrac{\pi}{2}\mathbf{r}\right] \Rightarrow R(\pi\mathbf{n})$$

It follows that, if an imaginary unit, like a pure normalized quaternion, is a rotation by π,

$$i = [0, \mathbf{i}] \Rightarrow R(\pi\mathbf{i})$$

its square must be a rotation by 2π:

$$R(2\pi\mathbf{i}) \Rightarrow [\cos\pi, \sin\pi\mathbf{i}] = -1!$$

So a rotation by 2π is not the identity, it changes the sign of its operand.

4 Rotations, quaternions, and mirrors

According to Hermann Weyl (1952), the rotation symmetries of the space can be condensed in the so-called four-group consisting of the identity and the binary rotation [*Umklappung*] around three mutually perpendicular axes. The requirement for binary rotations in $\mathbb{R}^3$ leads to introduce the reflection $\mathbf{R}^*$ in the origin which carries any point σ into its antipode σ^*. By including rotations of the form $\mathbf{R}^*\mathbf{R}$ in the four-group $\mathbf{4G}$ one obtains the group $\mathbf{4G}^* = \mathbf{4G} + \mathbf{R}^*\mathbf{4G}$ which doubles $\mathbf{4G}$. Notice that Weyl praises Leonardo da Vinci for making up "a *complete* list of *orthogonally inequivalent* finite groups of orthogonal transformations."
[...]

The Cayley-Klein parameters provide the keys to transfer the group of rotations of the real space $\mathbb{R}^3$ into the complex space $\mathbb{C}^2$: to any rotation which leaves invariant the angular separation between points of the unit sphere, there corresponds *two* unitary operators $\mathbf{U}$ and $-\mathbf{U}$ on the set of rays of $\mathbb{C}^2$, which leave invariant the angular separation between rays. One peculiar feature of matrices involving Cayley-Klein parameters is the presence of half-angles. As it happened, Hamilton's quaternions would turn into "Pauli spin matrices". [...]

In $\mathbb{C}^2$, the general form of a matrix representing an observable S_σ can be written as $\mathbf{S}_\sigma = x\sigma_x + y\sigma_y + z\sigma_z$,[5] where σ_x, σ_y, σ_z are the Pauli spin matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

They are orthogonal "*mirrors*" perpendicular to $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ respectively.
[...]

The presence of half-angles in the 2x2 unitary matrices corresponding to 3x3 rotation matrices entails some strange properties of the complex representation

[5] $\mathbf{S}_\sigma = \begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix}$

space. Whereas, in the Euclidean space, a rotation by $\theta = 2\pi$ about z results in the identity transformation $\mathbf{R}_{2\pi} = \mathbf{1}$, the corresponding matrix[6] is $\mathbf{U}_{2\pi} = -\mathbf{1}$! Two 2x2 complex matrices, $\mathbf{1}$ and $-\mathbf{1}$, correspond to the same 3x3 real matrix $\mathbf{1}$. When the unitary matrix $\mathbf{U}$ corresponds to one orthogonal real matrix, so does $-\mathbf{U}$. Does any physical meaning attach to the structure of such a space? [...]

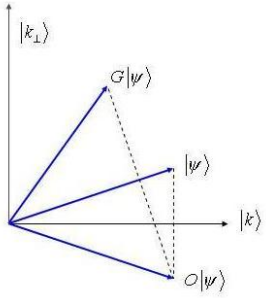
In quantum theory, incompatible observables are knitted together in a way precisely captured by its representation space. Their mutual interdependence has an essentially probabilistic character, but not of the kind found in classical physics. The symmetries reflecting any quantum state *and* its alternatives demand an "imaginary" dimension, hence *complex* probability *amplitudes*. The way in which the probability relations between quantum observables are determined by the symmetries of the Euclidean space typifies a way in which theoretical constructions are determined by symmetries in nature. The way in which quantum theory depicts those symmetries in the complex space is reminiscent of the way in which theoretical constructions are determined by perspective in art.

Complex numbers provide quantum theory with a looking glass. Through the looking glass, quantum observables become intelligible in their multiplicity and mutability. May complex numbers also throw light on computational processes?

5 Search Algorithm through the Looking Glass

Grover's search algorithm constitues a remarkable result for quantum computing. It is a tecnique for searching N possibilities in $O(\sqrt{N})$ steps. In his 2001, Grover calls for an incisive explanation:[7]

> *What is the reason that one would expect that a quantum mechanical scheme could accomplish the search in $O(\sqrt{N})$ steps*? It would be insightful to have a simple two line argument for this without having to describe the details of the search algorithm. (Grover 2001, p. 15)



Grover's algorithm consists of repeated applications of the same unitary transformation $O\left(2^{\frac{n}{2}}\right)$ times. The operation applied at each individual iteration, the "Grover iterate", can be written $G = (2\,|\psi\rangle\,\langle\psi| - \mathbf{1})\,O$. From a geometrical point of view, it appears as a rotation of $|\psi\rangle$ in the space spanned by the initial vector $|\psi\rangle$ and the state $|k\rangle$ consisting of a uniform superposition of solutions to the search problem.

In line with the argument sketched in this paper, the reason for accomplishing the search in $O(\sqrt{N})$ steps is that, in the computational space opened by complex numbers, the rotation involved in the Grover's algorithm can

[6] $$\mathbf{R}_\theta = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \Leftrightarrow \mathbf{U}_\theta = \begin{pmatrix} e^{\frac{i\theta}{2}} & 0 \\ 0 & e^{-\frac{i\theta}{2}} \end{pmatrix}$$

[7] I am grateful to Giuseppe Castagnoli for drawing my attention to this point. His answer (2008) is that any quantum algorithm takes the time of a classical algorithm knowing in advance 50% of the information that specifies the solution of the problem.

be viewd as a *double reflection* about two rays whose angular separation is *half* of the rotation angle. This double reflection, or double projection, seems to act as a sort of "counter-diagonalization" reducing the number of steps from N to the square root of N.
[...]

References

1. Altmann, S. L.: *Rotations, Quaternions, and Double Groups* (Oxford University Press, Oxford 1986)
2. Altmann, S. L.: *Icons and Symmetries* (Clarendon, Oxford 1992)
3. Bell, J. S.: Subject and Object. In: *The Physicist's Conception of Nature* (Reidel, Dordrect 1973)
4. Castagnoli, G.: The quantum speed up as advanced knowledge of the solution. In: *IJTP* (2008)
5. Chaitin G.: *Conversations with a Mathematician. Math, Art, Science and the Limits of Reason* (Springer, Berlin 2002)
6. Copeland, B. J. (ed): *The Essential Turing* (Clarendon, Oxford 2004)
7. Gandy R.: Church's Thesis and the Principles for Mechanism. In: Barwise J. *et al.* (eds): *The Kleene Symposium* (North-Holland, Amsterdam 1980)
8. Gödel K.: Undecidable diophantine propositions. In: Feferman S. *et al.* (eds): *Collected Works*, 3 voll. (Oxford Univ. Press, Oxford 1990-95)
9. Goldstein, H.: *Classical Mechanics* (Addison-Wesley, Reading Mass. 1950)
10. Grover, L. K.: From Schrödinger's equation to the quantum search algorithm (Quant-ph/0109116 2001)
11. Hilbert D.: Grundsätzliche Fragen der Modernen Physics (1923). In: Cod. Ms. Hilbert 596, SUB, Handschriftenabteilung
12. Hilbert D.: Naturerkennen und Logik. In: *Naturwissenschaften*, **18** (1930)
13. Hughes, R. I. G.: *The Structure and Interpretation of Quantum Mechanics* (Harvard Univ. Press, Cambridge Mass. 1989)
14. Turing, A. M.: On computable numbers with an application to the *Entscheidungsproblem*. In: *Proceedings of the London Mathematical Society*, Serie 2, **43** (1937). Reprinted in: Copeland 2004
15. Turing, A. M.: Intelligent Machinery (1948). In: Copeland 2004
16. von Neumann J.: Zum Hilbertschen Beweistheorie (1927) In: *Collected Works*, vol. I (Pergamon Press, Oxford 1961)
17. von Neumann J.: Unsolvable Problems in Mathematics (1954). In: Redei M., Stöltzner M. (eds): *John von Neumann and the Foundations of Quantum Physics* (Kluwer, Dordrecht 2001)
18. Weyl, H.: *Philosophy of Mathematics and Natural Sciences* (Princeton Univ. Press, Princeton 1949)
19. Weyl, H.: *Symmetry* (Princeton Univ. Press, Princeton 1952)
20. Zeilinger, A.: *Einstein Schleier. Die neue Welt der Quantenphysick* (Verlag, München 2003)

A Completeness Theorem for General Relativity

Judit Madarász, Istvan Németi, and Gergely Székely

Rényi Institute of Mathematics, Budapest
madarasz@renyi.hu, nemeti@renyi.hu, turms@renyi.hu

Abstract. This talk is based on the tutorial: Adreka et. al. Axiomatization of Physics in a logical framework. Our general aim is to axiomatize relativity theories in first-order logic. The scope of the tutorial contains many theories of relativity ranging from special relativity through general and cosmological relativity theories.
In this talk we will concentrate on general relativity. We will recall our axiom system for general relativity from the tutorial and we will also axiomatize Lorentzian manifolds in first-order logic.
Fulfilling the main aim of this talk, we are going to prove that these two axiom systems are definitionally equivalent. This theorem means that our axiomatic theory of general relativity and the theory of Lorentzian manifolds are essentially the same theory.

Access Control in a Hierarchy by Quantum Means

Naya Nagy and Selim G. Akl

School of Computing, Queen's University
nagy,akl@cs.queensu.ca

Abstract. Access control in a hierarchy refers to a selective access to a database. A large number of users work with the same database. These users are organized in a hierarchical structure and therefore have different access rights to the data.
This paper offers a solution to the problem of access control in a hierarchy based on quantum cryptography. Each user has two keys: a classical key and a quantum key. Our scheme offers several security advantages over the classical schemes to date. It protects users from identity theft and prevents collusion attacks. Most importantly though, our scheme adapts to dynamic changes of the user hierarchy: users may join, leave, or change position in the hierarchy, without affecting the rest of the user structure.

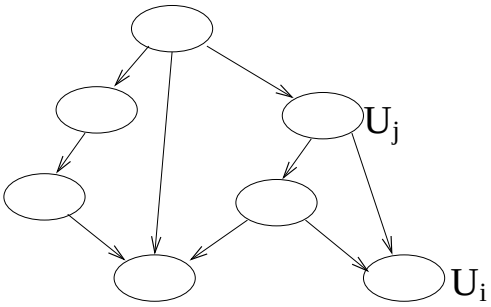


Fig. 1. Formal sets in a poset.

1 Introduction

This paper revisits the problem of access control in a hierarchy [1]. A collection of data, such as a database, is accessed by a very large number of users. Users have different access rights to the data items. Regular users are organized in groups and may access group specific data and data of general interest. Managers and

directors are able to access data belonging to a whole category or group of users and may also access data to remain secret from regular users. Groups of users are organized as a partially ordered set (poset), where each node, or group of users, represents a category with identical access rights (see Fig. 1). A node that is in a parent position in the hierarchy shares all access rights of its children.

To formalize, consider two sets U_i and U_j (see Fig. 1), that are members of the poset. The partial order $U_i \leq U_j$ means that U_i is on a lower level than U_j in the poset. Additionally, there is a line in the diagram connecting U_i with U_j. U_i, on the lower level, has less access rights than U_j, and conversely, U_j can do anything that U_i can do.

2 Access Control in a Hierarchy Using Classical Cryptography

Classical cryptographic solutions to the problem of access control build on a system of secret keys. Each user has a secret key that is used by the encryption/decryption function to transform encoded information into readable format. A manager or director, who is the root of a subtree, has a key that subsumes all keys in the subtree [4].

The mechanism that uses secret keys, generally works with a secret encryption key k^e and a secret decryption key k^d. The original text v is encrypted with an encryption function E to obtain the encrypted text u:

$$u = E_{k^e}(v).$$

The original text can be retrieved by decrypting u with the key k^d:

$$v = E_{k^d}(u).$$

Depending on the encryption method, the two keys, the encryption key and the decryption key, may coincide.

Several cryptographic solutions have been proposed. They all come with a specific range of successful applicability as well as their weakness or disadvantages:

1. First straightforward solution. The first solution (see Fig. 2) begins by assigning separate keys to all members of the poset. These keys are independent, meaning one key cannot be obtained from another. The users of authority, higher up in the poset, inherit all keys from children groups below. Though this solves the problem of access to the database, the direct disadvantage is that groups high in the poset own too many keys.

2. Solution for a totally ordered set. If instead of a poset, the users are organized in a totally ordered set (see Fig. 3), there is a simple solution that assigns exactly one key per user. We define a one-way function f and also initialize the key of the root. Consequently, the key of a child is computed as the function f of the key of the parent: $k_i = f(k_j)$ and $k_l = f(k_i)$. The child is given

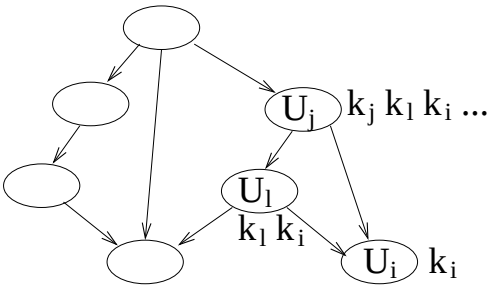


Fig. 2. Straightforward cryptographic solution.

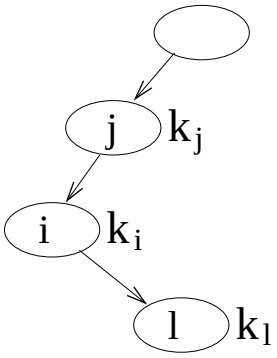


Fig. 3. Solution for a totally ordered set.

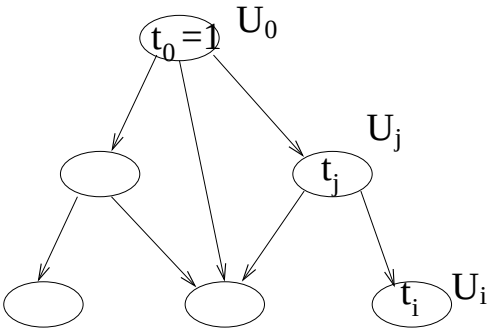


Fig. 4. Solution to a poset that computes keys in an up-down fashion. First, assign public integers t to each group in a partially ordered set, then compute the keys.

its key only. Because of the noninversability of f, the child cannot compute the keys of its ancestors.

3. Up-down computable keys for a partially ordered set. The method we describe now, combines the advantages of the previous two methods. It assigns exactly one secret key to each group of users, while preserving the order of a partially ordered set. By choice, the root is the first to be assigned a secret key K_0. This key is known only to the root, thus hidden from anybody else in the poset. Again, by choice, a number M is defined as the product of two large primes p and q: $M = p \times q$. The number M will be used for modulo operations. Additionally, a structure of integers $t_i, t_j, ...,$ is assigned to the poset (see Fig. 4). These integers are public.

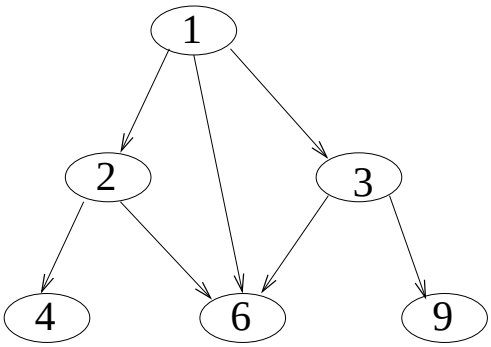


Fig. 5. The integer assigned to a child node is a common multiple of the integers of its parents.

The condition on the integers is that the integer of a parent divides the integers of its children (see Fig. 5). Formally, if $U_i \leq U_j$ then $t_j | t_i$. For a node with several parents, the integer assigned to it may be the least common multiple of the integers of all parents, or simply *some* common multiple.

Now all secret keys can be computed. For group U_i, with its integer t_i, the secret key is a power of the initial K_0:

$$K_i = {K_0}^{t_i} \bmod M.$$

Each user of some group U_i gets only the key of its group K_i.

This ingenious scheme now allows a user to compute all keys that are below in the hierarchy. For $U_i \leq U_j$, U_j, using K_j, can compute K_i, namely

$$K_i = {K_0}^{t_i} = [{K_0}^{t_j}]^{\frac{t_i}{t_j}} = [K_j]^{\frac{t_i}{t_j}} \bmod M.$$

By intention, U_i cannot compute K_j, as it is computationally intractable to extract roots modulo a large number.

This method gives simplicity to the access procedure. The major disadvantage is that it can be broken by collusion attacks. This means that users on lower levels can collaborate to compute a higher level key. The following is an example of a collusion attack. Suppose U_l, with $t_l = 4$, collaborates with U_i, with $t_i = 9$. Their secret keys are $K_l = {K_0}^4$ and $K_i = {K_0}^9$, respectively. The operation $(K_l)^{-2}K_i = {K_0}^{-8}{K_0}^9 = K_0 \bmod M$ computes the secret key of the root.

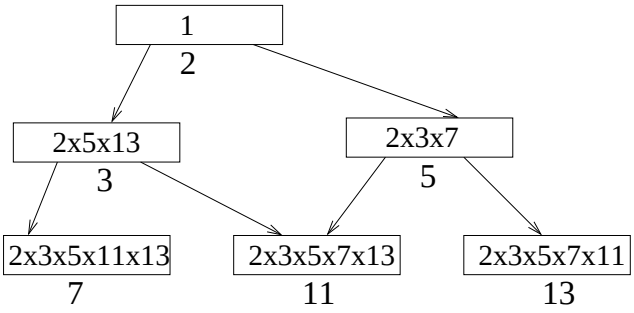


Fig. 6. Robust up-down computable keys using a structure of primes.

4. Robust up-down computable keys for a partially ordered set. The shortcomings of the previous solution can be solved, by choosing the integers t_i appropriately. First, the poset is assigned a structure of primes p_i (see Fig. 6). The integer t_i, associated with class U_i, will be defined as the product of all primes associated with nodes not below U_i in the poset.

$$t_i = \prod_{U_j \not\leq U_i} p_j$$

The secret keys are computed as before $K_i = {K_0}^{t_i} \bmod M$. The advantage of this scheme is that it eliminates collusion attacks. A parent can still easily compute the keys of its children.

This scheme satisfies an organization where the users are stable, and few people join or leave the organization over a considerable time. This may be unrealistic in real life. If a user leaves the organization, the user's secret key has to be invalidated. This means right away that the whole group that the user belonged to has to receive a new secret key. Because keys in the hierarchy are interdependent, invalidating one key may affect a whole area of the poset. In the worst case, the whole poset needs to receive new keys. The same problem arises when a user gets a promotion and becomes a member of a different group. Again, the same difficulty may appear when a new group of users are to be added. It may be easier to add a group at the bottom of the poset and more difficult to insert a group at some arbitrary level of the poset. In the literature [2] [3] [4]

[5] [7] [8] [10], numerous schemes based on cryptography have been proposed to address these problems, but none of them truly succeeds.

We will see in the next section that a quantum solution addresses many of these issues.

3 Quantum Setting for Access Control

The quantum scheme designed here takes full advantage of quantum cryptography [6] [9]. It achieves the following improvements. Any local change to a user does not affect the other users. In particular, any user may join or leave the system without affecting the other members of the user community. Also, if a user changes its position in the hierarchy, such as being promoted to a manager position, it is only this user's key that will have to be changed.

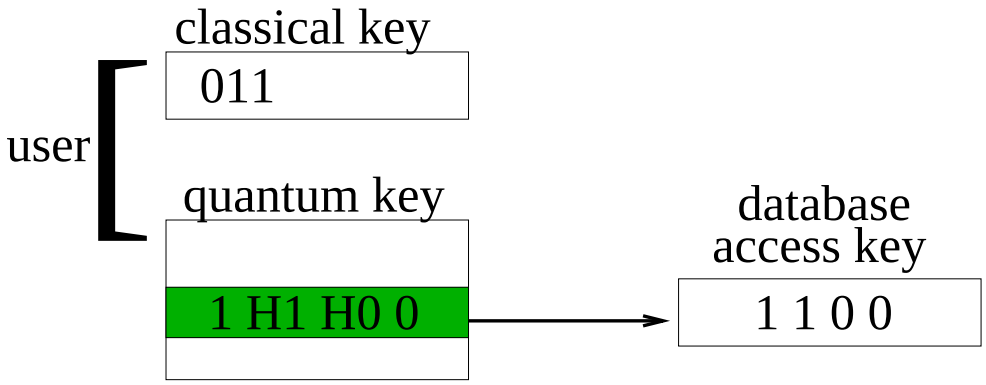


Fig. 7. Each user has two keys: a classical key and a quantum key.

3.1 Quantum Card and Classical Key

The access to the database is managed by two keys. Every user has **two keys**: a classical key and a quantum key (see Fig. 7). The purpose of the user's keys is to provide the information necessary to generate a database access key, *dbAccess*. The database access key is the one that defines the access rights of the user to the database. *dbAccess* is not in the direct possession of the user. And again the two keys that *are* in the possession of the user serve the sole purpose to retrieve *dbAccess* and are meant to *hide* the value of *dbAccess* from the user.

The user's *classical key* is a binary number. This number is unique for each user and secret, expected to be known to that user only. It is the equivalent of a password and thus it is the user's responsibility to keep it secret.

The user's *quantum key* is an array of qubits and is registered on a card. The quantum key is a quantum encrypted version of the database access key *dbAccess*. This key may be unique to the user or even unique for each session. This means that each time a user connects to the database, the quantum card may have another quantum key written on it.

The quantum key is not known to the user. Although, the key is written on the card, the user does not need to know its qubit values. The qubits written on the card are in different quantum states. Some states represent classical values, such as $|0\rangle$ and $|1\rangle$. Other qubits are in a balanced superposition of $|0\rangle$ and $|1\rangle$, namely $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = H|0\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|1\rangle$. As the user does not know which qubits are simple states and which are in a superposition, the user cannot retrieve his/her *dbAccess* key by illicitly reading the card. Moreover, reading the card destroys the quantum states of the qubits, as they collapse to some classical value. Thus, a card that has been illicitly read, cannot be used afterwards to connect to the database.

The quantum encryption is not unique. Each bit of *dbAccess* may be quantum encrypted in four different ways:

1. *****: Copied directly with no change.
2. **NOT**: The bit is negated.
3. **H**: The bit is transformed with a Hadamard gate.
4. **H NOT**: The bit is negated and then transformed with a Hadamard gate.

Access Key	Quantum Key	Decryption Mask
1100	0 1 H0 0	$NOT * H *$
	H0 H1 0 1	$(H\ \ NOT)\ H * NOT$
	H1 1 H0 H0	$H * H\ H$
	H0 1 H0 H1	$(H\ \ NOT) * H\ (H\ \ NOT)$

Table 1. There are 4^n quantum keys that encrypt the same access key. The decryption mask yields the reading strategy to obtain the access key.

If the access key is n bits long, there are 4^n possible quantum keys that encrypt the same *dbAccess* key. In table 1 the second column shows possible encryptions of a short example-key $dbAcces = 1100$.

3.2 The Access Control Unit

To know how to retrieve the access key from the array of qubits of the quantum key, we need to have a decryption mask. The decryption mask simply says how to read the qubits of the quantum key in order to obtain the intended binary value. It shows the positions in the qubit array of the quantum key that are in superposition and/or negated. The third column of table 1 defines the decryption masks for the quantum keys of the second column.

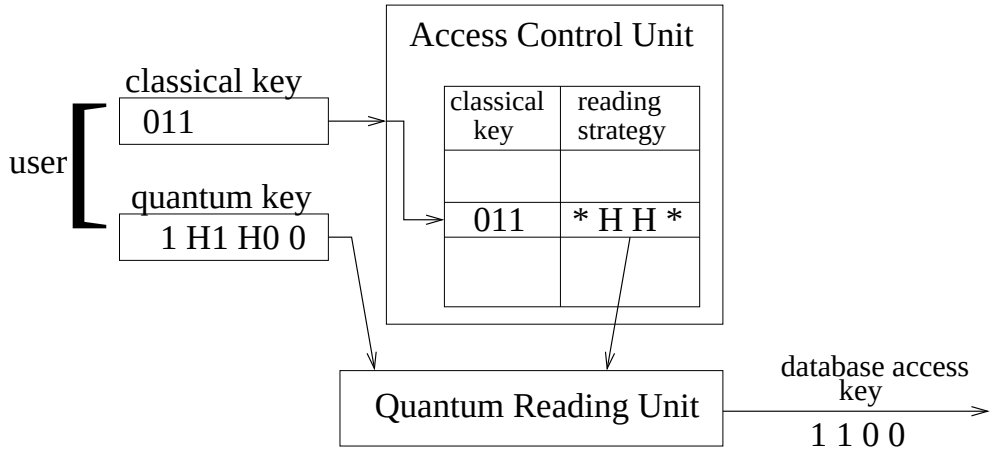


Fig. 8. How the database access key is obtained.

In order to manage the decryption of the quantum keys, there is a unit attached to the database, called the access control unit, ACU (see Fig. 8). The ACU translates the two user keys into the final database access key. The ACU has a table that has entries for each user's classical key. A decryption mask, or reading strategy mask, corresponds to each classical key value. This reading strategy mask is then submitted to the quantum reading unit. This unit is now able to correctly read the quantum key. Simple qubits are read directly and qubits in superposition are first transformed by a Hadamard gate. If necessary, the bits are then negated. Thus, the output of the quantum reading unit is the final database access key.

Note that the ACU is attached to the database. It is not visible to the user. Once deployed, the ACU does not need to be managed by a human.

When a user accesses the database, the user has to type in the classical key and also provide the quantum card for reading. Whenever the card is used, the quantum key is destroyed by reading. Therefore, at the end of each session, the card needs to be restored, meaning that the quantum key is written back to the card. It is interesting to note, that the quantum key need not be the same. At the end of the session, the ACU may generate a new, random reading mask, and then write a new quantum key on the card. Thus, the user has a quantum key per session.

3.3 Changes in the User Structure of the Organization

The clear advantage of this scheme is that the user is disconnected from the database access key. The user has no knowledge of its value and no way of retrieving information about it.

Now the *dbAccess* key defines the access rights of the user as a member of the poset. As *dbAccess* is hidden from the user, a hierarchical structure, as the ones described in section 2, will serve the purpose. For example, the *dbAccess* can be obtained by the up-down computable key method.

When a user joins an existing group, this means that the node of the poset exists and is working. The particular access key of the group will be assigned to the new user, using some arbitrary quantum encryption. A line is added to the ACU's table to represent the new user. In addition, a classical key will be given to the user. This key is independent of the poset structure and is an index in the ACU's table. When a user leaves the organization, its line in the ACU table is invalidated. Therefore, there is no entry in the ACU for this particular classical key. The user can no longer access the ACU with the classical key. Normally, the user would be required to return the quantum card, but this does not affect the security of the system, as will be seen in the next subsection.

When a user changes its position in the hierarchy of the poset, its quantum card needs to be updated to a quantum encryption of the new database access key. The quantum card will reflect the change in the access rights. The classical key may remain the same. Also, the ACU table needs to be updated with a new decryption mask.

Note that all changes described above affect exactly one user, namely the user whose status is changed. This is remarkable, compared to all previous classical solutions existing in the literature, as described in section 2. In our scheme, a change in the status of an arbitrary user leaves all the rest of the users undisturbed. This is an important advantage, when considering large organizations with millions of users and presumably a very dynamic structure.

The scheme designed is less adaptable to changes in the poset structure itself. Adding another leaf to the poset, that is, creating a new group, should pose no problems, as a new *dbAccess* can be created to define the node. This key would be some common multiple of its parents. Yet, it might be difficult to insert a node in some arbitrary position of the poset, as an appropriate *dbAccess* key might not be available. Deleting a group of users is easy again, as it simply means to stop using a certain database access key.

3.4 What the Intruder Can/Cannot Do

Let us consider first that the intruder has access to the property of the users only, but cannot access the ACU, as it is stored in a secure place.

If the intruder, Eve, steals the classical key, she will have absolutely no access to the system without the quantum card. In the same way, if Eve steals the quantum card, but does not know the classical key, she cannot access the ACU.

Because of the nonclonability theorem, Eve cannot copy the quantum card. If Eve tries to read the quantum card, she destroys the quantum key, and the card will be unusable to the legitimate user as well. As Eve does not know the decryption mask, there is no way of reading the quantum card and gaining some knowledge about *dbAccess*. In fact, if Eve guesses a reading strategy, the probability on each qubit to be measured correctly is still 50%.

Therefore, Eve's only option is to steal both the classical key and the quantum card. Note that this is a complete identity theft. The legitimate user has lost its quantum card. Yet, this theft is *detectable*, the legitimate user will *know* that his/her identity has been stolen: *the user cannot find his/her card*. In this case, the user's identity has to be invalidated from the cryptographic system, and a new identity has to be given to the user. Again, an aspect specific to this quantum system is that an identity cannot be copied. It is not possible that two persons carry the same cryptographic identity.

Let us consider now that Eve may gain access to the ACU. The ACU has no *dbAccess* key stored into it. Just looking at the ACU's table does not reveal anything about the access key, as the access key is solely written on the quantum card. Therefore, Eve has absolutely no gain from looking at the ACU, unless she also has both a classical key and a quantum key. This means that Eve would need to steal both the identity of a user and access the ACU in a very short interval, which is practically difficult.

Also, two or more users cannot collaborate to break the system. They cannot even gain knowledge about their own *dbAccess*. This is because their quantum keys are different and have no meaningful connection to the value of *dbAccess*, except through the decryption mask.

4 Conclusion

Our scheme shows that adding quantum keys to the access mechanism of a database has advantages both in terms of security of the system and of adaptability to changes in the underlying user structure.

The system cannot be broken easily, as the quantum key cannot be copied, and in fact may be unique for the session. The identity of a user cannot be stolen without the user noticing the theft.

Also the system is designed to support a large variety of changes in the user structure. Users may join and leave the organization without affecting the security system at large.

The idea of using two keys, a classical key and a quantum key, is not necessarily connected to this specific application, namely access control in a hierarchy. In fact, the access system behind the two-key front end, may have any structure. The idea may be successfully applied, whenever the user is to be distanced from the actual security access of the sensitive data. In our scheme, the user is distanced from the hierarchical structure of the users' security rights. It is the specific value of *dbAccess* for each user, which reflects the security rights. The *dbAccess* keys of all users form the poset structure and are therefore considered to be defined according to the up-down computable method.

The vulnerability of the database itself, or the ACU has not been treated in this scheme. As a future work, we envision to store both the database and the table of the ACU using quantum memories. This would allow the definition of a security scheme for these data based on a quantum cryptographic approach. For example, the database could be quantum encrypted.

References

1. Selim G. Akl and Peter D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems*, 1(3):239–248, 1983.
2. G. Ateniese, A. De Santis, A. L. Ferrara, and B. Masucci. Provably-secure time-bound hierarchical key assignment schemes. In *Proceedings of 13th ACM Conference on Computer and Communication Security (CCS'06)*, pages 288–297, 2006.
3. J. Crampton. Cryptographically-enforced hierarchical access control with multiple keys. In *Proceedings of 12th Nordic Workshop on Secure IT Systems (NordSec 2007)*, pages 49–60, 2007.
4. Stephen J. MacKinnon, Peter D. Taylor, Henk Meyer, and Selim G. Akl. An optimal algorithm for assigning cryptographic keys to control access in a hierarchy. *ACM Transactions on Computers*, c-34(9):797–802, 1985.
5. I. Ray and N. Narasimhamurthi. A cryptographic solution to implement access control in a hierarchy and more. In *Proceedings of the 7th ACM Symposium on Access Control Models and Technologies*, pages 65–73, Monterey, CA, 2002.
6. Jr. S. J. Lomonaco. A Talk on Quantum Cryptography or How Alice Outwits Eve. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 237–264, Washington, DC, January 2002.
7. R. Sandhu. Cryptographic implementation of tree hierarchy for access control. *Information Processing Letters*, 27:1–100, January 1988.
8. V. Shen and T. Chen. A novel key management scheme based on discrete logarithms and polynomial interpolations. *Computers and Security*, 21(2):164–171, 2002.
9. L. Vaidman. Teleportation of quantum states. *Phys. Rev. A*, 49(2):1473–1476, Feb 1994.
10. C. Yang and C. Li. Access control in a hierarchy using one-way functions. *Elsevier: Computers and Security*, 23:659–644, 2004.

Physics and Proof Theory

Bruno Woltzenlogel Paleo[1,2]

[1] Institut für Computersprachen, Vienna University of Technology, Austria
[2] INRIA, LORIA, Nancy, France
bruno@logic.at, Bruno.WoltzenlogelPaleo@loria.fr

Abstract. Axiomatization of Physics (and Science in general) has many drawbacks that are correctly criticized by opposing philosophical views of Science. This paper shows that, by giving formal proofs a more prominent role in the formalization, many of the drawbacks can be solved and many of the opposing views are naturally conciliated. Moreover, this approach allows, by means of Proof Theory, to open new conceptual bridges between the disciplines of Physics and Computer Science.

Keywords: Proof Theory, Physics, Formalization of Science

1 Introduction

"Science is built up with facts, as a house is with stones.
But a collection of facts is no more a science
than a heap of stones is a house."
- Poincaré

Foundational works on the formalization of Physics typically consider a physical theory as a collection of facts, i.e. as a set of sentences closed under logical consequence. However, not as much attention has been given to studying how these facts are or should be organized or, equivalently, how the physical theory is or should be structured. Usually, the only structure considered is a distinction of facts either as axioms or as derivable theorems (i.e. axiomatization). Although simple, this approach has a few drawbacks.

Firstly, from an epistemological point of view, the mentioned approach suffers from a logical omniscience problem: although physicists might know the axioms of their theories, it is certainly not the case that they know all the logical consequences of these axioms, simply because they have limited reasoning resources. Therefore, the approach of defining a theory as a set of sentences closed under logical consequence fails to capture the notion of theory as perceived by resource-bounded physicists; it is just an idealized approximation.

Secondly, the selection of which facts should be taken as axioms is arbitrary and frequently based on subjective criteria such as elegance. For example, there are axiomatizations of physics that do not rely on the rather natural concepts of space and time [18]. Should they be considered more elegant, useful or correct?

And finally, there are cases of physical theories, such as Newtonian mechanics and Lagrangean mechanics, that are considered equivalent to each other according to the mentioned approach, because their sets of sentences closed under logical equivalence are the same, even though they actually differ significantly in how easily they can be used to solve certain classes of problems.

The second and third drawbacks mentioned above have been main reasons for criticism on the whole enterprise of formalizing Science [20]. However, they actually only apply to (unstructured) axiomatization. As a response to the criticism, there was a rise of semantic approaches, which adopted a more model-theoretic approach to the formalization of Science [20]. Advances in the sibling discipline of proof theory, on the other hand, have not been given much attention.

The main goal of this paper is to advocate in favor of a more prominent role for proofs in the formalization of physics, and consequently, for proof theory in approaches to Hilbert's sixth problem [22] and in studies of the foundations of physics. If a physical theory is considered not as a collection of sentences closed under logical consequence, but rather as a collection of proofs, the above mentioned drawbacks are naturally solved. Non-idealized resource-bounded physicists know only what they have proved so far. Axioms are simply the assumptions of the proofs contained in the physical theory. And various physical theories can be objectively compared with respect to the structure of the proofs they contain. This proposal is in line with current work in the formalization of mathematics, where mathematical knowledge is formalized as collections of proofs with the assistance of interactive theorem provers[3].

The use of proofs to formalize computations of solutions of physical problems is exemplified with a simple problem of Newtonian mechanics in Section 3. The proof calculus used, known as sequent calculus, is briefly explained in Section 2. Finally, Section 4 discusses the benefits and challenges of using proofs in the formalization of Physics, from philosophical and computational points of view.

2 The Sequent Calculus $\mathbf{LK^P}$

The formal proofs in this paper are written in an extension of Gentzen's sequent calculus **LK** [11]. A *sequent* is a pair $\Gamma \vdash \Delta$, where Γ (the antecedent) and Δ (the succedent) are multisets of formulas, with the intuitive intended meaning that the disjunction of the formulas in Δ is provable assuming the formulas in Γ. An **LK**-proof is a (hyper)tree of sequents, such that the leaves are *axiom sequents* of the form $F \vdash F$, where F is an arbitrary formula, and the (hyper)edges are instances of the inference rules specified by the calculus. The sequent calculus **LK** has inference rules for propositional connectives (e.g. $\vee$, $\rightarrow$, $\neg$ and $\wedge$), as exemplified below for the $\wedge$ connective:

$$\frac{\Gamma \vdash \Delta, A \quad \Pi \vdash \Lambda, B}{\Gamma, \Pi \vdash \Delta, \Lambda, A \wedge B} \wedge : r \qquad \frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge : l1 \qquad \frac{A, \Gamma \vdash \Delta}{B \wedge A, \Gamma \vdash \Delta} \wedge : l2$$

[3] Examples of proof assistants are Mizar (http://mizar.uwb.edu.pl/), Coq (http://coq.inria.fr/) and Isabelle (http://www.cl.cam.ac.uk/research/hvg/Isabelle/).

The following inference rules for quantifiers are also available (with the important restriction that the $\forall : r$ and $\exists : l$ rules must satisfy the eigenvariable condition, i.e. the variable α must occur neither in Γ nor in Δ nor in A):

$$\frac{A\{x \leftarrow t\}, \Gamma \vdash \Delta}{(\forall x)A, \Gamma \vdash \Delta} \; \forall : l \qquad \frac{\Gamma \vdash \Delta, A\{x \leftarrow \alpha\}}{\Gamma \vdash \Delta, (\forall x)A} \; \forall : r$$

$$\frac{A\{x \leftarrow \alpha\}, \Gamma \vdash \Delta}{(\exists x)A, \Gamma \vdash \Delta} \; \exists : l \qquad \frac{\Gamma \vdash \Delta, A\{x \leftarrow t\}}{\Gamma \vdash \Delta, (\exists x)A} \; \exists : r$$

Moreover, the sequent calculus **LK** also provides structural rules such as contraction, weakening and, most importantly, the cut rule, which, as discussed in Section 4, eases the structured formalization of Physics:

$$\frac{\Gamma \vdash \Delta, F \quad F, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \; cut$$

However, the pure sequent calculus **LK** does not provide any built-in support for equality handling, arithmetical simplifications, and differentiation and integration. Therefore, formalizing physics in the pure sequent calculus **LK** would be tedious and uncomfortable, since the lack of built-in support would require adding several additional assumptions to the antecedents of the sequents, which would render the proofs large, unreadable and difficult to construct. The sequent calculus $\mathbf{LK^P}$ addresses this issue by extending **LK** with the following rules:

- **Built-in Support for Equality:**

$$\frac{\Gamma, s = t, A[t] \vdash \Delta}{\Gamma, s = t, A[s] \vdash \Delta} \; =_l \qquad \frac{\Gamma, s = t \vdash \Delta, A[t]}{\Gamma, s = t \vdash \Delta, A[s]} \; =_r$$

$$\frac{\Gamma, s = t, A[s] \vdash \Delta}{\Gamma, s = t, A[t] \vdash \Delta} \; =_l \qquad \frac{\Gamma, s = t \vdash \Delta, A[s]}{\Gamma, s = t \vdash \Delta, A[t]} \; =_r$$

  where s and t do not contain variables that are bound in A.
- **Built-in Support for Definitions:**[4] They correspond directly to the *extension principle* and introduce new predicate and function symbols as abbreviations for formulas and terms. Let $A[x_1, \ldots, x_k]$ be an arbitrary formula with free-variables $x_1, \ldots, x_k$ and P be a *new* k-ary predicate symbol defined by $P(x_1, \ldots, x_k) \leftrightarrow A[x_1, \ldots, x_k]$. Let $t[x_1, \ldots, x_k]$ be an arbitrary term with free-variables $x_1, \ldots, x_k$ and f be a *new* k-ary function symbol defined by $f(x_1, \ldots, x_k) = t[x_1, \ldots, x_k]$. Then, for arbitrary sequences of terms $t_1, \ldots, t_k$, the rules are:

$$\frac{A[t_1, \ldots, t_k], \Gamma \vdash \Delta}{P(t_1, \ldots, t_k), \Gamma \vdash \Delta} \; d_l \qquad \frac{\Gamma \vdash \Delta, A[t_1, \ldots, t_k]}{\Gamma \vdash \Delta, P(t_1, \ldots, t_k)} \; d_r$$

$$\frac{F[t[t_1, \ldots, t_k]], \Gamma \vdash \Delta}{F[f(t_1, \ldots, t_k)], \Gamma \vdash \Delta} \; d_l \qquad \frac{\Gamma \vdash \Delta, F[t[t_1, \ldots, t_k]]}{\Gamma \vdash \Delta, F[f(t_1, \ldots, t_k)]} \; d_r$$

[4] Definition rules have been succesfully used for formalization and analysis of mathematical proofs [3]. They are closely related to *superdeduction rules* [6], which can provide even more concise, natural and readable formal proofs. However they are not as simple to describe, and hence definition rules have been used in this paper.

– **Built-in Support for Simplification:** let t (or t') be obtainable from t' (t) by algebraic or arithmetical simplifications[5]. Then the rules are:

$$\frac{F[t'], \Gamma \vdash \Delta}{F[t], \Gamma \vdash \Delta}\ s_l \qquad\qquad \frac{\Gamma \vdash \Delta, F[t']}{\Gamma \vdash \Delta, F[t]}\ s_r$$

– **Built-in Support for Integration and Differentation:**[6] let t_1 (t_2) be a term denoting the integral of the function denoted by t'_1 (t'_2) on the interval (x_1, x_2). Then the rules are:

$$\frac{F[t'_1 = t'_2], \Gamma \vdash \Delta}{F[t_1 = t_2], \Gamma \vdash \Delta}\ \int_{x_1}^{x_2} : l \qquad\qquad \frac{\Gamma \vdash \Delta, F[t'_1 = t'_2]}{\Gamma \vdash \Delta, F[t_1 = t_2]}\ \int_{x_1}^{x_2} : r$$

3 A Simple Example: Energy Conservation as a Cut

To solve problems of physics, certain invariants (such as energy) are frequently used. This is so because solving problems by using a derived principle (such as the principle of energy conservation) is usually easier than solving them by using the most basic physical laws or axioms. This section intends to exemplify how problem solution can generally be seen from a proof-theoretic perspective in which the use of derived principles corresponds to an implicit use of the cut rule. The following simple problem of Newtonian mechanics shall be considered:

> An object of mass m is dropped from height h_0 and with initial velocity equal to zero. The only force acting on the object is the force of gravity (with an intensity mg). What is the velocity of the object when its height is equal to zero?

A typical solution (Solution 1) to this problem uses the principle of energy conservation, as follows:

1. Let t_f be the time when the object reaches height zero.
2. According to the principle of energy conservation, $e(t_f) = e(0)$, i.e. the energy at t_f is equal to the initial energy.
3. Hence, by definition of gravitational potential energy in a uniform gravitational field and by definition of kinetic energy, $mgh(t_f) + m\frac{\dot{h}(t_f)^2}{2} = mgh(0) + m\frac{\dot{h}(0)^2}{2}$.

[5] It is beyond the scope of this paper to define precisely the allowed simplifications. This kind of rule is inspired by *deduction modulo*, whose precise definitions can be found in [9]. In principle, simplification rules are not necessary, because they can be simulated by equality rules together with the arithmetical and algebraic axioms as additional assumptions in the antecedentes of the sequents. However, the goal of simplification rules (and deduction modulo) is to hide uninteresting computational details of the underlying theories (e.g. arithmetics), in order to obtain concise formal proofs that show only interesting information related to the theory under investigation (e.g. newtonian mechanics).

[6] Integration and Differentiation Rules have been inspired by emerging idea of integrating computer algebra systems and automated theorem provers.

4. According to the initial conditions, $h(0) = h_0$ and $\dot{h}(0) = 0$. Moreover, by assumption, $h(t_f) = 0$. Therefore, $m\frac{\dot{h}(t_f)^2}{2} = mgh_0$.
5. Hence, the result is $\dot{h}(t_f) = -\sqrt{2gh_0}$.

Another solution (Solution 2) computes the velocity as a function of time by integrating the acceleration produced by the gravitational force. Then it determines the time when the object reaches height zero, and computes the velocity at that time. The details are shown below:

1. According to Newton's second law of motion, $f(t) = m\ddot{h}(t)$ at any time t. Moreover, the uniform gravitational field produces a force $f(t) = -mg$. Hence, $\ddot{h}(t) = -g$.
2. By integration, $\dot{h}(t) = -gt + \dot{h}(0)$.
3. According to the initial conditions, $\dot{h}(0) = 0$, and hence $\dot{h}(t) = -gt$.
4. By integration again, $h(t) = -g\frac{t^2}{2} + h(0)$.
5. According to the initial conditions, $h(0) = h_0$, and hence $h(t) = -g\frac{t^2}{2} + h_0$.
6. For $h(t_f) = 0$ to hold, it must be the case that $t_f = \sqrt{\frac{2h_0}{g}}$.
7. Hence $\dot{h}(t_f) = -g\sqrt{\frac{2h_0}{g}}$, which can be simplified to $\dot{h}(t_f) = -\sqrt{2gh_0}$.

Solution 2 is simpler in the sense that it uses only the basic physical laws of motion (here assumed to be Newton's laws of motion) and of uniform gravitational fields. Solution 1, on the other hand, assumes that energy is conserved, without actually proving it from Newton's basic laws.

In order to view problem solving from a proof theoretic perspective, it is necessary to formalize problem solving as theorem proving. In the example above, the problem can be stated as the following theorem to be proved:

$$(\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

Solving the given problem then consists of finding a proof of the theorem above such that v is instantiated by a ground term. Interestingly, formalizing the problem as a theorem to be proved enforces the explicit mention of the hidden assumption that the height eventually becomes zero; otherwise the variable t' would be free and the theorem would be open.

Traditionally, works of axiomatization have formalized physical laws as axioms that are supposed to be used as assumptions in proofs [20]. In a more modern proof-theoretical approach, however, definition rules often provide a more convenient alternative. The axioms corresponding to certain physical laws can be seen as defining new symbols. This is the case, for example, of Newton's second law, which states that force equals mass times acceleration ($f(t) = m\ddot{h}(t)$). It can be seen as defining the function symbol f. Similarly, the equation for energy of a single object in a uniform newtonian gravitational field ($e(t) = mgh(t) + m\frac{\dot{h}(t)^2}{2}$) can be seen as defining the function symbol e.For convenience, the defined predicate symbols below are also used in the following formal proofs:

Initial Conditions: $I \leftrightarrow \mathit{Init} \leftrightarrow h(0) = h_0 \wedge \dot{h}(0) = 0$
Uniform Gravitation: $G \leftrightarrow \mathit{Gravity} \leftrightarrow (\forall t)(f(t) = -mg)$
Fall of the Object: $F \leftrightarrow \mathit{Fall} \leftrightarrow (\exists t)\ h(t) = 0$
Energy Conservation: $EC \leftrightarrow \mathit{EnergyConservation} \leftrightarrow (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j)$

Solution 1 can be easily formalized as the proof φ_1 below (where φ_1' is a subproof consisting of the single axiom sequent $h(t_f) = 0 \vdash h(t_f) = 0$):

$$\dot{h}(t_f) = -\sqrt{2gh_0} \vdash \dot{h}(t_f) = -\sqrt{2gh_0}$$

$$\exists_r$$

$$\dot{h}(t_f) = -\sqrt{2gh_0} \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$s_l$$

$$mg0 + m\frac{\dot{h}(t_f)^2}{2} = mgh_0 + m\frac{0^2}{2} \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$w_l$$

$$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, mg0 + m\frac{\dot{h}(t_f)^2}{2} = mgh_0 + m\frac{0^2}{2} \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$=_l$$

$$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, mgh(t_f) + m\frac{\dot{h}(t_f)^2}{2} = mgh(0) + m\frac{\dot{h}(0)^2}{2} \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$d_l$$

$$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, e(t_f) = e(0) \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$\forall_l$$

$$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$\forall_l$$

$$\varphi_1' \qquad h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash (\exists v)\ \dot{h}(t_f) = v$$

$$\wedge_r$$

$$h(t_f) = 0, h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash h(t_f) = 0 \wedge (\exists v)\ \dot{h}(t_f) = v$$

$$c_l$$

$$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash h(t_f) = 0 \wedge (\exists v)\ \dot{h}(t_f) = v$$

$$\exists_r$$

$$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$\exists_l$$

$$(\exists t)\ h(t) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$\wedge_l$$

$$(\exists t)\ h(t) = 0, h(0) = h_0 \wedge \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$d_l$$

$$\mathit{Fall}, \mathit{Init}, \mathit{EnergyConservation} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

Solution 2 can be formalized as the following proof φ_2:

$$h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0$$

$$w_l$$

$$h(0) = h_0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0$$

$$\dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -\sqrt{2gh_0} \vdash \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -\sqrt{2gh_0}$$

$$\exists_r$$

$$\dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -\sqrt{2gh_0} \vdash (\exists v)\ \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$$

$$s_l$$

$$\dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -g\sqrt{\frac{2h_0}{g}} \vdash (\exists v)\ \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$$

$$\forall_l$$

$$(\forall t)(\dot{h}(t) = -gt) \vdash (\exists v)\ \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$$

$$w_l$$

$$\dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists v)\ \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$$

$$\wedge_r$$

$$h(0) = h_0, \dot{h}(0) = 0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \wedge (\exists v)\ \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$$

$$\exists_r$$

$$h(0) = h_0, \dot{h}(0) = 0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$s_l$$

$$h(0) = h_0, \dot{h}(0) = 0, h\left(\sqrt{\frac{2h_0}{g}}\right) = -g\frac{\left(\sqrt{\frac{2h_0}{g}}\right)^2}{2} + h_0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$\forall_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(h(t) = -g\frac{t^2}{2} + h_0), (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$=_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(h(t) = -g\frac{t^2}{2} + h(0)), (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$\textstyle\int_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt), (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$c_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$s_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + 0) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$=_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + \dot{h}(0)) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$\textstyle\int_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\ddot{h}(t) = -g) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$s_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(m\ddot{h}(t) = -mg) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$d_l$$

$$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$\wedge_l$$

$$h(0) = h_0 \wedge \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

$$d_l$$

$$\mathit{Init}, \mathit{Gravity} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)$$

As expected φ_1 is not only smaller than φ_2, but also simpler in the sense that it does not use integration. Furthermore, while in φ_2 the time when the object hits the floor has to be computed explicitly (i.e. t' is instantiated to a ground term), in φ_1 this is not so (i.e. t' is instantiated to a variable).

Solution 1 implicitly uses cuts, because *EnergyConservation* and *Fall* are not considered to be basic laws of physics. In principle, φ_1 must be composed with a proof φ_E of *EnergyConservation* and a proof φ_F of *Fall*. This is done with two cuts, as shown in the following proof φ:

$$\dfrac{\dfrac{\dfrac{\varphi_F}{\mathit{Init}, \mathit{Gravity} \vdash \mathit{Fall}} \quad \dfrac{\dfrac{\varphi_E}{\mathit{Gravity} \vdash \mathit{EC}} \quad \dfrac{\varphi_P}{\mathit{Init}, \mathit{Fall}, \mathit{EC} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)}}{\mathit{Init}, \mathit{Gravity}, \mathit{Fall} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)}\ cut}{\mathit{Init}, \mathit{Init}, \mathit{Gravity}, \mathit{Gravity} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)}\ cut}{\mathit{Init}, \mathit{Gravity} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\ \dot{h}(t') = v)}\ c_l$$

Where φ_F is the proof below, proving that the object will eventually fall to height zero under the gravitational field and the initial conditions specified in the description of the problem:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{h(0) = h_0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0}{h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash (\exists t')\ h(t') = 0}\ \exists_r}{h\left(\sqrt{\frac{2h_0}{g}}\right) = -g\frac{\left(\sqrt{\frac{2h_0}{g}}\right)^2}{2} + h_0 \vdash (\exists t')\ h(t') = 0}\ s_l}{(\forall t)(h(t) = -g\frac{t^2}{2} + h_0) \vdash (\exists t')\ h(t') = 0}\ \forall_l}{h(0) = h_0, (\forall t)(h(t) = -g\frac{t^2}{2} + h_0) \vdash (\exists t')\ h(t') = 0}\ w_l}{h(0) = h_0, (\forall t)(h(t) = -g\frac{t^2}{2} + h(0)) \vdash (\exists t')\ h(t') = 0}\ =_l}{h(0) = h_0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')\ h(t') = 0}\ \int_l}{h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + 0) \vdash (\exists t')\ h(t') = 0}\ w_l}{h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + \dot{h}(0)) \vdash (\exists t')\ h(t') = 0}\ s_l}{h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\ddot{h}(t) = -g) \vdash (\exists t')\ h(t') = 0}\ =_l}{h(0) = h_0, \dot{h}(0) = 0, (\forall t)(m\ddot{h}(t) = -mg) \vdash (\exists t')\ h(t') = 0}\ \int_l}{h(0) = h_0, \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')\ h(t') = 0}\ d_l}{h(0) = h_0 \wedge \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')\ h(t') = 0}\ \wedge_l}{\mathit{Init}, \mathit{Gravity} \vdash \mathit{Fall}}\ d$$

And φ_E is the proof that energy is conserved in a uniform gravitational field:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash gh(0)+\frac{\dot{h}(0)^2}{2}=gh(0)+\frac{\dot{h}(0)^2}{2}}\,w_l}{(\dot{h}(\alpha)=-g\alpha+\dot{h}(0)),(h(t)=-g\frac{\alpha^2}{2}+\dot{h}(0)\alpha+h(0)),(\dot{h}(\beta)=-g\beta+\dot{h}(0)),(h(\beta)=-g\frac{\beta^2}{2}+\dot{h}(0)\beta+h(0))\vdash gh(0)+\frac{\dot{h}(0)^2}{2}=gh(0)+\frac{\dot{h}(0)^2}{2}}\,{=}^*_r,{s}^*_r}}{(\dot{h}(\alpha)=-g\alpha+\dot{h}(0)),(h(t)=-g\frac{\alpha^2}{2}+\dot{h}(0)\alpha+h(0)),(\dot{h}(\beta)=-g\beta+\dot{h}(0)),(h(\beta)=-g\frac{\beta^2}{2}+\dot{h}(0)\beta+h(0))\vdash gh(\alpha)+\frac{\dot{h}(\alpha)^2}{2}=gh(\beta)+\frac{\dot{h}(\beta)^2}{2}}\,\forall_l}}{(\forall t)(\dot{h}(t)=-gt+\dot{h}(0)),(\forall t)(h(t)=-g\frac{t^2}{2}+\dot{h}(0)t+h(0)),(\forall t)(\dot{h}(t)=-gt+\dot{h}(0)),(\forall t)(h(t)=-g\frac{t^2}{2}+\dot{h}(0)t+h(0))\vdash gh(\alpha)+\frac{\dot{h}(\alpha)^2}{2}=gh(\beta)+\frac{\dot{h}(\beta)^2}{2}}\,c_l}}{\dfrac{(\forall t)(\dot{h}(t)=-gt+\dot{h}(0)),(\forall t)(h(t)=-g\frac{t^2}{2}+\dot{h}(0)t+h(0))\vdash gh(\alpha)+\frac{\dot{h}(\alpha)^2}{2}=gh(\beta)+\frac{\dot{h}(\beta)^2}{2}}{(\forall t)(\dot{h}(t)=-gt+\dot{h}(0))\vdash gh(\alpha)+\frac{\dot{h}(\alpha)^2}{2}=gh(\beta)+\frac{\dot{h}(\beta)^2}{2}}\,\int_l}\,c_l}}{\dfrac{(\forall t)(\ddot{h}(t)=-g)\vdash gh(\alpha)+\frac{\dot{h}(\alpha)^2}{2}=gh(\beta)+\frac{\dot{h}(\beta)^2}{2}}{(\forall t)(m\ddot{h}(t)=-mg)\vdash mgh(\alpha)+m\frac{\dot{h}(\alpha)^2}{2}=mgh(\beta)+m\frac{\dot{h}(\beta)^2}{2}}\,d_r}\,s}}{\dfrac{\dfrac{(\forall t)(m\ddot{h}(t)=-mg)\vdash e(\alpha)=e(\beta)}{(\forall t)(f(t)=-mg)\vdash e(\alpha)=e(\beta)}\,d_l}{(\forall t)(f(t)=-mg)\vdash e(\alpha)=e(\beta)}\,\forall_r}}{(\forall t)(f(t)=-mg)\vdash(\forall t_i)(\forall t_j)\ e(t_i)=e(t_j)}\,\forall_r}}{Gravity\vdash EnergyConservation}\,d$$

4 Benefits and Challenges of a Proof-Theoretical Approach to the Formalization of Physics

The following subsections are devoted to discussing what proof theory has to offer to the formalization of Physics, with emphasis on computational and philosophical aspects.

4.1 Cut-Introduction

The example discussed in the previous section illustrates that an essential task of theoretical science is to invent or discover important concepts that are useful to solve problems, such as the principle of energy conservation in newtonian mechanics. Nevertheless, in a traditional axiomatization approach, such principles have no prominent role, because they are merely theorems derivable from the axioms. In a more proof-theoretic approach, on the other hand, proofs allow a structured formalization of the scientific knowledge, where important principles like energy conservation appear prominently formalized as active formulas in cut inferences, as shown in the formal proof φ of Section 3. Indeed, reductionism in Science can generally be captured by the proof-theretical notion of cut. Consequently, a significant part of the usual scientific activity can be formally described as cut-introduction.

Cut-introduction also leads to the compression of proofs. Although the general problem of finding the shortest proofs by means of cut-introduction is undecidable [5], there are a few preliminary algorithms that introduce simple cuts [15, 10, 24], and it has been shown that some techniques of machine learning, such as decision tree learning, can be seen as cut-introduction techniques from a proof-theoretical point of view [23]. Therefore, a potential benefit of using proofs to formalize Physics is the possibility of applying cut-introduction techniques in order to automatically discover useful physical concepts. However, it must be noted that current cut-introduction techniques are still not sophisticated enough to be applied to formalized proofs of Physics.

4.2 Cut-Elimination

The problem of eliminating cuts from proofs is much easier than the problem of introducing cuts and has been much more deeply investigated [11, 4]. By using cut-elimination algorithms, it might be possible to automatically transform a solution that uses a derived principle (i.e. a cut) such as energy conservation (e.g. Solution 1 in Section 3) into a solution that uses only the basic laws of a theory (e.g. Solution 2 in Section 3). This is advantageous in certain cases, for in a cut-free proof it is easy, via Gentzen's Midsequent Theorem [11] or more general Herbrand sequent extraction algorithms [16], to extract a Herbrand disjunction [12] that contains instances of the quantified variables of the problem. For example, in the cut-free proof of Solution 2, the existentially quantified variable for the time when the object reaches height zero is instantiated by a

ground term that denotes exactly when this happens. In the proof with cuts that formalizes Solution 1, on the other hand, it is instantiated by an eigenvariable, and hence the time when the object reaches height zero is not known. Therefore, cut-elimination could in principle be used as an algorithm that instantiates the variables of a problem that were left unsolved. However, even though this idea has been succesfully used in mathematics [14], the challenge in the case of Physics is to make cut-elimination algorithms work with high-level calculi such as $\mathbf{LK^P}$.

4.3 Logic Programming

The idea of formalizing a problem as a theorem and in such a way that its solution is in the instances used for the quantified variables in the proof is the fundamental principle behind the logic programming paradigm of computation, of which Prolog [19] is the most prominent language. Therefore, the proof-theoretical approach to the formalization of Physics brings a new paradigm of computation that might be the subject of studies from the point of view of Physics itself, as imperative computation, which is modeled by Turing machines, has been.

4.4 Functional Programming and the Curry-Howard Isomorphism

The Curry-Howard isomorphism [8] states that there is a correspondence between proofs of the implicational fragment of intuitionistic logic and lambda terms. A proof is essentially a functional program. Cut-elimination corresponds to beta-reduction, which is the execution of the program. Cut-introduction corresponds to structuring of the program and possibly to code reuse. By extrapolating this isomorphism, theories of Physics formalized as collections of proofs can be seen as collections of programs. This kind of computation, which is implicit in the formalization of Physics, is yet another link between Physics and computation that might be the target of future work.

4.5 Instrumentalism: Truth versus Usefulness

From an instrumental viewpoint, "the most important function of a theory is not to organize or assert statements that are true or false but to furnish material principles of inference that may be used in inferring one set of facts from another". This idea is supported by the proof-theoretical approach described here, as shown in the formal proof φ_2 in Section 3, where Newton's law of motion was not merely a statement; it was used as a principle of inference, in the form of a definition inference rule. Instrumentalism also judges theories by how useful they are in solving problems. The proof-theoretical approach naturally embraces this criterium of usefulness, since solutions to problems can be formalized as proofs, as shown by φ_1 and φ_2. And as the commitment to truth is not given up, it conciliates two opposing positions in the philosophy of science.

4.6 The Evolution of Theories

Another philosophical viewpoint that opposes axiomatization is that of Weltanschauungen analyses, according to which science ought to be viewed as "an ongoing social enterprise [and] epistemic understanding of scientific theories could only be had by seeing the dynamics of theory development" [20]. "An ultimately meaningful answer to the question 'what is a scientific theory?' cannot be given in terms of the kinds of concepts considered earlier [axiomatization and semantics]. An adequate and complete answer can be given only in terms of an explicit and detailed consideration of both the producers and consumers of the theory." [21]. Proof theory conciliates formalization with this philosophical viewpoint in the following way: by defining scientific theories as collections of proofs, they can evolve by the addition of new proofs, and Kuhn's major paradigm shifts can be seen as major proof transformations (e.g. cut-elimination, cut-introduction and addition of new definitions).

4.7 Algorithmic Information Theory

Algorithmic Information Theory (AIT) sees scientific theories as data compressed in the form of programs. It provides a very simple, elegant and general criterium to judge and compare theories: the smaller the program, the better the theory. However, the proponents of AIT are currently making an unfortunate choice of how to encode their data, and this causes the limitations of their approach. Diagrams in [7] suggest that theories/programs should correspond to axioms, and the execution of the program by a computer, regarded as an automated theorem prover, should output empirical data in the form of theorems. Therefore, they essentially adhere to the traditional Hilbert-style axiomatization approach, and hence they suffer the same drawbacks, which are nicely explained from a computational point of view in [7]. Two of them can be summarized as follows: in current AIT, computation time is ignored, because only program size matters; and the theory/program's language is static, implying that new concepts can never emerge and the theory can never evolve.

Fortunately, proof theory can rescue AIT as well, and even provide further insight. The idea is that AIT's principle of program-size minimality should be applied not to axioms (artificially encoded as programs) but rather to the proofs that formalize a scientific theory. From a conceptual point of view, it is clear that proof theory and AIT fit perfectly together, because proofs are already programs according to the (extrapolated) Curry-Howard isomorphism. The computation time that was previously ignored now appears explicitly as the length of proofs [17] and theories can naturally evolve by the addition and transformation of proofs in the collection, with new concepts emerging by the introduction of cuts and definition inferences.

Another indication that AIT and proof theory fit well together is the natural relation between cut-introduction and kolmogorov complexity [13]. The Kolmogorov complexity $C(\psi)$ of a proof ψ can be defined as the size of the shortest proof ψ' that can be obtained by cut-introduction from ψ (and, conversely, such that ψ can be reconstructed from ψ' by cut-elimination).

5 Conclusions

"It is unheard of to find a substantive example of a theory actually worked out as a logical calculus in the writings of most philosophers of science. Much handwaving is indulged in to demonstrate that this [...] is simple in principle and only a matter of tedious detail, but concrete evidence is seldom given." [21]. In Section 3, an example of problem solution in Newtonian mechanics has been successfully worked out in a sequent calculus extended with sophisticated simplification, integration and definition rules, inspired by recent advances in Proof Theory. These extensions are the key to the small size and significantly reduced amount of tedious detail in the obtained formal proofs.

Section 4 showed that this proof-theoretical approach successfully conciliates and unifies various philosophical views of Science, such as formalism, instrumentalism and Weltanschauungen analyses. The essence of these achievements lies in seeing scientific theories not just as collections of facts, as assumed by traditional axiomatization. Scientific theories ought to be formalized as collections of proofs. The structure of scientific knowledge can be nicely formalized with cuts, and much of the scientific activity can be formally described as proof generation or proof transformation. The task of organizing knowledge, for example, can be formally described as cut-introduction.

Moreover, cut-introduction potentially compresses proofs, which can also be seen as programs according to the (extrapolated) Curry-Howard isomorphism. This indicates a tight relation between cut-introduction and Kolmogorov complexity, and thus the use of proofs clarifies, conceptually improves and solves some limitations of the ideas of algorithmic information theory with respect to the formalization of Science.

The proof-theoretical approach advocated here should be seen not as competing against existing axiomatic and semantical approaches, but rather as complementing them by enriching their formalizations with structure.

Future work should concentrate on applying these proof-theoretical ideas to complement the formalization of more interesting physical theories, such as Relativity (e.g. [2]) and Quantum Mechanics (e.g. [1]); on improving proof assistants and proof-theoretical techniques, such as cut-elimination and cut-introduction, in order to support logical calculi at least as sophisticated as $\mathbf{LK^P}$; and on investigating the new links between Physics and Computation that are opened by Proof Theory.

References

1. Diederik Aerts. Quantum mechanics: Structures, axioms and paradoxes. In *Quantum Mechanics and the Nature of Reality*, pages 141–205. Kluwer Academic, 1999.
2. Hajnal Andréka, Judit X. Madarász, István Németi, and Gergely Székely. Axiomatizing relativistic dynamics without conservation postulates. *Studia Logica*, 89(2):163–186, 2008.
3. Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Cut-Elimination: Experiments with CERES. In Franz Baader and Andrei

Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR) 2004*, volume 3452 of *Lecture Notes in Computer Science*, pages 481–495. Springer, 2005.
4. Matthias Baaz and Alexander Leitsch. Towards a clausal analysis of cut-elimination. *Journal of Symbolic Computation*, 41(3–4):381–410, 2006.
5. Matthias Baaz and Richard Zach. Algorithmic structuring of cut-free proofs. In *CSL '92: Selected Papers from the Workshop on Computer Science Logic*, pages 29–42, London, UK, 1993. Springer-Verlag.
6. Paul Brauner, Clement Houtmann, and Claude Kirchner. Principles of Superdeduction. In *Twenty-Second Annual IEEE Symposium on Logic in Computer Science (LiCS)*, 2007.
7. G. Chaitin. The intelligibility of the universe and the notions of simplicity, complexity and irreducibility, 2002.
8. Philippe De Groote, editor. *The Curry-Howard Isomorphism*. 1995.
9. Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. Rapport de Recherche 3400, Institut National de Recherche en Informatique et en Automatique, April 1998.
10. Marcelo Finger and Dov M. Gabbay. Equal rights for the cut: Computable non-analytic cuts in cut-based proofs. *Logic Journal of the IGPL*, 15(5-6):553–575, 2007.
11. G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1934–1935.
12. J. Herbrand. *Recherches sur la Theorie de la Demonstration*. PhD thesis, University of Paris, 1930.
13. Stefan Hetzl. Proof Fragments, Cut-Elimination and Cut-Introduction. manuscript.
14. Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. Herbrand sequent extraction. In *Proceedings of the Conferences on Intelligent Computer Mathematics*, number 5144 in LNAI, 2008.
15. Dale Miller and Vivek Nigam. Incorporating tables into proofs. In J. Duparc and T.A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646, pages 466–480. Springer, 2007.
16. Bruno Woltzenlogel Paleo. *Herbrand Sequent Extraction*. VDM-Verlag, Saarbruecken, Germany, 1 2008.
17. Pavel Pudlak. *The Length of Proofs*, chapter The Length of Proofs, pages 548–637. Elsevier Science B.V., 1998.
18. Adonai S. Sant'Anna. An axiomatic framework for classical particle mechanics without space-time. 0000.
19. Ehud Shapiro and Leon Sterling. *The Art of Prolog: Advanced Programming Techniques*. The MIT Press, April 1994.
20. Frederick Suppe. *The Structure of Scientific Theories*. University of Illinois Press, 2 edition, 1977.
21. Patrick Suppes. What is a scientific theory? *Philosophy of Science Today*, pages 55–67, 1967.
22. A.S. Wightman. Hilbert's sixth problem: Mathematical treatment of the axioms of physics. In *Procceding of Symposia in Pure Mathematics*, volume 28, 1976.
23. Bruno Woltzenlogel Paleo. *A General Analysis of Cut-Elimination by CERes*. PhD thesis, Vienna University of Technology, 2009.
24. Bruno Woltzenlogel Paleo. Atomic cut introduction by resolution: Proof structuring and compression. submitted, 2010.

Bertlmann's Chocolate Balls and Quantum Type Cryptography

Karl Svozil

Institute for Theoretical Physics, University of Technology Vienna,
Wiedner Hauptstrasse 8-10/136, 1040 Vienna, Austria
emailsvozil@tuwien.ac.at
http://tph.tuwien.ac.at/ svozil

Abstract. Some quantum cryptographic protocols can be implemented with specially prepared chocolate balls, others protected by value indefiniteness cannot. Similarities and differences of cryptography with quanta and chocolate are discussed. Motivated by these considerations it is proposed to certify quantum random number generators and quantum cryptographic protocols by value indefiniteness. This feature, which derives itself from Bell- and Kochen-Specker type arguments, is only present in systems with three or more mutually exclusive outcomes.

Keywords: Quantum Information, Quantum Cryptography, Singlet States, Entanglement, Quantum Nonlocality

1 Quantum Resources for Cryptography

Quantum cryptography[1] uses quantum resources to encode plain symbols forming some message. Thereby, the security of the code against cryptanalytic attacks to recover that message rests upon the validity of physics, giving new and direct meaning to Landauer's dictum [6] "information is physical."

What exactly are those quantum resources on which quantum cryptography is based upon? Consider, for a start, the following qualities of quantized systems:

(i) randomness of certain individual events, such as the occurrence of certain measurement outcomes for states which are in a superposition of eigenstates associated with eigenvalues corresponding to these outcomes;
(ii) complementarity, as proposed by Pauli, Heisenberg and Bohr;
(iii) value indefiniteness, as attested by Bell, Kochen & Specker and others (often, this property is referred to as "contextuality");
(iv) interference and quantum parallelism, allowing the co-representation of classically contradicting states of information by a coherent superposition thereof;

[1] In view of the many superb presentations of quantum cryptography — to name but a few, see Refs. [1, 2] and [3, Chapter 6] (or, alternatively, [4, Section 6.2]), as well as [5, Section 12.6]; apologies to other authors for this incomplete, subjective collection — we refrain from any extensive introduction.

(v) entanglement of two or more particles, as pointed out by Schrödinger, such that their state cannot be represented as the product of states of the isolated, individual quanta, but is rather defined by the *joint* or *relative* properties of the quanta involved.

The first quantum cryptographic protocols, such as the ones by Wiesner [7] and Bennett & Brassard [8, 9], just require complementarity and random individual outcomes. This might be perceived ambivalently as and advantage — by being based upon only these two features — yet also as a disadvantage, since they are not "protected" by Bell- or Kochen-Specker type value indefiniteness.

This article addresses two issues: a critical re-evaluation of quantum cryptographic protocols in view of quantum value indefiniteness; as well as suggestions to improve them to assure the best possible protection "our" [10, p. 866] present quantum theory can afford. In doing so, a toy model will be introduced which implements complementarity but still is value definite. Then it will be exemplified how to do perform "quasi-classical" quantum-like cryptography with these models. Finally, methods will be discussed which go beyond the quasi-classical realm.

Even nowadays it is seldom acknowledged that, when it comes to value definiteness, there definitely *is* a difference between two- and three-dimensional Hilbert space. This difference can probably be best explained in terms of (conjugate) bases: whereas different basis in two-dimensional Hilbert space are disjoint and separated (they merely share the trivial origin), from three dimensions onwards, they may share common elements. It is this inter-connectedness of bases and "frames" which supports both Gleason's and the Kochen-Specker theorem. This can, for instance, be used in derivations of the latter one in three dimensions, which effectively amount to a succession of rotations of bases along one of their elements (the original Kochen-Specker [11] proof uses 117 interlinked bases), thereby creating new rotated bases, until the original base is reached. Note that certain (even dense [12]) "dilutions" of bases break up the possibility to interconnect, thus allowing value definiteness.

The importance of these arguments for physics is this: since in quantum mechanics the dimension of Hilbert space is determined by the number of mutually exclusive outcomes, a *necessary* condition for a quantum system to be protected by value indefiniteness thus is that the associated quantum system has *at least three* mutually exclusive outcomes; two outcomes are insufficient for this purpose. Of course, one could argue that systems with two outcomes are still protected by complementarity.

2 Realizations of Quantum Cryptographic Protocols

Let us, for the sake of demonstration, discuss a concrete "toy" system which features complementarity but (not) value (in)definiteness. It is based on the partitions of a set. Suppose the set is given by $S = \{1, 2, 3, 4\}$, and consider two of its equipartitions $A = \{\{1, 2\}, \{3, 4\}\}$ and $B = \{\{1, 3\}, \{2, 4\}\}$, as well as the

usual set theoretic operations (intersection, union and complement) and the subset relation among the elements of these two partitions. Then A and B generate two Boolean algebras $L_A = \{\emptyset, \{1,2\}, \{3,4\}, S\}$ and $L_B = \{\emptyset, \{1,3\}, \{2,4\}, S\}$ which are equivalent to 2^2; with two atoms $a_1 = \{1,2\}$ & $a_2 = \{3,4\}$, as well as $b_1 = \{1,3\}$ & $b_2 = \{2,4\}$ per algebra, respectively. Then, the partition logic $L_A \oplus L_B = L_{A,B} = \langle \{L_A, L_B\}, \cap, \cup, ', \subset \rangle$ is obtained as a pasting construction from L_A and L_B: only elements contribute which are in L_A, or in L_B, or in both $L_A \cap L_B$ of them (the atoms of this algebra being the elements $a_1, \ldots, b_2$), and all common elements — in this case only the smallest and greatest elements $\emptyset$ and S — are identified. $L_{A,B}$ "inherits" the operations and relations of its subalgebras (also called *blocks* or *contexts*) L_A and L_B. This pasting construction yields a nondistributive and thus nonboolean, orthocomplemented propositional structure. Nondistributivity can quite easily be proven, as $a_1 \wedge (b_1 \vee b_2) \neq (a_1 \wedge b_1) \vee (a_1 \wedge b_2)$, since $b_1 \vee b_2 = S$, whereas $a_1 \wedge b_1 = a_1 \wedge b_2 = \emptyset$. Note that, although $a_1, \ldots, b_2$ are compositions of elements of S, not all elements of the power set $2^S \equiv 2^4$ of S, such as $\{1\}$ or $\{1,2,3\}$, are contained in $L_{A,B}$.

Figure 1(a) depicts a Greechie (orthogonality) diagram of $L_{A,B}$, which represents elements in a Boolean algebra as single smooth curves; in this case there are just two atoms (least elements above $\emptyset$) per subalgebra; and both subalgebras are not interconnected.

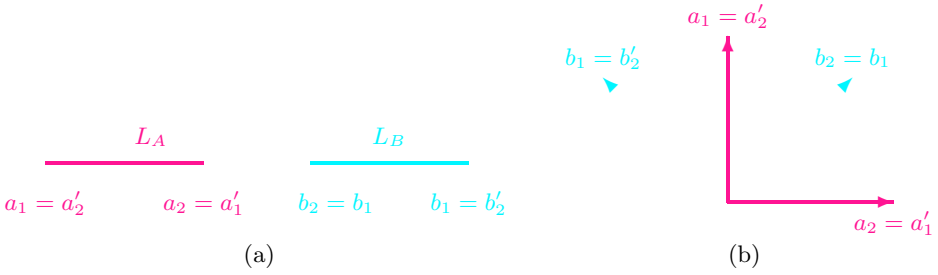


Fig. 1. (Color online) (a) Greechie diagram of $L_{A,B}$, consisting of two separate Boolean subalgebras L_A and L_B; (b) two-dimensional configuration of spin-$\frac{1}{2}$ state measurements along two noncollinear directions. As there are only two mutually exclusive outcomes, the dimension of the Hilbert space is two.

Several realizations of this partition logic exist; among them

(i) the propositional structure [13, 14] of spin state measurements of a spin-$\frac{1}{2}$ particle along two noncollinear directions, or of the linear polarization of a photon along two nonorthogonal, noncollinear directions. A two-dimensional Hilbert space representation of this configuration is depicted in Figure 1(b). Thereby, the choice of the measurement direction decides which one of the two complementary spin state observables is measured;

(ii) generalized urn models [15, 16]; in particular ones with black balls painted with two symbols having two possible values (say, "0 and "1) in two colors

(say, "red" and "green"), resulting in four types of balls — more explicitly, carrying all variation of the symbols 00, 01, 10, as well as 11 — many copies of which are randomly distributed in an urn. Suppose the experimenter looks at them with one of two differently colored eyeglasses, each one ideally matching the colors of only one of the symbols, such that only light in this wave length passes through. Thereby, the choice of the color decides which one of the two complementary observables associated with "red" and "green" is measured. Propositions refers to the possible ball types drawn from the urn, given the information printed in the chosen color.

(iii) initial state identification problem for deterministic finite (Moore or Mealy) automata in an unknown initial state [17, 18]; in particular ones $\langle S, I, O, \delta, \lambda\rangle$ with four internal states $S = \{1, 2, 3, 4\}$, two input and two output states $I = O = \{0, 1\}$, an "irreversible" (all-to-one) transition function $\delta(s, i) = 1$ for all $s \in S$, $i \in I$, and an output function "modelling" the state partitions by $\lambda(1, 0) = \lambda(2, 0) = 0$, $\lambda(3, 0) = \lambda(4, 0) = 1$, $\lambda(1, 1) = \lambda(3, 1) = 0$, $\lambda(2, 1) = \lambda(4, 1) = 1$. Thereby, the choice of the input symbol decides which one of the two complementary observables is measured.

Let us, for the moment, consider generalized urn models, because they allow a "pleasant" representation as chocolate balls coated in black foils and painted with color symbols. With the four types of chocolate balls 00, 01, 10, and 11 drawn from an urn it is possible to execute the 1984 Bennett-Brassard (BB84) protocol [8, 9] and "generate" a secret key shared by two parties [19]. Formally, this reflects (i) the random draw of balls from an urn, as well as (ii) the complementarity modeled *via* the color painting and the colored eyeglasses. It also reflects the possibility to embed this model into a bigger Boolean (and thus classical) algebra 2^4 by "taking off the eyeglasses" and looking at both symbols of those four balls types simultaneously. The atoms of this Boolean algebra are just the ball types, associated with the four cases 00, 01, 10, and 11. The possibility of a classical embedding is also reflected in a "sufficient" number (i.e., by a separating, full set) of two-valued, dispersionless states $P(a_1) + P(a_2) = P(b_1) + P(b_2) = 1$, with $P(x) \in \{0, 1\}$. These two-valued states can also be interpreted as logical truth assignments, irrespective of whether or not the observables have been (co-)measured.

The possibility to ascribe certain "ontic states" interpretable as observer-independent "omniscient elements of physical reality" (in the sense of Einstein, Podolsky and Rosen [20, p. 777], a paper which amazingly contains not a single reference) even for complementarity observables may raise some skepticism or even outright rejection, since that is not how quantum mechanics is known to perform "at its most mind-boggling mode." Indeed, so far, the rant presented merely attempted to convince the reader that one can have complementarity *as well as* value definiteness; i.e., complementarity is not sufficient for value indefiniteness in the sense of the Bell- and Kochen-Specker argument.

Unfortunately, the two-dimensionality of the associated Hilbert space is also a feature plaguing present random number generators based on beam splitters [21–24]. In this respect, most of the present random number generators using beam splitters are protected only by the randomness of single outcomes as well as by complementarity, but are not by certified value indefiniteness, as guaranteed by quantum theory in its standard form [25]. Their methodology should also be improved by the methods discussed below.

3 Supporting Cryptography with Value Indefiniteness

Alas, quantum mechanics is more resourceful and mind-boggling than that, as it does not permit any two-valued states which may be ontologically interpretable as elements of physical reality. So we have to go further, reminding ourselves that value indefiniteness comes about only for Hilbert spaces of dimensions three and higher. There are several ways of doing this. The following options will be discussed:

(i) the known protocols can be generalized to three or more outcomes [26];
(ii) entangled pairs of particles [27] associated with statistical value indefiniteness may be considered;
(iii) full, nonprobabilistic value indefiniteness may be attempted, at least counterfactually.

3.1 Generalizations to three and more outcomes

In constructing quantum random number generators *via* beam splitters which ultimately are used in cryptographic setups, it is important (i) to have full control of the particle source, and (ii) to use beam splitters with three or more output ports, associated with three- or higher-dimensional Hilbert spaces. Thereby, it is *not sufficient* to compose a multiport beam splitter by a succession of phase shifters and beam splitters with two output ports [28, 29], based on elementary decompositions of the unitary group [30].

Dichotomic sequences could be obtained from sequences containing more than two symbols by discarding all other symbols from that sequence [31], or by identifying the additional symbols with one (or both) of the two symbols. For standard normalization procedures and their issues, the reader is referred to Refs. [32–37].

One concrete realization would be a spin-$\frac{3}{2}$ particle. Suppose it is prepared in one of its four spin states, say the one associated with angular momentum $+\frac{3}{2}\hbar$ in some arbitrary but definite direction; e.g., by a Stern-Gerlach device. Then, its spin state is again measured along a perpendicular direction; e.g., by another, differently oriented, Stern-Gerlach device. Two of the output ports, say the ones corresponding to positive angular momentum $+\frac{3}{2}\hbar$ and $+\frac{1}{2}\hbar$, are identified with the symbol "0," the other two ports with the symbol "1." In that way, a random sequence is obtained from quantum coin tosses which can be ensured to operate

under the conditions of value indefiniteness in the sense of the Kochen-Specker theorem. Of course, this protocol can also be used to generate random sequences containing four symbols (one symbol per detector).

With respect to the use of beam splitters, the reader is kindly reminded of another issue related to the fact that beam splitters are *reversible* devices capable of only translating an incoming signal into an outgoing signal in a *one-to-one* manner. The "nondestructive" action of a beam splitter could also be demonstrated by "reconstructing" the original signal through a "reversed" identical beam splitter in a Mach-Zehnder interferometer [38]. In this sense, the signal leaving the output ports of a beam splitter is "as good" for cryptographic purposes as the one entering the device. This fact relegates considerations of the quality of quantum randomness to the quality of the source. Every care should thus be taken in preparing the source to assure that the state entering the input port (i) either is pure and could subsequently be used for measurements corresponding to conjugate bases, (ii) or is maximally mixed, resulting in a representation of its state in finite dimensions proportional to the unit matrix.

3.2 Configurations with statistical value indefiniteness

Protocols like the Ekert protocol [27] utilize two entangled two-state particles for a generation of a random key shared by two parties. The particular Einstein-Podolsky-Rosen configuration [20] and the singlet Bell state communicated among the parties guarantee stronger-than-classical correlations of their sequences, resulting in a violation of Bell-type inequalities obeyed by classical probabilities.

Although criticized [39] on the grounds that the Ekert protocol in certain cryptanalytic aspects is equivalent to existing ones (see Ref. [40] for a reconciliation), it offers additional security in the light of quantum value indefiniteness, as it suggests to probe the nonclassical parts of quantum statistics. This can best be understood in terms of the impossibility to generate co-existing tables of all — even the counterfactually possible — measurement outcomes of the quantum observables used [41]. This, of course, can only happen for the four-dimensional Hilbert space configuration proposed by Ekert, and not for effectively two-dimensional ones of previous proposals. As a result, the Eckert protocol cannot be performed with chocolate balls. Formally, this is due to the nonexistence of two-valued states in four-dimensional Hilbert space.

Suppose one would nevertheless attempt to "mimic" the Ekert protocol with a classical "singlet" state which uses compositions of two balls of the form $\mathbf{00}$ — $\mathbf{11}$ / $\mathbf{01}$ — $\mathbf{10}$ / $\mathbf{10}$ — $\mathbf{01}$ / $\mathbf{11}$ — $\mathbf{00}$, with strictly different (alternatively strictly identical) particle types. The resulting probabilities and expectations would obey the classical Clauser-Horne-Shimony-Holt bounds [42]. This is due to the fact that generalized urn models have quasi-classical probability distributions which can be represented as convex combinations of the full set of separable two-valued states on their observables.

3.3 Nonprobabilistic value indefiniteness

In an attempt to fully utilize quantum value indefiniteness, we propose a generalization of the BB84 protocol on a propositional structure which does not allow any two-valued state. In principle, this could be any kind of finite configuration of observables in three- and higher-dimensional Hilbert space; in particular ones which have been proposed for a proof of the Kochen-Specker theorem.

For the sake of a concrete example, we shall consider the tightly interlinked collection of observables in four-dimensional Hilbert space presented by Cabello, Estebaranz and García-Alcaine [43, 44], which is depicted in Figure 2. Instead of two measurement bases of two-dimensional Hilbert space used in the BB84 protocol, nine such bases of four-dimensional Hilbert space, corresponding to the nine smooth (unbroken) orthogonal curves in Fig. 2 are used. In what follows, it is assumed that any kind of random decision has been prepared according to the protocol for generating random sequences sketched above.

(i) In the first step, "Alice" randomly picks an arbitrary basis from the nine available ones, and sends a random state to "Bob."
(ii) In the second step, Bob independently from Alice, picks another basis at random, and measures the particle received from Alice.
(iii) In the third step, Alice and Bob compare their bases over a public channel, and keep only those events which were recorded either in a common basis, or in an observable interlinking two different bases.
(iv) Both then exchange some of the remaining matching outcomes over a public channel to assure that nobody has attended their quantum channel.
(v) Bob and Alice encode the four outcomes by four or less different symbols. As a result, Bob and Alice share a common random key certified by quantum value indefiniteness.

The advantage of this protocol resides in the fact that is does not allow its realization by any partition of a set, or any kind of colored chocolate balls. Because if it did, any such coloring could be used to generate "classical" two-valued states, which in turn may be used towards a classical re-interpretation of the quantum observables; an option ruled out by the Kochen-Specker theorem.

Readers not totally convinced at this point might, for the sake of demonstration, consider a generalized urn model with nine colors, associated with the nine bases in Figure 2. Suppose further that there is a uniform set of symbols, say $\{0, 1, 2, 3\}$ for all four colors. If all varieties (permutations) contribute, the number of different types of balls should be 4^9. Note, however, that every interlinked color must have *identical* (or at least unique "partner") symbols in the interlinking colors; a condition which cannot be satisfied "globally" for all the interlinks in Figure 2.

A simplified version of the protocol, which is based on a subdiagram of Figure 2, contains only three contexts, which are closely interlinked. The structure of observables is depicted in Figure 3(a). The vectors represent observables in four-dimensional Hilbert space in their usual interpretation as projectors generating the one-dimensional subspaces spanned by them. In addition to this quantum

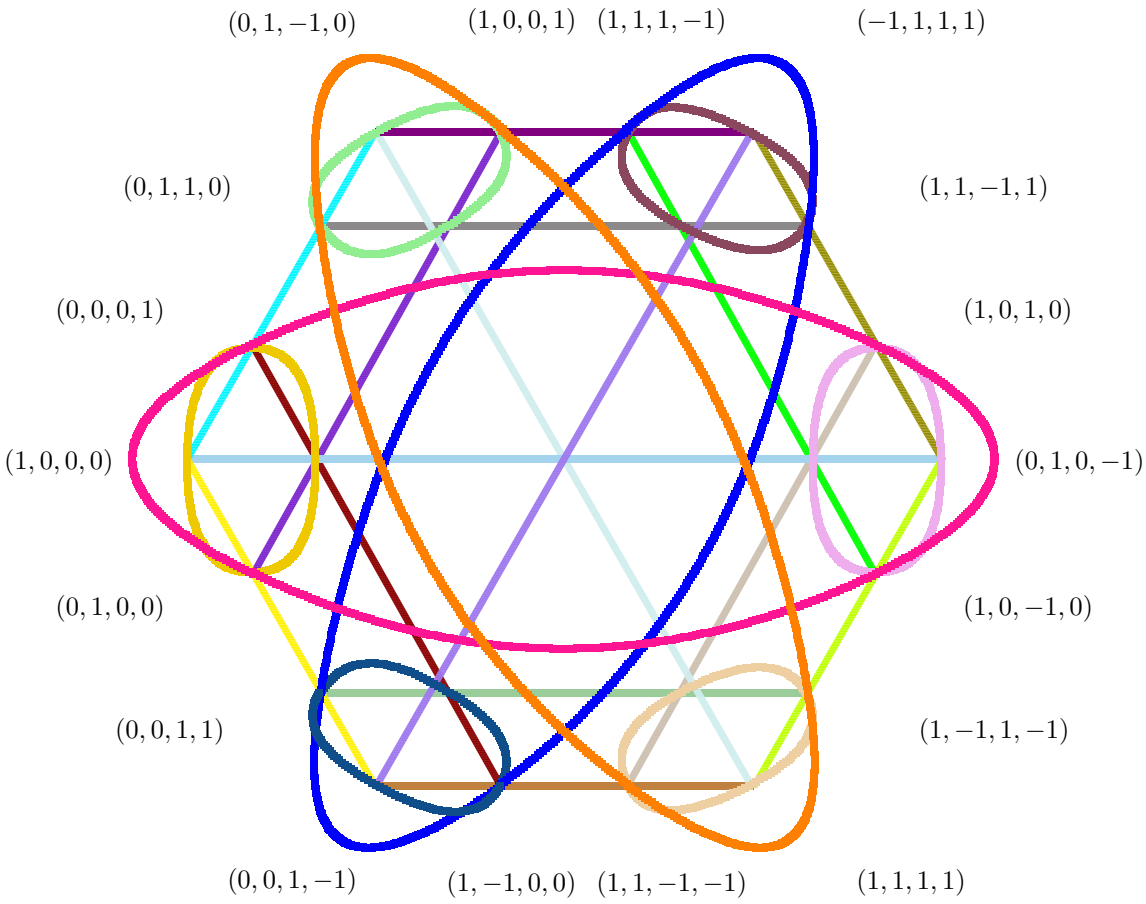


Fig. 2. (Color online) Greechie orthogonality diagram of a "short" proof [43, 44] of the Kochen-Specker theorem in four dimensions containing 24 propositions in 24 tightly interlinked contexts [45]. The graph cannot be colored by the two colors red (associated with truth) and green (associated with falsity) such that every context contains exactly one red and three green points. For the sake of a proof, consider just the six outer lines and the three outer ellipses. Then in a table containing the points of the contexts as columns and the enumeration of contexts as rows, every red point occurs in exactly *two* contexts, and there should be an *even* number of red points. On the other hand, there are nine contexts involved; thus by the rules it follows that there should be an *odd* number (nine) of red points in this table (exactly one per context).

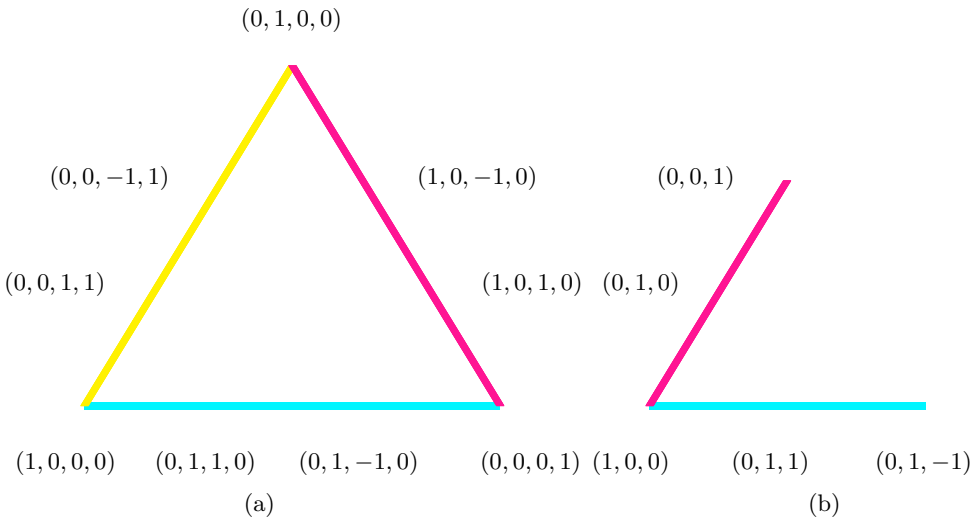


Fig. 3. (Color online) Subdiagrams of Figure 2 allowing (value definite) chocolate ball realizations.

mechanical representation, and in contrast to the Kochen-Specker configuration in Figure 2, this global collection of observables still allows for value definiteness, as there are "enough" two valued states permitting the formation of a partition logic and thus a chocolate ball realization; e.g.,

$$\{\{\{1,2\},\{3,4,5,6,7\},\{8,9,10,11,12\},\{13,14\}\},$$
$$\{\{1,4,5,9,10\},\{2,6,7,11,12\},\{3,8\},\{13,14\}\},$$
$$\{\{1,2\},\{3,8\},\{4,6,9,11,13\},\{5,7,10,12,14\}\}\}.$$

The three partitions of the set $\{1, 2, \ldots, 14\}$ have been obtained by indexing the atoms in terms of all the nonvanishing two-valued states on them [18, 46], as depicted in Figure 4. They can be straightforwardly applied for a chocolate ball configuration with three colors (say green, red and blue) and four symbols (say 0, 1, 2, and 3). The 14 ball types corresponding to the 14 different two-valued measures are as follows: **000** , **010** , **121** , **102** , **103** , **112** , **113** , **221** , **202** , **203** , **212** , **213** , **332** , and **333** .

Figure 3(b) contains a three-dimensional subconfiguration with two complementary contexts interlinked in a single observable. It again has a value definite representation in terms of partitions of a set, and thus again a chocolate ball realization with three symbols in two colors; e.g., **00** , **11** , **12** , **21** , and **22** .

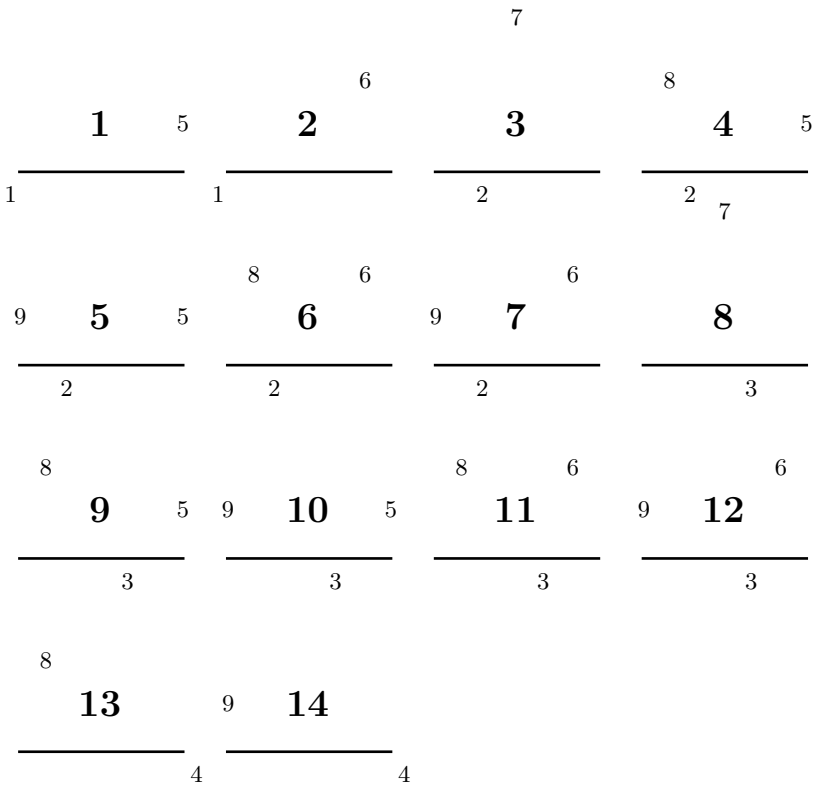


Fig. 4. Two-valued states interpretable as global truth functions of the observables depicted in Figure 3(a). Encircled numbers count the states, smaller numbers label the observables.

4 Noncommutative Cryptography Which cannot be Realized Quantum Mechanically

Quantum mechanics does not allow a "triangular" structure of observables similar to the one depicted in Fig. 3 with *three* instead of four atoms per block (context), since no geometric configuration of tripods exist in three-dimensional vector space which would satisfy this scheme. (For a different propositional structure not satisfiable by quantum mechanics, see Specker's programmatic article [47] from 1960.) It contains six atoms $1, \ldots, 6$ in the blocks 1–2–3, 3–4–5, 5–6–1. In order to obtain a partition logic on which the chocolate ball model can be based, the four two-valued states are enumerated and depicted in Figure 5.

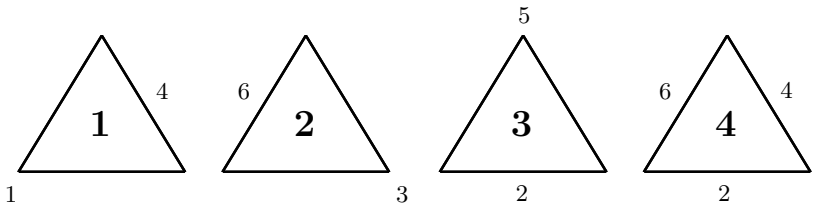


Fig. 5. Two-valued states on triangular propositional structure with three atoms per context or block.

The associated partition logic is given by

$$\{\{\{1\},\{2\},\{3,4\}\},\\ \{\{1,4\},\{2\},\{3\}\},\\ \{\{1\},\{2,4\},\{3\}\}\}.$$

Every one of the three partitions of the set $\{1, \ldots, 4\}$ of ball types labelled by 1 through 4 corresponds to a color; and there are three symbols per colors. For the first (second/third) partition, the propositions associated with these protocols are:

- "when seen through light of the first (second/third) color (e.g., pink/light blue/yellow), symbol "0" means ball type number 1 (2/3);"
- "when seen through light of the first (second/third) color (e.g., pink/light blue/yellow), symbol "1" means ball type number 3 or 4 (1 or 4/2 or 4);"
- "when seen through light of the first (second/third) color (e.g., pink/light blue/yellow), symbol "2" means ball type number 2 (3/1)."

More explicitly, there are four ball types of the form 012 , 201 , 120 , and

111 . The resulting propositional structure is depicted in Fig. 6. With respect to realizability, cryptographic protocols — such as the one sketched above — based on this structure are "stranger than quantum mechanical" ones.

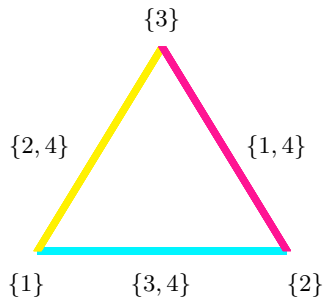


Fig. 6. (Color online) Propositional structure allowing (value definite) chocolate ball realizations with three atoms per context or block which does not allow a quantum analog.

5 Summary and Discussion

It has been argued that value indefiniteness should be used as a quantum resource against cryptanalytic attacks, as complementarity may not be a sufficient resource for the type of "objective" security envisaged by quantum cryptography. A necessary condition for this quantum resource is the presence of at least three mutually exclusive outcomes.

It may be objected that quantum complementarity suffices as resource against cryptanalytic attacks, and thus the original BB84 protocol needs not be amended. To this criticism I respond with a performance of the original BB84 protocols with chocolate balls [19]; or more formally, by stating that configurations with just two outcomes leave open the possibility of a quasi-classical explanation, as they cannot rule out the existence of sufficiently many two-valued states in order to construct homeomorphisms, i.e., structure-preserving maps between the quantum and classical observables. Thus, when it comes to fully "harvesting" the quantum, it appears prudent to utilize value indefiniteness, one of its most "mind-boggling" features encountered if one assumes the existence of nonoperational yet counterfactual observables.

Acknowledgements

The author gratefully acknowledges discussions with Cristian Calude and Josef Tkadlec, as well as the kind hospitality of the *Centre for Discrete Mathematics and Theoretical Computer Science (CDMTCS)* of the *Department of Computer Science at The University of Auckland.* This work was also supported by *The Department for International Relations* of the *Vienna University of Technology.* The pink–light blue–yellow coloring scheme is by Renate Bertlmann; communicated to the author by Reinhold Bertlmann.

References

1. N. Gisin, G. Ribordy, W. Tittel, H. Zbinden, Quantum cryptography, Review of Modern Physics 74 (2002) 145–195.
2. V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, M. Peev, The security of practical quantum key distribution, Reviews of Modern Physics 81 (2009) 1301–1350.
3. N. D. Mermin, Lecture notes on quantum computation, 2002-2008.
4. N. D. Mermin, Quantum Computer Science, Cambridge University Press, Cambridge, 2007.
5. M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000.
6. R. Landauer, Information is physical, Physics Today 44 (1991) 23–29.
7. S. Wiesner, Conjugate coding, SIGACT News 15 (1983) 78–88. Manuscript written *circa* 1970 [9, Ref. 27].
8. C. H. Bennett, G. Brassard, Quantum cryptography: Public key distribution and coin tossing, in: Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India, IEEE Computer Society Press, 1984, pp. 175–179.
9. C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, J. Smolin, Experimental quantum cryptography, Journal of Cryptology 5 (1992) 3–28.
10. M. Born, Zur Quantenmechanik der Stoßvorgänge, Zeitschrift für Physik 37 (1926) 863–867.
11. S. Kochen, E. P. Specker, The problem of hidden variables in quantum mechanics, Journal of Mathematics and Mechanics (now Indiana University Mathematics Journal) 17 (1967) 59–87. Reprinted in Ref. [48, pp. 235–263].
12. D. A. Meyer, Finite precision measurement nullifies the Kochen-Specker theorem, Physical Review Letters 83 (1999) 3751–3754.
13. G. Birkhoff, J. von Neumann, The logic of quantum mechanics, Annals of Mathematics 37 (1936) 823–843.
14. K. Svozil, Quantum Logic, Springer, Singapore, 1998.
15. R. Wright, Generalized urn models, Foundations of Physics 20 (1990) 881–903.
16. A. Dvurečenskij, S. Pulmannová, K. Svozil, Partition logics, orthoalgebras and automata, Helvetica Physica Acta 68 (1995) 407–428.
17. E. F. Moore, Gedanken-experiments on sequential machines, in: C. E. Shannon, J. McCarthy (Eds.), Automata Studies, Princeton University Press, Princeton, 1956, pp. 129–153.
18. K. Svozil, Logical equivalence between generalized urn models and finite automata, International Journal of Theoretical Physics 44 (2005) 745–754.
19. K. Svozil, Staging quantum cryptography with chocolate balls, American Journal of Physics 74 (2006) 800–803.
20. A. Einstein, B. Podolsky, N. Rosen, Can quantum-mechanical description of physical reality be considered complete?, Physical Review 47 (1935) 777–780.
21. K. Svozil, The quantum coin toss—testing microphysical undecidability, Physics Letters A 143 (1990) 433–437.
22. J. G. Rarity, M. P. C. Owens, P. R. Tapster, Quantum random-number generation and key sharing, Journal of Modern Optics 41 (1994) 2435–2444.
23. T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, A. Zeilinger, A fast and compact quantum random number generator, Review of Scientific Instruments 71 (2000) 1675–1680.

24. A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, H. Zbinden, Optical quantum random number generator, Journal of Modern Optics 47 (2000) 595–598.
25. J. von Neumann, Mathematische Grundlagen der Quantenmechanik, Springer, Berlin, 1932. English translation in Ref. [49].
26. H. Bechmann-Pasquinucci, A. Peres, Quantum cryptography with 3-state systems, Physical Review Letters 85 (2000) 3313–3316.
27. A. K. Ekert, Quantum cryptography based on Bell's theorem, Physical Review Letters 67 (1991) 661–663.
28. M. Reck, A. Zeilinger, H. J. Bernstein, P. Bertani, Experimental realization of any discrete unitary operator, Physical Review Letters 73 (1994) 58–61.
29. K. Svozil, Noncontextuality in multipartite entanglement, J. Phys. A: Math. Gen. 38 (2005) 5781–5798.
30. F. D. Murnaghan, The Unitary and Rotation Groups, Spartan Books, Washington, D.C., 1962.
31. C. Calude, I. Chiţescu, Qualitative properties of P. Martin-Löf random sequences, Unione Matematica Italiana. Bollettino. B. Serie VII 3 (1989) 229–240.
32. J. von Neumann, Various techniques used in connection with random digits, National Bureau of Standards Applied Math Series 12 (1951) 36–38. Reprinted in *John von Neumann, Collected Works, (Vol. V)*, A. H. Traub, editor, MacMillan, New York, 1963, p. 768–770.
33. P. A. Samuelson, Constructing an unbiased random sequence, Journal of the American Statistical Association 63 (1968) 1526–1527.
34. P. Elias, The efficient construction of an unbiased random sequence, Ann. Math. Statist. 43 (1972) 865–870.
35. Y. Peres, Iterating Von Neumann's procedure for extracting random bits, The Annals of Statistics 20 (1992) 590–597.
36. M. Dichtl, Bad and good ways of post-processing biased physical random numbers, in: A. Biryukov (Ed.), Fast Software Encryption. Lecture Notes in Computer Science Volume 4593/2007, Springer, Berlin and Heidelberg, 2007, pp. 137–152. 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers.
37. P. Lacharme, Post-processing functions for a biased physical random number generator, in: K. Nyberg (Ed.), Fast Software Encryption. Lecture Notes in Computer Science Volume 5086/2008, Springer, Berlin and Heidelberg, 2008, pp. 334–342. 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers.
38. D. M. Greenberger, M. A. Horne, A. Zeilinger, Multiparticle interferometry and the superposition principle, Physics Today 46 (1993) 22–29.
39. C. H. Bennett, G. Brassard, D. N. Mermin, Quantum cryptography without Bell's theorem, Physical Review Letters 68 (1992a) 557–559.
40. C. H. Bennett, G. Brassard, A. K. Ekert, Quantum cryptography, Scientific American 267 (1992b) 50–57.
41. A. Peres, Unperformed experiments have no results, American Journal of Physics 46 (1978) 745–747.
42. J. F. Clauser, M. A. Horne, A. Shimony, R. A. Holt, Proposed experiment to test local hidden-variable theories, Physical Review Letters 23 (1969) 880–884.
43. A. Cabello, J. M. Estebaranz, G. García-Alcaine, Bell-Kochen-Specker theorem: A proof with 18 vectors, Physics Letters A 212 (1996) 183–187.
44. A. Cabello, Experimentally testable state-independent quantum contextuality, Physical Review Letters 101 (2008) 210401.

45. J. Tkadlec, 2009. Private communication.
46. K. Svozil, Contexts in quantum, classical and partition logic, in: K. Engesser, D. M. Gabbay, D. Lehmann (Eds.), Handbook of Quantum Logic and Quantum Structures, Elsevier, Amsterdam, 2009, pp. 551–586.
47. E. Specker, Die Logik nicht gleichzeitig entscheidbarer Aussagen, Dialectica 14 (1960) 239–246. Reprinted in Ref. [48, pp. 175–182]; English translation: *The logic of propositions which are not simultaneously decidable*, Reprinted in Ref. [50, pp. 135-140].
48. E. Specker, Selecta, Birkhäuser Verlag, Basel, 1990.
49. J. von Neumann, Mathematical Foundations of Quantum Mechanics, Princeton University Press, Princeton, 1955.
50. C. A. Hooker, The Logico-Algebraic Approach to Quantum Mechanics. Volume I: Historical Evolution, Reidel, Dordrecht, 1975.

Coalgebras and non-determinism: an application to multilattices*

I. P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, and M. Ojeda-Aciego

Dpto. Matemática Aplicada. Universidad de Málaga.
{ipcabrera, ggutierrez,jmartinezd,pcordero,aciego}@uma.es

Abstract. Multilattices are a suitable generalization of lattices which enables to accommodate the formalization of non-deterministic computation; specifically, the algebraic characterization for multilattices provides a formal framework to develop tools in several fields of computer science. On the other hand, the usefulness of coalgebra theory has been increasing in the recent years, and its importance is undeniable. In this work, we define a new kind of coalgebras (the ND-coalgebras) that allows to formalize non-determinism, and show that several concepts, widely used in computer science are, indeed, ND-coalgebras. Within this formal context, we study a minimal set of properties which provides a coalgebraic definition of multilattices.

1 Introduction

The notion of multilattice was introduced by Benado [1], as an extension of the concept of lattice by means of multi-suprema (minimal upper bounds) and multi-infima (maximal lower bounds).

Although its original motivation was purely theoretical, multilattices (and relatives such as multisemilattices) have been identified in several disparate research areas: (1) in the field of automated deduction, specifically when devising a theory about implicates and implicants for certain temporal logics during the development of automated theorem provers for those logics [2]; (2) unification for logical systems, whose starting point was the existence of a most general unifier for any unifiable formula in Boolean logic: in 1999, Ghilardi [4] proved that there are no most general unifiers in intuitionistic propositional calculus but instead there is a finite set of maximal general unifiers.

The first applicable algebraic characterization is relatively recent [6], and it reflects much better the corresponding classical theory about lattices than those given previously. Since then, several works have been published about the mathematical theory of multilattices and, in general, about hyperstructures and non-deterministic structures [3]. It is convenient to state that, in the meantime, several other generalizations of the notion of lattice have been developed so far: for instance, *nearlattices*, *hyperlattices*, or *superlattices*.

* Partially supported by Spanish Science Ministry project TIN09-14562-C05-01 and Junta de Andalucía project P09-FQM-5233.

We are focusing our attention on multilattices since we believe their computational properties are better suited to the aims stated as follows: The idea underlying the algebraic study of multilattices is the development of a new theory involving *non-deterministic operators* as a framework for formalizing key notions in computer science and artificial intelligence. For instance, non-determinism has been considered considered under the combination of modal and temporal logics to be used in communication systems; new results have been recently obtained in database theory as well. A lot of effort is being put in this area, as one can still see recent works dealing with non-determinism both from the theoretical and from the practical point of view [5, 7].

This work is concerned with coalgebras as well. Rutten developed the theory of coalgebras which can be seen as a sort of dualization of universal algebra, when considered from a category-theoretical standpoint. This theory is becoming an ideal framework for formalization in diverse branches of computer science. Specifically, concepts as important as Kripke structures, labeled transition systems, various types of automata (in particular, non-deterministic automata), reactive systems, causal maps, ambient calculus, services and contracts, have a coalgebraic explanation.

Certain abstract structures can be thought of both algebraically and coalgebraically. The context and the aims of the work usually indicates which framework one should consider; for instance, when non-deterministic behavior is assumed, the coalgebraic framework is generally preferred because it appears to fit more naturally. Following this trend, we started a research line consisting in developing a coalgebraic view of several mathematical structures of interest for the handling of non-determinism, in particular, for multilattices.

A typical example of coalgebra is the non-deterministic automaton in which, in its simplest version, we have a set of states S and a transition function between states $S \to \mathcal{P}(S)$. Now, let us consider that such an automaton corresponds to an agent within a multiagent framework containing $n + 1$ agents interacting. Each agent changes its state depending on its own state and the state of the rest of the agents. Thus, the transition function between states would be of type $S^{n+1} \to \mathcal{P}(S)$. However, the agent knows its own state whereas the rest of states have to be consulted, in such a way that the transition function can be considered of type $S \to \mathcal{P}(S)^{S^n}$. As a result, the properties of the transition function can be separated into two levels: those known to the agent, and those to be consulted. Note that the transformation from $S^{n+1} \to \mathcal{P}(S)$ to $S \to \mathcal{P}(S)^{S^n}$ is just an instance of the currying process (or partial application), which transforms a function that takes a tuple of arguments in such a way that it can be called as a chain of functions each with a single argument.

Following the trend of developing a coalgebraic approach for several non-deterministic structures, we have defined a suitable class of coalgebras, the ND-coalgebras, and developed a thorough analysis of the required properties in order to achieve a convenient coalgebraic characterization of multilattices which complements the algebraic one given in [2].

The class of ND-coalgebras can be regarded as a collection of coalgebras underlying non-deterministic situations, and creates a setting in which many other structures could be suitably described. A possible issue to be tackled in the future might be the coalgebraic explanation of a more general type of multisemilattices and multilattices which were thoroughly studied in [6]. For this purpose, it would be necessary to extend the definitions and properties introduced for binary and doubly binary ND-coalgebras.

References

1. M. Benado. Asupra unei generalizări a noţiunii de structură. *Acad. RP Romania, Bul. St., Sect. Mat. Fiz.*, 5:41–48, 1953.
2. P. Cordero, G. Gutiérrez, J. Martínez, and I. P. de Guzmán. A new algebraic tool for automatic theorem provers. *Ann. Math. Artif. Intell.*, 42(4):369–398, 2004.
3. P. Corsini and V. Leoreanu. *Applications of hyperstructure theory.* Kluwer, 2003.
4. S. Ghilardi. Unification in intuitionistic logic. *The Journal of Symbolic Logic*, 64(2):859–880, 1999.
5. J. Khan and A. Haque. Computing with data non-determinism: Wait time management for peer-to-peer systems. *Computer Comm.*, 31(3):629–642, 2008.
6. J. Martínez, G. Gutiérrez, I. P. de Guzmán, and P. Cordero. Generalizations of lattices via non-deterministic operators. *Discrete Math.*, 295(1-3):107–141, 2005.
7. D. Varacca and G. Winskel. Distributing probability over non-determinism. *Mathematical Structures in Computer Science*, 16(1):87–113, 2006.

Error Scaling in Fault Tolerant Quantum Computation

Marco Lanzagorta[1] and Jeffrey Uhlmann[2]

[1] ITT Corporation, 2560 Huntington Ave., Alexandria, VA 22303, USA
marco.lanzagorta at itt.com
[2] University of Missouri-Columbia, Columbia, MO 65211, USA
uhlmannj at missouri.edu

Abstract. The threshold theorem states that quantum computations can scale robustly in the presence of certain types of noise processes (e.g., Markovian) as long as the probability of error for each physical component remains below a critical threshold. To satisfy this threshold a theoretical circuit requiring $O(s)$ idealized noiseless gates can be implemented with $O(s \text{ polylog } s)$ gates to maintain an error rate that is constant with increasing s. In this paper, we argue that maintaining a fixed error rate is necessary *but not sufficient* to preserve complexity results obtained under an assumption of noiseless gates. Specifically, we show that nontrivial quantum algorithms exhibit nonlinear sensitivity to *any* circuit error and that this sensitivity affects algorithmic complexity. The joint effects of circuit error and quantum-algorithmic iteration are examined for the case of quantum search, and more complete complexity results are derived.

Keywords: Quantum Computing, Quantum Error Correction, Fault Tolerant Quantum Computation, Threshold Theorem, Quantum Complexity

1 Introduction

Fault tolerant quantum computation relies on quantum error correction (QEC) to control the error probability associated with each computational component. This entails the use of a multi-qubit state to encode each logical qubit in a quantum register and the application of fault tolerant logical gates to operate on the register [1, 2]. This use of information and gate redundancy can reduce the expected average gate-error probability from p to $O(cp^2)$, where c represents the total number of points where a failure may occur. It can be shown that for h layers with QEC the total probability of error is:

$$\epsilon \equiv \frac{(cp)^{2^h}}{c}. \tag{1}$$

Thus to enforce $\epsilon < p$ requires $p < 1/c$. In other words, as long as the probability of error of a single physical component is below a certain value $p_{th} \equiv 1/c$, it is

possible to bound the overall error of the circuit to satisfy a given error threshold. This is referred to as the *threshold theorem* [3–9].

Suppose now that we have a quantum algorithm which requires a circuit consisting of s noiseless gates. The threshold theorem states that if the probability of error of each of the s gates[3] in the circuit is kept smaller than a certain threshold then the error associated with the output from the overall circuit can also be kept below a given threshold. In other words, a circuit of noisless gates can be simulated with a larger circuit of noisy gates with an error rate r that can be made to satisfy $0 < r < t$ for an arbitrarily small threshold t.

If we wish to achieve a final accuracy of $\tilde{\epsilon}$ in the simulation of this circuit then we require that:

$$\frac{(cp)^{2^h}}{c} \leq \frac{\tilde{\epsilon}}{s} \tag{2}$$

From this equation one can derive the upper bound on $\tilde{h}_o$ necessary to achieve the desired algorithmic accuracy $\tilde{\epsilon}$:

$$\tilde{h}_o \approx \log\left(\frac{\log(s/c\tilde{\epsilon})}{\log(1/pc)}\right) \tag{3}$$

It is also possible to show that the size of the noisy circuit grows as d^h, where d is a constant that represents the maximum number operations used in a fault tolerant procedure for a single logical gate. Thus, the circuit complexity scales as:

$$\mathcal{O}(s) \longrightarrow \mathcal{O}(s \times d^{\tilde{h}}) = \mathcal{O}\left(s \times \left(\frac{\log(s/c\tilde{\epsilon})}{\log(1/pc)}\right)^{\log d}\right) \tag{4}$$

where

$$d^{\tilde{h}} = \left(2^{\log d}\right)^{\tilde{h}} = \left(2^{\tilde{h}\log d}\right) = \left(2^{\tilde{h}}\right)^{\log d} \tag{5}$$

With p, c, and d as constants we obtain the following:

$$\mathcal{O}\left(s \times \log^r(s/\tilde{\epsilon})\right) \tag{6}$$

where:

$$r \equiv \log d \tag{7}$$

Therefore, under the conditions of the threshold theorem, a fault tolerant quantum circuit incurs only a poly-logarithmic overhead factor on the number of noisy gates.

At this point it is important to consider what the threshold theorem implies and does not imply. It ensures that an ideal circuit of s noiseless gates can be simulated by circuit of noisy gates that is only larger by a polylogarithmic factor

[3] The error models considered in the literature treat the gates as the only possible points of failure so that $c = s$, e.g., instead of $c = O(2^s)$ in which larger subsets of gates may jointly contribute to a fault. We will accept $c = s$ as an assumption for the analysis in this paper without any comment on its reasonableness.

and has an error rate that is less than any desired nonzero threshold. Therefore quantum *circuits* can be said to scale efficiently with bounded error. However, the threshold theorem does not necessarily imply that a given quantum *algorithm* on a simulated circuit will have the same complexity as is possible on an idealized noiseless circuit. This is because the algorithm may be superlinearly sensitive to noise, so any *nonzero* threshold may be insufficient to preserve the algorithm's complexity on a noiseless circuit.

In this paper we argue that the circuit error threshold has a functional dependency on the algorithm's sensitivity to noise. In sections 2, we review some basic concepts of complexity theory and algorithmic accuracy. In section 3 we conduct algorithmic analysis for the case of a noisy circuit with an uncorrected error probability ϵ. We show that uncorrected constant errors may affect algorithmic complexity even if they are arbitrarily small. As a consequence, these errors may alter complexity classifications. In section 4 we demonstrate that bounded – though nonzero – error affects the complexity of amplitude amplification, e.g., as applied in Grover's algorithm for quantum search. In section 5 we explain why the complexities of classical search algorithms are not affected by bounded-error gates. And in section 6 we discuss our results present our conclusions.

2 Probability Amplification

A typical quantum algorithm produces an output with a known probability of being correct. If the scaling parameters are fixed, the probability that the output is correct is a constant. This constant can be made to satisfy a given threshold t simply by repeating the algorithm a number of times until the probability of not finding the correct answer falls below $1 - t$.

For example, consider the case of a quantum algorithm with a binary output: $\uparrow$ is the correct outcome and $\downarrow$ is the incorrect outcome with the probability of getting either output is exactly 1/2. After the algorithm completes, we check if the outcome is correct or not, a process which we presume can be accomplished in constant time. After we run the algorithm once, the probability that we will obtain the right answer is 1/2. But if we run the algorithm twice, then we will obtain the right answer in 3 out of the 4 possible outcomes ($\uparrow\uparrow$, $\uparrow\downarrow$, and $\downarrow\uparrow$), and we will not get the right answer in 1 out of the 4 possible outcomes ($\downarrow\downarrow$). In other words, the probability that we get an incorrect answer in both runs is $(\frac{1}{2})^2 = \frac{1}{4}$. Similarly, the probability of getting an incorrect answer for each of three runs is $(\frac{1}{2})^3 = \frac{1}{8}$, so the probability of finding the correct answer is 7/8.

In general, if the probability of algorithmic error is $\tilde{\epsilon}$, then after k experiments we have a probability of error $\tilde{\epsilon}^k$ and a probability of success of $1 - \tilde{\epsilon}^k$. As $\tilde{\epsilon} < 1$, this process effectively reduces the probability of error and can be used to increase the probability of success of the algorithm to a number arbitrarily close to 1. Indeed, we can always choose k such that $1 - \tilde{\epsilon}^k = 1 - \delta$, where δ is the target error probability for the quantum algorithm. Explicitly:

$$k = \frac{\log(\delta)}{\log(\tilde{\epsilon})} \tag{8}$$

which is constant as long as δ and $\tilde{\epsilon}$ are constants, so iterating the algorithm k times will not change its complexity. As we will see in the next section, however, the total algorithmic error that emerges from error-correction procedures is not a constant; rather it depends on the scaling variable, i.e., the number of qubits and gates involved in the implementation of the quantum algorithm. As a consequence, this dependency affects the overal complexity.

3 Error Probability Analysis

Suppose we have a simple quantum algorithm that requires of a single gate $\hat{U}$ that is applied m times to a given quantum state. Also, we assume that the gate has a constant probability of failing, ϵ, and in such a case it produces the operation $\hat{U}_f$. This error may be due to (1) a faulty compiled design that performs the wrong operation, (2) higher order errors left after error correction, or (3) the approximate implementation of the gate using a finite set of elementary gates.

The effect of this operation on a general quantum state $\rho^{(0)}$ can be described mathematically in its Kraus representation as:

$$\rho^{(1)} = (1-\epsilon)\ \hat{U}\rho^{(0)}\hat{U}^\dagger + \epsilon\ \hat{U}_f\rho^{(0)}\hat{U}_f^\dagger \tag{9}$$

and the second iteration of the algorithm will look like:

$$\begin{aligned}\rho^{(2)} &= (1-\epsilon)\ \hat{U}\rho^{(1)}\hat{U}^\dagger + \epsilon\ \hat{U}_f\rho^{(1)}\hat{U}_f^\dagger \\ &= (1-\epsilon)^2\ \hat{U}\hat{U}\rho^{(0)}\hat{U}^\dagger\hat{U}^\dagger + \epsilon\ (1-\epsilon)\ \hat{U}\hat{U}_f\rho^{(0)}\hat{U}_f^\dagger\hat{U}^\dagger \\ &\quad + \epsilon\ (1-\epsilon)\ \hat{U}_f\hat{U}\rho^{(0)}\hat{U}^\dagger\hat{U}_f^\dagger + \epsilon^2\ \hat{U}_f\hat{U}_f\rho^{(0)}\hat{U}_f^\dagger\hat{U}_f^\dagger\end{aligned} \tag{10}$$

Thus, with probability $(1-\epsilon)^2$ there will be no faults; with probability $2\epsilon(1-\epsilon)$ there will be exactly one fault; and with probability ϵ^2 there will be two faults.

After m iterations the quantum algorithm produces the mixed state:

$$\begin{aligned}\rho^{(m)} &= (1-\epsilon)\ \hat{U}\rho^{(m-1)}\hat{U}^\dagger + \epsilon\ \hat{U}_f\rho^{(m-1)}\hat{U}_f^\dagger \\ &= (1-\epsilon)^m\ \hat{U}^m\rho^{(0)}\hat{U}^{\dagger m} + ...\end{aligned} \tag{11}$$

Define $P(j)$ to be the probability that after m iterations the algorithm is completed with j errors:

$$\begin{aligned}P(0) &= (1-\epsilon)^m \\ P(1) &= (1-\epsilon)^{m-1}\ \epsilon\ m \\ ... &\quad ... \\ P(j) &= (1-\epsilon)^{m-j}\ \epsilon^j \binom{m}{j} \\ ... &\quad ... \\ P(m) &= \epsilon^m\end{aligned} \tag{12}$$

which clearly satisfies:

$$\sum_{i=0}^{m} P(i) = 1 \tag{13}$$

Then P_{err}, the probability that after m iterations the algorithm will be completed using at least one faulty gate, is given by:

$$P_{err} \equiv \sum_{i=1}^{m} P(i) \;=\; 1 - (1-\epsilon)^m \tag{14}$$

where it is clear that the probability of algorithmic error not only depends on the net failure probability ϵ but also on m.

Note that if we expand P_{err} and only consider the leading order term in m, we get the original error model expressed in equation 2:

$$P_{err} \approx 1 - (1 - m\epsilon) = m\epsilon \tag{15}$$

where $m \propto \mathcal{O}(s)$ and $P_{err} \propto \tilde{\epsilon}$. As a consequence, P_{err} grows montonically with m and approaches 1. This functional dependency is to be expected: the more times a faulty gate is used in a procedure, the more likely it is that there will be a failure. Furthermore, in most algorithms of interst m is a function that grows with n, the number of qubits involved in the algorithm, so P_{err} goes to 1 for any constant ϵ. Therefore, if a quantum algorithm has an intrinsic success probability of $P = 1 - \delta$ using noiseless gates then the probability of success with noisy gates will be $P_f = 1 - \delta_f$ with $\delta < \delta_f$.

As has been discussed, however, the algorithm can be iterated to increase the probability of success to satisfy any desired threshold. For example, after k runs of the algorithm the probability of at least one failed operation becomes:

$$(P_{err})^k = (1 - (1-\epsilon)^m)^k \tag{16}$$

where k can be chosen to satisfy:

$$(P_{err})^k \approx \delta \tag{17}$$

for $0 < \delta < 1$. The desired value of k is then:

$$k \approx \frac{\log \delta}{\log\left(1 - (1-\epsilon)^m\right)} \tag{18}$$

For large m (large n limit) we have:

$$(1-\epsilon)^m \ll 1. \tag{19}$$

Using the small limit approximation:

$$\log(1-x) \approx -x \tag{20}$$

the result for large m is:

$$\begin{aligned} k &\approx \frac{\log \delta}{\log \left(1-(1-\epsilon)^m\right)} \\ &\approx \frac{\log \delta}{-(1-\epsilon)^m} \\ &\approx -\log \delta \times \left(\frac{1}{1-\epsilon}\right)^m \end{aligned} \tag{21}$$

Thus, the number of iterations of the algorithm required to mitigate the effect of noisy gates grows exponentially. This result is independent of the value of ϵ, which has been considered as an arbitrarily small, but constant, value. The only way to reduce the number of iterations is to reduce the value of ϵ, e.g., with more layers of error correction. However, the complexity of the overall algorithm must reflect the complexity of the extra error correction.

As an example, consider Grover's algorithm [10]. This amplitude amplification algorithm uses a quantum state of $O(n)$ qubits to search for an item in a dataset of size $N = 2^n$. The algorithm requires $O(\sqrt{N})$ iterations of the Grover operator G, which is given by:

$$G = D \times O \tag{22}$$

where O is an oracle and D the inverse around the mean operator[11]. We assume that both operators can be implemented with $O(n)$ elementary gates from a universal set of quantum gates. Therefore, we can make the following estimation:

$$m = O(n \times 2^{n/2}) = O(\sqrt{N} \log N) \tag{23}$$

Then:

$$\begin{aligned} k &\approx -\log \delta \times \left(\frac{1}{1-\epsilon}\right)^{n\ 2^{n/2}} \\ \implies k &= \mathcal{O}\left(a^{\sqrt{N}\log N}\right) \end{aligned} \tag{24}$$

where:

$$a \equiv \frac{1}{1-\epsilon} > 1 \tag{25}$$

As a consequence, the overall complexity of Grover's algorithm is:

$$Grover = \mathcal{O}\left(\sqrt{N} \times k\right) \approx \mathcal{O}\left(\sqrt{N} \times a^{\sqrt{N}\log N}\right) \tag{26}$$

Figure 1 shows the scaling implications of this result.

If the probability of error per gate, ϵ, is indeed very small, then:

$$\frac{k}{-\log \delta} \approx \left(\frac{1}{1-\epsilon}\right)^m$$

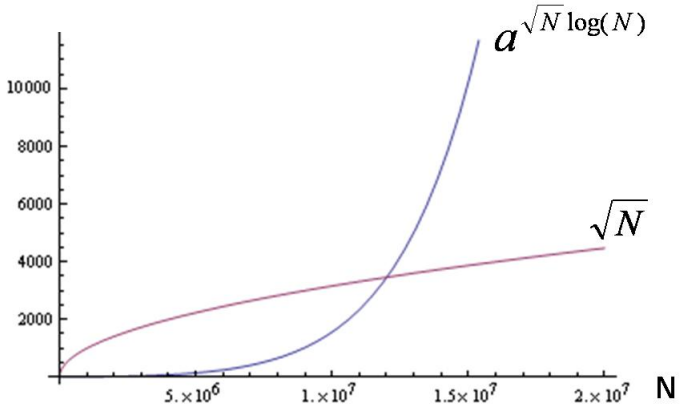


Fig. 1. Plots of $\sqrt{N}$ and $a^{\sqrt{N}\log N}$ for $N \approx 10^7$ and $\epsilon = 0.0001$.

$$\begin{aligned} &\approx (1+\epsilon)^m \\ &= \sum_{r=0}^{m} \binom{m}{r} \epsilon^r \\ &= 1 + m\epsilon + ... \end{aligned} \tag{27}$$

Thus, unless ϵ is identically zero, k will depend on m and therefore N.

If $m \propto \sqrt{N}\log N$, then:

$$k = \mathcal{O}\left(\sqrt{N}\log N\right) \tag{28}$$

and the overall complexity of Grover's algorithm is:

$$Grover = \mathcal{O}\left(\sqrt{N} \times \sqrt{N}\log N\right) = \mathcal{O}\left(N \log N\right) \tag{29}$$

This implies that the efficient theoretical complexity of Grover's algrorithm is undermined by any constant noise process, i.e., nonzero value ϵ independent of N.

4 Error Scaling and Circuit Complexity

The problem posed by a constant probability of error per gate is that errors multiply with each iteration of the algorithm and thus will grow without bound. The only way to limit this error growth is to increase the number of layers of

quantum error correction so that the probability of success for the overall algorithm is fixed as N increases. However, this means that the amount of performed error correction must increase with N because the effective value of ϵ must decrease as a function of N, which we will denote as $\epsilon(N)$. If the complexity of increasing error correction is comparable to that of the uncorrected error rate, then nothing is achieved. In this section we will show that this is not the case.

The goal is to define $\epsilon(N)$ so that the number of iterations is constant. That is:

$$k \approx -\log \delta \times \left(\frac{1}{1-\epsilon}\right)^m = O(1) \tag{30}$$

Letting $\epsilon = \epsilon(N)$ (because $m(N)$ replaces m) gives:

$$\epsilon(N) = 1 - (-\log \delta)^{1/m(N)} \tag{31}$$

with $m(N)$ expressed as:

$$m(N) = \sum \tilde{m}_i(N)\tilde{s}_i(N) \tag{32}$$

where $\tilde{s}_i(N)$ is the number of gates of type i that appear in the quantum circuit and $\tilde{m}_i(N)$ is the number of times that these gates are iterated during the quantum algorithm. In the case of Grover's algorithm these definitions lead to:

$$\epsilon(N) \approx 1 - (-\log \delta)^{1/O(\sqrt{N}\log N)} \tag{33}$$

In light of this analysis we can generalize the threshold theorem. Specifically, we require that:

$$\frac{(cp)^{2^h}}{c} \leq \epsilon(N) \tag{34}$$

where $\epsilon(N)$ now takes the form:

$$\epsilon(N) = 1 - (-\log \delta)^{1/m(N)} \tag{35}$$

which leads to:

$$\frac{(cp)^{2^h}}{c} + (-\log \delta)^{1/m(N)} \leq 1 \tag{36}$$

This expression can be used to determine the optimal number of iterations of quantum error correction layers that are required to avoid a penalty on the algorithmic complexity. That is, we require that:

$$\begin{aligned} &\frac{(cp)^{2^{\tilde{h}(N)}}}{c} \approx \epsilon(N) \\ &\implies \tilde{h}(N) \approx \log\left(\frac{\log(1/c\epsilon(N))}{\log(1/cp)}\right) \end{aligned} \tag{37}$$

As a consequence, $\tilde{h}(N)$ scales as:

$$\log\left(\log(1/\epsilon(N))\right) \tag{38}$$

in contrast to $\tilde{h}_o(N) = \log(\log(s(N)))$ derived earlier. The difference in the number of iterations is not a simple multiplicative constant, as $\tilde{h}(N)$ and $\tilde{h}_o(N)$ have completely different functional dependencies. In particular:

$$\frac{\tilde{h}(N)}{\tilde{h}_o(N)} \longrightarrow \infty \quad \text{as} \quad N \longrightarrow \infty \tag{39}$$

We can determine the number of gates required to satisfy the fault tolerance inequality:

$$\mathcal{O}\left(s \times \log^r(1/\epsilon(N))\right) \tag{40}$$

We are interested in the overhead factor on the number of gates:

$$\log\left(\frac{1}{1-\left(\frac{-\log\delta}{\xi}\right)^{1/m(N)}}\right) \tag{41}$$

and how it compares to the original case:

$$\log\left(s(N)\right) \tag{42}$$

Our complexity is poly-logarithmic rather than strictly logarithmic, but this overhead is unavoidable because:

$$\frac{\log\left(\frac{1}{\epsilon(N)}\right)}{\log\left(s(N)\right)} \longrightarrow \infty \quad \text{as} \quad N \longrightarrow \infty \tag{43}$$

similar to the scaling of the number of layers of the error correction encoding $\tilde{h}(N)$ and $\tilde{h}_o(N)$.

5 The Classical Case

We have shown that a nonzero probability of error for each gate impacts the complexities derived under the (implicit) assumption of noiseless gates. It may reasonably be questioned whether the same analysis similarly impacts classical algorithms. To show why it does not we examine the case of classical linear search and contrast its robustness to gate noise with that of Grover's quantum search algorithm.

As we did in the quantum case, we assume that the classical search oracle produces errors with probability ϵ. Most of the analysis remains the same, with the exception that $k \approx \mathcal{O}(N)$, and the overall complexity becomes:

$$\textit{Classical Brute Force} \approx \mathcal{O}(N \times N) \approx \mathcal{O}\left(N^2\right) \tag{44}$$

Interestingly, under the exact same assumptions of uncorrectable faulty gates with a small error probability, Grover's algorithm provides a quadratic improvement in scaling. Within the classical framework, however, there is additional

flexibility to address errors so that noisy gates can be used to simulate noiseless ones without impacting algorithmic complexity.

Consider the case of a classical brute-force (CBF) search of a dataset S of size N using an oracle O, where each application has a probability ϵ of producing an errored result, i.e., returning a '0' for the correct solution or a '1' for an incorrect solution.

Our algorithm is as follows: for each element S_i of our dataset we evaluate $\mathcal{O}(S_i)$. If S_i is assessed to be a solution then we re-evaluate $\mathcal{O}(S_i)$ m times to ensure that it is in fact a solution. The probability that a spurious solution passes all m tests is ϵ^m, so the probability that S_i is a true solution is approximately[4] $1 - \epsilon^m$.

The expected number of spurious solutions that will be initially assessed as actual solutions is $\epsilon \times N$, so $\mathcal{O}(m \times \epsilon \times N)$ re-evaluations will be performed, and the complexity of a single execution of the algorithm becomes:

$$\mathcal{O}((1 + m \times \epsilon) \times N). \tag{45}$$

If no solution is found then the entire iteration will have to be re-executed. This will occur with probability ϵ – the case in which an error occurs when the oracle is applied to the true solution. The probability that the entire iteration is executed k times is ϵ^k. Including this multiplicative factor into the complexity of Eqn.(1) gives an overall complexity of:

$$\mathcal{O}((1 + m \times \epsilon)k \times N) \tag{46}$$

If we want to fix the probability p of finding the correct solution for any value of N, we need to ensure that the probability of returning a spurious solution (ϵ^m) and the probability of failing to recognized the correct solution (ϵ^k) are both less than $1 - p$. This means that k and m must be less than $\alpha = \log(1 - p)/\log(\epsilon)$, which is a constant for fixed p and ϵ. Substituting α for m and k into the overall complexity given in Equation 46 shows that classical brute force has $\mathcal{O}(N)$ complexity in the presence of uncorrected constant errors.

The difference between the classical and quantum models is that in classical search each computational step determines if an element is a solution or not. In the quantum case, by contrast, the no-cloning theorem precludes the ability to re-run a particular step in the algorithm. The poly-logarithmic space increase needed in the quantum case cannot generally be ignored because many classical algorithms admit a nonlinear space/time complexity tradeoff, i.e., a classical algorithm may be able to reduce its associated run-time complexity if given the same poly-logarithmic factor increase in space.

6 Conclusions

In this paper we have shown the following:

[4] Errors may occur during the m iterations, but their contribution to the overall analysis is negligible.

– Even if it is arbitrarily small, a *constant* uncorrected error probability will undermine the complexity advantages of most (if not all) quantum algorithms. To mitigate this effect it is necessary to apply error correction to scale the error level according to the scaling parameters of the algorithm.
– This scaling of the error probability can be used to compute the maximum error probability allowed at a given scale, the optimal number of layers of the error correction encoding, and the overhead in circuit size that results from fault tolerant procedures.
– In the classical domain it is possible to have uncorrected constant error probabilities that do not affect algorithmic complexity. This is not the case in the quantum domain because of the restrictions imposed by the cloning theorem.
– The error scaling behavior implies a more demanding model of fault tolerant quantum computing. The required number of layers of error correction encoding and the overhead in the size of the circuit are larger than those previously reported in the literature.

From a practical point of view, these results provide useful formulas for the optimal number of layers of quantum error correction encoding determined by the specific algorithm algorithm that needs to be implemented in noisy hardware. This is particularly useful for the design of smart compilers able to dynamically allocate the the optimal amount of error correction for a given program. In these circumstances, trade-offs between logical quantum gates and time may become important.

Acknowledgments

ML's research efforts were supported by a grant from the Office of Naval Research (ONR) Quantum Information Science (QIS) Basic Research Challenge (BRC) program.

References

1. M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
2. F. Gaitan, *Quantum Error Correction and Fault Tolerant Quantum Computing*, CRC Press, Boca Raton, 2008.
3. D. Aharonov and M. Ben-Or, "Fault tolerant computation with constant error", *Proceedings of the* 29^{th} *Annual ACM Symposium on the Theory of Computing*, 1997.
4. D. Aharonov and M. Ben-Or, "Fault tolerant quantum computation with constant error rate", quant-ph/9906129, 1999.
5. A.Y. Kitaev, "Quantum Computations: Algorithms and Error Correction", *Russ. Math. Surv.* 52 (1997) 1191-1249.
6. A.Y. Kitaev, "Quantum error correction with imperfect gates", in: *Quantum Communication, Computing, and Measurement*, A.S. Holevo, O. Hirota, and C.M. Caves (eds.), Plenum Press, New York, 1997, pp. 181-188.

7. D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. Thesis, California Institute of Technology, 1997.
8. J. Preskill, "Reliable Quantum Computers", it Proc. R. Soc. London, A454 (1998) 385-410.
9. E. Knill, R. Laflamme, and W.H. Zurek, "Resilient quantum computation", *Science*, Vol. 279, No.5349 (1998) 342-345.
10. L.K. Grover, "Quantum mechanics helps in searching for a needle in a haystack", *Phys. Rev. Lett.* 79 (1997) 325-328.
11. M. Hirvensalo, *Quantum Computing*, Springer, Berlin, 2001.

Computation and the Illusion of Physical Reality

(An Informal Presentation to Physics and Computation 2010)

Mike Stannett *

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom
m.stannett@dcs.shef.ac.uk

Abstract. The role of theoretical physics is to investigate, represent and thereby explain the nature of physical reality. We claim that this goal is unattainable using current standard mathematical models of physics, not just for practical reasons, but as a matter of logical necessity. Standard models of quantum theory and relativistic spacetime are logically equivalent to models in which the nature of classically observable motions is a form of *necessary illusion*. Consequently, no standard deductions as to the nature of space, time and motion can be deemed sound.

Keywords: physics and computation, formal models of physics, arrow of time, interpretations of quantum mechanics, first-order relativity theory

1 Manifesto

All mathematical theories of physics (even quantum theories) ultimately depend for their validation upon classical observations. This is inevitable, since no matter what form a physical apparatus may take, the observations made using that apparatus must ultimately be conveyed to and interpreted by human beings using biological sensory systems that have evolved, for better or worse, to interpret the world directly in classical terms.

Being classical, these observations all involve a physical instantiation of motion. Whether it be the movement of a needle on a voltmeter, the creation by subatomic particle of a path in a cloud chamber, the collision of a photon with an observer's eye, or the arrival of salt molecules on a subject's tongue, classical observation cannot exist in the absence of motion.

Ultimately, therefore, current mathematical theories of physics are really only theories of observable classical motion. Those theories which correctly predict how entities will move in an experimental system survive, while those which fail to do so are rejected. The underlying theories need not themselves be classical, but the predictions they generate must be expressed in classical terms if they are to be humanly testable.

* We are grateful to the EPSRC for their support (EPSRC Hypercomputation Network (HyperNet), grant reference EP/E064183/1).

If it can be proven of two distinct physical theories that they necessarily generate the same physical predictions under all applicable circumstances, those theories either stand or fall together (we shall call such theories *equivalent*). If one represents an unfalsified description of physical reality, then so does the other. If one is invalidated by experimental observations, then so is the other.

Suppose, then, that M_1 and M_2 are equivalent theories of physics, and that M_1 incorporates axiomatically some independent assumption A_1 concerning the nature of physical motion, while M_2 incorporates another such assumption A_2. If the assumptions A_1 and A_2 contradict one another in physical terms, then neither assumption can be deemed intrinsically sound, since all experiments which would validate A_1 by validating M_1 would also validate A_2 by validating M_2. While both assumptions remain meaningful components of their respective theories, any claim for the ultimate physicality of one as opposed to the other must be considered unscientific as long as both theories remain unfalsified by experimental data.

In such circumstances, the apparent validity of two contradictory theories may be seen as pointing to an incompleteness in our understanding of physical reality. Either the notion of physical motion is logically irrelevant, so that both A_1 and A_2 may simply be dispensed with, or else (which is essentially the same statement) both A_1 and A_2 can be deduced as theorems by adding some deeper axiomatization of motion to the theories $M_1' = M_1 \setminus A_1$ and $M_2' = M_2 \setminus A_2$, respectively.

2 Argument

We present two arguments that no assumption can be deemed sound, given currently accepted mathematizations of physics, that purports to characterise the 'true' nature of physical motion, whether in space or time, as either discrete or continuous. The nature of motion – and hence of humanly observable reality – must run deeper than this: physical reality, as currently understood, supports equivalent theories incorporating discrete and continuous representations of motion, and so motion cannot itself meaningfully be constrained to either condition. We conclude that *all* observable motion may be deemed an artefact of our models, whence physical reality as we currently formulate it is an unscientific illusion. In truth, we understand nothing.

2.1 The continuity/discontinuity argument

In [Stannett(2009a)] we considered the case of quantum theory. We showed that Feynman's path-integral formulation of quantum theory is logically equivalent to a computation-based theory in which particles jump at random from one spacetime location to another, provided the action associated with each such 'hop' is set equal to the classical action for the same relocation (or that of the time-reversed equivalent anti-particle in the case of hops taking the particle backwards through time). We claim that Feynman's theory (which is itself equivalent

to Schrödinger's) assumes that time's arrow is forward-pointing, and that basic particle trajectories are continuous paths drawn on a spacetime that is itself a continuum. The hop-based model also assumes that spacetime is a continuum, but assumes that motion is discrete and that time has no arrow. As explained in Sec. 1, it follows that neither continuity nor discreteness of motion can safely be deemed fundamentally valid, and that any attempt to assert the physicality of one above the other is inherently unscientific. In particular, therefore

– the classical contention (inherent in Newton's laws, for example) that particles move forward continuously through time along continuous spatial paths cannot be supported. We can instead regard both the arrow of time, and the associated concept of continuous motion as a form of 'necessary illusion', forced upon classical observers even within the inherently discrete hop-based theory.

2.2 The measurement-field argument

In [Stannett(2009b)] we looked at first-order models of relativity theory, and noted that there is no obvious reason why the number field Q used to record measurements (for example, of mass) should be the same as the field R used to coordinatize spacetime, though we would expect Q to be a subfield of R. By adapting the argument in [Stannett(2009a)], we argued that the 'hops' used to generate the illusion of continuous classical motion can be chosen so that any particle whose location is coordinatized entirely by Q is constrained to remain absolutely fixed in space and time. Since all observations involve such objects, this tells us that no 'truly observable' entity in this theory is capable of motion. Nonetheless, we again obtain the 'necessary illusion' of classical motion along continuous paths. It follows that

– the classical contention that objects can move *at all* can also be regarded as a necessary illusion caused in part by our incorrectly identifying the field Q of physically measurable values with the field R of idealised coordinates.

2.3 Conclusion

Since no reliance can be placed on the classical physicality of motion, and since current physical theories require the observation of such motions for their validation, no current physical theory can be considered to have explanatory power. We understand and can explain nothing about the physical world.

References

[Stannett(2009a)] M. Stannett, The Computational Status of Physics, Natural Computing: an international journal 8 (3) (2009a) 517–538.

[Stannett(2009b)] M. Stannett, Modelling Quantum Theoretical Trajectories within Geometric Relativistic Theories, in: Mathematics, Physics and Philosophy in the Interpretations of Relativity Theory (PIRT 2), Budapest, 4-6 Sept, 2009b.

A Note on the Categorical Nature of Causality (II)

Karin Verelst

FUND-CLEA
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussels
kverelst@vub.ac.be

Abstract. Discussions on causality abound, but rare are the attempts at precise definition of what is meant. The reason might be that the concept in itself is intrinsically pluriform, but even then theories enclosing some kind of causation should exhibit certain common structural characteristics, otherwise the use of the common term would be absolutely pointless. I show that a fairly straightforward categorical characterisation of causation is possible when one takes both the history of the concept and Meyerson's careful analysis of the relation between causation and time into account. Historically it has been seen (by Aristotle) that a causal relation between events is never simply straightforward, but always implies — explicitly or not — a connection between a universal (global) and a particular (local) level. This is why the idea of cause can be linked to the idea of lawfulness. But there is a difference between a law and a cause because of the asymmetry between space and time: space is actual everywhere but time only at this moment. Laws define the identical, but identity as well is only unproblematic at this moment. Meyerson shows that causality therefore somehow implies the conservation of identity through time. The idea of conservation is essential here. Now when causal connections are interpreted as order relations (as is the case in, e.g., relativistic theories), then causation appears as the Galois adjoint to identity, and causality will be aequivalent to the idea of physical law. This allows to formally characterise causality in this type of theories, without having to "explain" it any further. Given the functoriality of the derivative and the interconnection between symmetry and conservation, this approach might be generalisable to other physically viable notions of causation through the use of Noether's Theorem.

References

1. F. Borceux, *Handbook of Categorical Algebra I*, Cambridge University Press, Cambridge, 1994.
2. E. Meyerson, *Identité et Réalité*, Félix Alcan, Paris, 1932.
3. E. Noether, "Invariante Variationsprobleme", *Nachr. d. König. Gesellsch. d. Wiss. zu Göttingen, Math-phys. Klasse*, pp. 235–257, 1918.
4. K. Verelst, "On what Ontology Is and not-Is", *Foundations of Science*, **13**, 3, 2008.

Author Index

PHYSICS AND COMPUTATION 2010