Common Sense (What is it?) and the Church Turing Hypothesis

P.J. Marcer,
55 rue Jean Jaures, 83600, Frejus, France
peter.marcer@orange.fr

Abstract
I have long felt that computation is a matter of common sense - English dictionary, sound practical judgement. This paper presented at the British Theoretical Computer Science Colloquium at the University of Warwick in March 24-26, 1986. proposed a definition of common sense as essential element in the solution of the problem of machine intelligence. It concluded, that common sense is both the deep syntax of natural processes and governs the basic semantics of logical or mathematical language. Then published only as an abstract, I felt it merited inclusion in CASYS 07
Keywords: Universal Turing computation, universal quantum computation, grammar, common sense, machine intelligence.

1 Introduction

This paper was/is divided into the following parts; a non technical section 2 with various subheadings and a technical section 3 with the following subheadings:
3.1 Deutsch's extension of Turing computation to quantum computation, including the subclass of quantum computers that define Turing computability.
3.2 The logical grammar appropriate to the class of quantum computers.
3.3 Gefwert's methodology by which he extended Martin-Lof's intuitionistic theory of types, this being an extension of Russell's doctrine of types, to become a logical depth grammar in Wittgenstein's sense in accordance with his proposition as rules idea. And a statement of the logical depth grammar.
3.4 The rules of definitional equality which apply to the logical depth grammar.

2 Non Technical Presentation of Computation as Common Sense

2.1 AI Hype

In order to illustrate this 1986 point of view, let me be a little unfair to one of the foremost and highly regarded specialists at the leading edge of the field, who said that a true learning system must have an encyclopaedic knowledge base. Any member of the public (using his/her common sense) seeing a new born child knows this to be wrong. Each child starts life with little or no knowledge base, yet it learns at an astonishing pace, and builds such a base as a consequence of a learning process.

International Journal of Computing Anticipatory Systems, Volume 22, 2008
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-09-1

However this specialist comes much closer when he said what experiments in machine learning lacked, was common sense. Common sense is however surely the colloquial phrase for what we all universally possess, an inherent inferential logical grammar that allows us to build logical models of 'reality' from our experience; these models being essential to our short and long term survival. And if this grammar is the basis of the learning process then ultimately I believe that we will be forced to conclude from the evidence collected from the very necessary extensive AI experiments conducted so far that ultra intelligent machines must be based on something other than general purpose Turing computation, whether by sequential or concurrent processing. That is Turing computation will allow us to demonstrate intelligent like behaviour, but not true intelligence. Is this why there have been no major (theoretical) breakthroughs in AI ?

2.2 The Reasons for my Belief that Models of AI Based on Turing Computation are Inadequate

I do not take seriously the prevailing belief that we or the ultra intelligent machines that are its goal, need be no more than a suitable collection of PCs loaded with the appropriate programs for intelligent symbol manipulation etc., or with the contention, widely held, that natural language lacks mathematical description or precision in regard to communication. For with regard to the former is it likely that if as Godel proved that arithmetic cannot be reduced to a finite set of algorithms, that intelligence can be so reduced. For I assume that when I speak to you or you to me, that the spontaneous reasoning of our speech must take place with complete mathematical consistency, so that if we are wrong in what we say, it must in general spring from the incompleteness of our knowledge, and not from the basic design of the human inference engine, as originated and tested throughout the long process of human evolution.

Equally Turing computation implies that all meaningful human communication ought to take place via exchange of algorithms; a very rarely used form. Similarly humans very easily acquire natural language, which digital computers can learn hardly at all. Why is this? It is that digital computation lacks common sense, a hypothesis that questions the Church-Turing hypothesis, that all forms of computable functionality or operations can be ultimately reduced to those that a Turing machine can compute?

2.3 Why the Church-Turing Hypothesis must be Brought into Question

Practical general purpose Turing computation must be called into question, firstly in the continuous regime of classical physics, because Turing computation, a purely discrete process is unlikely to be possible physically (in this context). For I know that the physical world is, as far as is known, ultimately describable through quantum theoretic processes. Is it these that make the Turing computation physically possible? For then the leading question is why is it possible to build a practical general purpose Turing computational engine? The answer - the laws of physics allow this! Our logical designs for discrete systems can thus be translated into physical (computing) machines; and call on the very quantum nature of physical processes to so do. The fact that Turing computation is possible physically is therefore of great fundamental importance not just practically but theoretically. It must be explained and it must be supposed that the existing Church Turing Hypothesis contains this implicit physical assumption.

How does this reflect on AI? We can either assume that our mental processes (i.e. our human intelligence) are grounded entirely on physical processes or not. If not, then AI science is in a dilemma for it assumes that these non-physical mental processes can be emulated on a general purpose Turing computer working through physical processes. And if yes then there is no reason to suppose that nature should not use such physical processes too, since in the final analysis no human observer could know the difference. Thus we must logically suppose the universality of quantum physical processes and hence of physical law or abandon the search for AI.

2.4 AI Turns Computer Science on its Head

The above conclusion is thus the first line of enquiry which should be eliminated as a possibility. That is, we must follow Bennett and Landauer [1985], that physical law must set finite limitations to the execution of all computable mathematical operations through which such law must ultimately be expressed. And conclude that thinking and indeed all mathematical activity, the current basis of computer science, is therefore conditioned by one's own mental (physical) activity governed through the laws of physics.

Thus constructive mathematics and computation does not subsume physics and physical processes as is widely believed, but co-exists in juxtaposition to physics; otherwise general purpose Turing computation would not be physically possible. What evidence is there for this? Certainly computation in current machines is subject to limitation by the speed of light, a feature absent from any of the existing models of Turing computation or computer science that I am aware of. How can this be? Clearly it is not true, that there exists in the physical world external to us, physical systems corresponding to all the abstract symbolic mathematical models known to exist; however this does not prevent these existing as physical systems in the brains of such mathematicians. It would explain why mathematics works! Why when we do model any finitely realizable physical system through say an appropriate differential equation, and predict, via the axioms of the calculus, the future behaviour of the system, that one gets valid results.

For this is indeed why science is judged to be so successful. It doesn't mean however that the physical system obeys these axioms in determining its own physical behaviour? Alternatively there may be no precise solution to the differential equation, yet reducing this equation to a difference equation, and then to a general purpose digital computer algorithm frequently can yield equally successful predictions. Why should this be?

If physical processes and their mathematical descriptions are inexorably linked through some new form of the Church-Turing Hypothesis, then all would be explained.

2.5 The Church-Turing Principle of Deutsch

In his (then recent) Royal Society paper, Deutsch (1985) has taken these arguments to their inevitable conclusion, and annunciated the Church-Turing Principle that '*every finitely realizable physical system can be perfectly simulated by a universal model computing machine working by finite means*'. This replaces the existing quasi mathematical Church Turing Hypothesis, one version of which is '*Every function which would be naturally be regarded as computable can be computed by the universal Turing machine.*' Deutsch then shows that subject to the Third law of Thermodynamics that '*no finite process can reduce the entropy or temperature of a finitely realizable physical system to zero*' that this principle is compatible with the notion of universal quantum computation where

a) universal quantum computers can do everything that universal Turing computers can do but have many other remarkable properties that are not Turing reproducible, and

b) that these physical processes can simulate universal Turing computation.

Thus it appears that just as classical physics has been replaced by quantum physics, so classical computer science i.e. that of Turing, must give way to quantum computer science, and that models of AI or intelligence should be based on the brain as a quantum rather than a Turing computer.

The former must surely be correct for one of foremost scientific discoveries of this, concerns the quantum inseparability of any finitely realizable physical system from any other and from the universe itself. Thus the human brain in the final analysis is no mere finite machine but inseparable quantum physically from every other brain and with the Universe that created it in the course of its evolution. And my common sense tells me that this model is much more in accord with my experience than existing ones grounded on universal Turing computation, for it says the latter may indeed show intelligent like behaviour in the way that expert systems do, but no true intelligence or ability to learn the way humans do. This is a property of universal quantum computers.

2.6 Common Sense

I therefore postulate that common sense is the universal property of universal quantum computers in the same way that universal Turing computation is the property of digital computers.

Universal quantum computation tells us that there must exist mathematical operations beyond those that allow us to simulate universal Turing computation. It is the physical nature of these very special operations therefore that must concern us. Deutsch (1985) tells us that they concern the mathematical operations linking the continuous and the discrete in relation to the quantum theoretic model for a single bit; i.e. the unitary operations on the two dimensional Hilbert space, the state space of the single bit. Moreover these operations are of a universal nature. They therefore define a universal set of transformations between the continuous and the discrete – a problem that has occupied natural philosophers and research workers in many fields since antiquity.

However I believe a much more appropriate view, in regard to intelligence, comes from the fact that physical law sets finite limitations to the execution of all computable mathematical operations through which such law must ultimately be expressed. In the final analysis this statement implies that physical law expressed in mathematical or logical form, must set logical limitations to the operation of logic or logical statements. Thus there must be a logical form that mathematical language must take if it is to remain mathematically or logically consistent (physically).

This has been demonstrated by C. Gefwert., a Finnish philosopher and logician and follower of Per Martin-Lof. An examination of both Gefwert's paper (1985) and Deutsch's (1985) shows that despite the radically different nature of their approaches that both the logical form of mathematical language and quantum computation necessitate the same rules of definitional equality or inferential logical grammar.

2.7 What does this Mean?

It means that there is indeed an inherent universal inferential logical grammar appropriate to all logical constructs we make through our (physical) mental processes. I call this grammar, common sense and say that it is the same as that generally referred to colloquially. Would it be surprising if logic, like any other language had its own grammar and that this grammar was universal? With hindsight this seems almost self evident. It makes clear why progress in AI has in the past settled on say Prolog so much closer to natural language than algorithmic programming languages. For just as Turing computation is the appropriate universal model for defining the syntax of algorithmic languages, then universal quantum computation must be the model for natural languages including mathematics.

Common sense therefore transcends Turing computability, and I believe it makes possible a new architecture of semantic machines and a science of semantic engineering, which encompasses both natural language and the engineering of thermodynamically possible machines (which will of course contain the existing so called general purpose Turing computers as a subset)[1]

2.8 Intermezzo – the Thesis

There is an especially intimate relationship between physical processes and their mathematical representations; a generic statement of which is the Church-Turing

---

[1] In particular, it becomes clear in 2007 with hindsight that the anticipatory computation, and the extensions to recursion, of incursion and hyperincursion, which Professor Dubois has so presciently pioneered, can be identified as examples of semantic computation and engineering. For the concept of semantics and grammar, would provide an alternative explanation of why these novel methodologies are so successful in arriving rapidly a sound computational solutions to difficult problems and of why the corresponding recursions in the form of ordinary digital computation often fail to do so.

principle of Deutsch. The reader may however feel that a formal philosophical investigation of mathematical language i.e. the process of construction of proofs through the theory of types, such as Gefwert has carried out, can have little to do with logical structure of quantum theory, which is the currently best available physical theory which science possesses to describe the fundamental processes of the natural world. However, even a perfunctory examination yields significant similarities.

Firstly Gefwert's paper (1986) concerns the essential logical rules or grammar which must govern the naming or labelling procedures for sets if the consistency of mathematical language is to be maintained; while quantum theory provides some of the most accurate models we have of the real world, where such models have been experimentally validated through real world entities described in relation to their attributes and dynamical behaviour. Thus, the physicist is able to attach labels to specific attributes and to assign to these specific numerical values so that these values are consistently reproduced when an experiment is repeated. Moreover these experimental procedures are essentially processes of counting against well specified standard measures. Quantum theory therefore shares with Gefwert, this concern with labels both in respect to type and value. And as we shall see in the next section, Deutsch in his approach to computation is also specific in his concern to labelling i.e. how the input and output states of a computational machine should be referred to. Analogously Gefwert is equally concerned that his labelling or typing procedures should have a computational nature. It appears from Deutsch's work that universal quantum computers are basically 'type' or labelling computers capable of dividing an environment into sets or domains of functionality by attribute, where no two different classes have an element in common, and every element is in some class. Further it is then possible to assign values to these classes of attributes by counting. And this is just what the Gefwert's rules of definitional equality require in respect of sets in regard to their type, and what Turing computation/'counting' will result in as regards value.

Moreover Conway has show in his book (1976) where Gefwert definitional equality rules apply in the form of lexicographical functionality in terms of labels or names, that these labels can be extended in the simplest possible way to sums, products, inverses, algebraic and transcendental extensions as successively more complicated concepts so as to form mathematically closed fields. But as Conway also shows these rules for number also generate all the numbers by value great and small including the transfinite and the infinitesimal. That the number/symbol for 1 should have the value1 is therefore, it can be hypothesized, a quite remarkable property of our universe i.e. it is the laws of physics, i.e. Pauli exclusion, that ensure that this is so! And it means that if the property of one, is the integer 1, then the computation of the property of the integer n it must have the value of n as the result. But there is now a limit imposed by the universe and its finite resources on the values of the integer n that can be computed in practice, i.e. although numbers as labels can always be exhibited say 1000^{1000} such a number symbol may be beyond the bounds of calculation.

This distinction between type and value is one which Gefwert finds essential to his analysis of mathematical language i.e. the construction of proofs.

3 Technical Presentation

3.1 Turing Computation

This part begins with a statement of how the usual definitions of Turing computation may be extended in order to define universal quantum computation. It is kept brief since a full and expertly detailed description is to be found in Deutsch (1985).
He tells us that two classical deterministic machines are 'computational equivalent' under given labellings of their input and output states if they compute the same function f under these labellings. But a quantum computing machine and indeed classical stochastic machines, do not compute functions in this sense; for the output state of stochastic machine, is random with only the probability distribution function for the possible output state depending on the input state.
The output state of a quantum machine although fully determined by its input state, is not an observable and so the user cannot in general observe its label. Nevertheless the notion of computational equivalence can be generalized to apply to such machines so that as far as input is concerned labels may be given for each of the possible ways of preparing the machine which corresponds by definition to all the possible input states. However whereas a quantum system can be prepared in any desired permitted input state, measurement cannot in general determine its output state and instead one must measure the value of some observable. Thus what must be labelled is the set of ordered pairs consisting of an output observable and a possible measured value of the observable i.e. an Hermitian operator and one of its eigenvalues. Such an ordered pair contains specifications of a possible experiment that could be made on the output, together with a possible result of that experiment.
Two quantum computers are then computationally equivalent under given labellings if in any possible experiment or sequence of experiments in which their inputs were prepared equivalently under the input labellings and observables corresponding to each other under output labellings were measured and the measured values of these observables for the two machines would be statistically indistinguishable. That is, the probability distribution function of the two machines are identical.
Such operations are therefore formalized by considering computing machines with two inputs, where the preparation of one constitutes a 'program' determining which function of the other is to be computed. To each such machine M, there corresponds a set C(M) of 'M computable functions' and a function f is M computable if M can compute f when prepared by some program. Like a Turing machine therefore, a model quantum computer Q has two components, a finite processor and an infinite memory of which only a finite portion is used. The computation may therefore proceed in steps of fixed duration T and during each step the processor and a finite part of the memory interact, the rest of the memory remaining static:-
The processor consists of M 2states observables $\mathbf{n} = \{n_i\}$ $(i \in \mathbf{Z}_M)$ where $\mathbf{Z}_M$ is the set of integers from 0 to M-1
The memory consists of an infinite sequence $\mathbf{m} = \{m_i\}$ $(i \in \mathbf{Z})$ of 2 state observables, and corresponding to the Turing machines tape position, in the quantum machine is

another observable **x** which has the whole of **Z** as its spectrum; x is the address number of the currently scanned tape location.
Thus the state of Q is the unit vector in the Hilbert space ***H*** spanned by the simultaneous eigenvectors $|\mathbf{x}\,;\mathbf{n}\,;\mathbf{m}> \equiv |\,x\,; n_0, n_1, \ldots n_{m-1}\,; \ldots m_{-1}, m_0, m \ldots >$
labelled by the corresponding eigenvalues of **x**, **n**, **m**, so that these form the computational basis states for the machine, where a 2 state observable with spectrum (0,1) ie Z_2 has a natural interpretation as a 'one-bit' memory element. The dynamics of Q can then be summarized by a constant unitary operator **U** on ***H*** such that if **U** specifies the evolution of any state $|\psi(t)> \varepsilon\, H$ during a single computational step then

$$|\psi(nT)> = \mathbf{U}^n\, |\psi(0)> \quad (n\, \varepsilon\, Z^+) \text{ and } \mathbf{U}^{+}\mathbf{U} = \mathbf{U}\mathbf{U}^{+} = 1 \qquad (1)$$

Thus if the computation begins at t = 0 and if at that time, **x** and **n** are prepared with value zero,and the state of a finite number of **m** is prepared as the program as input and the rest are set to zero, then $|\psi(0)> = \sum_m \lambda_m |0;\mathbf{0};\mathbf{m}>$ and $\sum|\lambda_m|^2 = 1$ where only a finite number of λ_m are non-zero and λ_m vanishes whenever an infinite number of the **m** are non-zero.
The 'by finite means' requirement for Q means the elements of **U** take the form

$$<x';\mathbf{n}';\mathbf{m}'|\mathbf{U}|x,\mathbf{n},\mathbf{m}> = [\delta^{x+1}{}_{x'}\,,\ \mathbf{U}^+(\mathbf{n}',m_x'|\mathbf{n},m_x) + \delta^{x-1}{}_{x'}\,\mathbf{U}^-(\mathbf{n}',m_x'|\mathbf{n},m_x)] \prod_{y \neq x} \delta^{my}{}_{my}$$

The continued product on the right ensures that only one memory bit the xth, participates in a single computational step. The terms $\delta^{x+1}{}_{x'}$ ensure that during each step the tape position x cannot change by more than one unit forwards, backwards or both. The functions $\mathbf{U}^{\pm}$ which represent a dynamical motion depending only on the 'local' observables $\mathbf{m_x}$ and **n** are arbitrary except for the requirement that **U** be unitary. Each choice defines a different quantum computer $Q(\mathbf{U}^+,\mathbf{U}^-)$.
Turing machines are said to halt, signalling the end of computation, when two consecutive states are identical and a valid programme is one that causes the machine to halt after a finite number of steps. However (1) shows that two non trivial consecutive states of a quantum computer Q can never be identical and moreover Q must not be observed before the computation is ended since this would alter its relative state. Therefore quantum computers need to signal actively that they have halted and so one of the processors bits say n_0 must be set aside for this purpose. Every valid Q program then sets n_0 to 1 when it terminates but does not interact with it otherwise. The observable n_0 can then be periodically observed from the outside without effecting the operation of Q and the analogue of the classical condition for a program to be valid would be that the expectation of n_0 must go to 1 in a finite time.
Deutsch then shows that because quantum computers $Q(\mathbf{U}^+,\mathbf{U}^-)$ are necessarily reversible, the unitarity of their dynamics, may be obtained by taking

$$\mathbf{U}^{\pm}(\mathbf{n}', m'|\, \mathbf{n}, m) = 1/2\ \delta_{\mathbf{n}'}{}^{A(\mathbf{n},m)} \delta_{m'}{}^{B(\mathbf{n}.m)}\, [1 \pm C(\mathbf{n},m)]$$

where A,B,C are functions with ranges $(\mathbf{Z}_2)^M$, $\mathbf{Z}_2$ and (-1,1) respectively

Turing machines are thus those quantum computers whose dynamics ensure that they remain in computational base state at the end of each step given that they start in one, where to ensure unitarity it is necessary and sufficient that the mapping

$$((\mathbf{n},m)) \longleftrightarrow ((A(\mathbf{n},m),B(\mathbf{n},m),C(\mathbf{n},m)))$$

is bijective. And since the functions A,B,C are otherwise arbitrary there must in particular exist choices that make Q equivalent to a universal Turing machine T.
By appealing to this Turing universality and the definition of M computability, for every recursive function f there exists a program $\pi(f)$ of T such that when the image of $\pi(f)$ is followed by the image of integer i in the input of T, T eventually halts with $\pi(f)$ and i themselves followed by the image f(i) with all the other bits zero.
That is, for some positive integer n $U^n|0;\mathbf{0};\pi(f),i,\mathbf{0}\rangle = |0;\mathbf{1};\mathbf{0};\pi(f), i, f(i),\mathbf{0}\rangle$
Here **0** denotes a sequence of zeros, and the zero eigenvalues of m_i $(i < 0)$ are not shown explicitly. And thus for each recursive function f and integers a and b as Deutsch explains, there exists a program $\pi(f, a, b)$ which computes the function f on the contents of a and places the result in b leaving a unchanged i.e.

$$|\pi(f, 2,3),i,j\rangle \longrightarrow \pi(f,2,3, i, j \oplus f(i)\rangle$$

where $\oplus$ is any associative and commutative operator with the properties $i \oplus i = 0$ and $i \oplus 0 = i$ which is the reversibility requirement so that the exclusive or function would do.

3.2 The Logical Grammar Appropriate to the Class of Quantum Computers

Thus not only does universal quantum computation prescribe Turing computability and determine the relationship between each program $\pi(f)$ and each recursive function f, but the requirement for reversibility imposes certain universal constraints on the domain of f through the space of all commutative and associative operators $\oplus$ for which it will be physically computable. One may therefore specify these universal constraints on the domains of all recursive functions f in terms the function F where respectively

$$F(a, b) = F(b,a) \quad \text{and} \quad F(a,F(b,c)) = F(F(a,b),c)$$

and where one subclass of F can be thought of as logical connectives so that the domain of all the recursive functions f is subject to a logical grammar where F(a,b) = F(b,a) must take the truth value true ie there is symmetry/reversibility, etc and where the Third Law of Thermodynamics requires that

F(a,a) = 0 i.e. must take the truth value false [5] [2]

[2] However I now know in 2007 that F(a,a) = 0 in the form of $a^2 = 0$ can mean $a \neq 0$ in the case where a is a nilpotent operator, and that this criterion corresponds to Pauli exclusion and a unique canonical computational solution.

Thus in quantum computation as required by the Third Law, there are no true decision processes, and so Turing computability is only simulated and all proof follows only from the falsification of some assertion It is noteworthy that this last relationship can be considered as a decision process, while the three relationships applied simultaneously define an equivalence class or classes.
F can also be thought of as the universal labelling function where a,b,c are sets or states in relation to the operator $\oplus$.

3.3 Gefwert's Methodology

Gefwert's paper "On the logical form of mathematical language" provides a formal philosophical investigation (of mathematical language) in accordance with the proposition as rules idea of Wittgenstein.
The explanation of defined concepts like real number, Euclidean space, etc. are handled adequately within mathematics. But the explanations of the explicit definitions themselves must be of different kind and the aim of Gefwert's paper is to contribute to a formal philosophical explanation accomplished by engaging in writing a logical depth grammar. This is done by extending Martin Lof's intuitionistic theory of types (1975) in accordance with Wittgenstein's principle 'the sense of a proposition can only be given once' i.e. the sense of it cannot be expressed except by repeating that proposition and therefore there is necessarily only one proposition for each fact that answers to it.
Thus to engage in writing a logical depth grammar is to show what makes it possible to produce a knowledge of facts. And this is the essence of a philosophical investigation or person programme, as Gefwert defines it. When a logical depth grammar is written, it distinguishes between a canonical syntax providing the essential explanation and the informal part (semantics) providing the informal, verbalized explanation where these are symptoms of there being a canonical form regulating their use. Thus symptoms provide sufficient conditions in order to engage in producing a knowledge of facts (e.g. by computation or measurement) whereas canonical forms are necessary conditions. (Comment. Here we see for the first time, through the words computation and measurement how the connection to Deutsch's paper is to be made i.e. a person program is just that, a program for a universal quantum computer or brain.)
And the relationship of computer programming to the intuitionistic theory of types has been explicitly dealt with by Martin Lof, where this theory itself is an extension of Russell's doctrine of types(1937) which says that every proposition $\Phi(x)$ .... has, in addition to its range of truth, a range of significance i.e. a range within which $\Phi(x)$ can be proposition at all, whether true or false. This is the first point in the theory of types. The second is that the ranges of significance form types i.e. if x belongs to a range of significance of $\Phi(x)$, then there is a class of objects, the type of x, all of which must also belong to the range of significance of $\Phi(x)$ however Φ may be varied; and the range of significance is always either of a single type, where every function and thus in particular every propositional function, will indeed have a type as its domain. This is almost verbatim the definition of the notion of set given by Bishop (1957). Very loosely therefore what is being said is that if x is in a range of significance, $\Phi(x)$ can be

computed and if it is not, then $\Phi(x)$ is non-computable where in the context of Deutsch's paper one now may have to distinguish between (reversible?) Turing computability i.e. syntax, a necessary condition, (stochastic) Turing computability (consequence?) and quantum computation (symptoms, semantics or sufficient condition?)
And so ranges of significance are defined in some sense by the kind of computational rules that are laid down for them, where an output can be computed, and Gefwert adopts the notation that if the assignment of arguments $x_1, x_2, ...x_k$ to variables $u_1, u_2 ....u_k$ of a functional expression and these correspond to an indication of what specific inputs have been fed into a computation

$$\frac{\bar{u}_1 = x_1, ...\bar{u}_k = x_k}{a(u_1,u_2.......u_k) = x}$$

then we may write the result of the computation as

$$a(u_1,......u_k) = x, \text{ for } u_1 = x_1 ....u_k = x_k$$

The critical aspect of Gefwert's argument then concerns the necessary distinction, not made in ordinary mathematics, between functions proper denoted by A,B,C . and functions as objects denotes by a,b,c, . which concern arbitrary type valued functional expressions and arbitrary object valued functional expressions respectively, which are vital to his analysis of the practice of mathematics with its distinction between informal verbalized language and the formal essential explanation of which the verbalized language is symptomatic. This distinction also exists in experimental practice between a physical process ie the function as object and its mathematical representation ie the function proper. Wittgenstein (1977) states ' The rules of grammar cannot be justified by showing that their application makes a representation agree with reality. For this justification would itself have to describe what is represented' Thus we must in the analyse of mathematical practice distinguish between a canonical part providing the essential (logical) explanation (syntax) and the informal part (semantics) providing the informal verbalized explanation (which concerns the knowledge of facts).
Thus it appears that what universal quantum computation provides as described by Deutsch (i.e.the Church Turing Principle where computation may perfect simulate any physical process) is indeed a representation/description for this justification, if we take quantum theory as our view of reality.
Thus there is, the Church Turing Principle implies a more general interpretation beyond that of logical connective functions, those of functions as objects in accord with Gefwert's argument. An example is the genetic code where this specifies the symbolic dynamics of the physical phenomena we call life. Another is the phenomena of strange attractors, where only certain aspect of their behaviour can be specified, i.e. the centres of attraction and the boundaries corresponding to those centres within which a particle will move if attracted to the particular centre in question. The general symbolic dynamics of such a phenomena as a whole, where the convention is to replace each centre of attraction by a symbol and represent the dynamics as an automata, is thus an

example, of the same notion which I believe we employ in natural language. That is, should an aspect of reality i.e. a physical process that we experience, be a symptom of that reality, then as a first step to constructing a logical model of that reality to convey its meaning is to introduce a symbol or label and then in defining its meaning we establish this by means of relationships to other symbols or labels already assigned for which our knowledge of facts is already established to some degree or other, and we check these specific facts to see to what extent they apply to our new symbol/label.[3]
For example the notions of set used in ordinary mathematics is really a mixture of the notions of both value and type i.e. E(u) and (Σ u ϵ A) E(u) respectively. Hence an object of the type (Σ u ϵ A) E(u) is an object x of the type which is the value A together with a proof y of the proposition which is of the value E(u) when u is assigned the value x. It is not enough to speak in the analysis of the object x just happening to satisfy E(u).
Using this critical distinction and notation, Gefwert (1986) then constructs arguments, not produced here, that beginning with x and $\Phi(x)$ that in order to engage in mathematical language i.e. the construction of proofs, it is necessary to have a logical depth grammar which requires sentences in the language of the following six forms only:-

$$a \equiv e,\ \ a = \mathrm{def}\ e,\ a = e\mathrm{Df}\ ;\ A \equiv E,\ A = \mathrm{def}E,\ A = E\mathrm{Df}$$

where the A valued function a and the E valued function e are definitionally equal or equivalent and the type valued functions A and B are definitionally equal or equivalent, respectively. Note that a language with these rules is the genetic code in molecular biology where we substitute the symbol A for adenine, the symbol U for uracil, ...

3.4 The Rules of Definitional Equality which Apply to the Logical Depth Grammar

Gefwert then proves that if $\underline{\bar{A}} = X$ is the value (denotation) of A i.e. A denotes X then he proves that there must be rules such that

$$\text{symmetry } \frac{\bar{A} = \bar{E}}{\bar{E} = \bar{A}}\ ;\ \text{transitivity}\ ;\ \frac{\bar{A} = \bar{E}\ ,\ \bar{E} = \bar{O}}{\bar{A} = \bar{O}}\ ;\ \text{reflexivity } \frac{\bar{A} \text{ is a type}}{\bar{A} = \bar{A}}$$

Where, for example, the first says. if we can compute $\bar{A}$ from $\bar{E}$ then we must be able to compute $\bar{E}$ from $\bar{A}$ using the same rule

[3] Thus meaning concerns these relationships but as shown now by our recent work in regard to the universal computational rewrite system in relation to quantum computation, there will always exist potentialities or emergent properties that link existing objects with others that may not currently be known or even exist.

4. Conclusion

Thus I conclude that the logical form of mathematical language is subject to a logical depth grammar, which I call 'the common sense' of mathematical sentence construction and that it these same rules of construction that quantum computation imposes on the domains of functionality of Turing computability, ie compare part 3.2 and 3.4, so that the semantics of mathematics are the same as syntax of quantum computation when it maps onto Turing computability.

References

1. Bennett C.H. and Landauer R. (1985) The fundamental physical limits of computation. Scientific American, July p38
2. Deutsch D (1985) Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer, Proceeding Royal Society .London A400,97-117.
3. Gefwert C.(1986) SLAC-PUB:-3625 Stanford University Linear Accelerator Center
4. Conway J.H. (1976) On Numbers and Games,' Academic Press
5. Marcer P.J. (1986) A solution to the problem of intelligence and human learning – the thermodynamics of computation - a basis for the ultimate theory of everything. Speculations in Science Technology. 9, 4, 247 -258
6. Martin Lof, P, (1975) An intuitionistic theory of types, Logic Colloquium '73 Rose and. Sheperdson (eds), Amsterdam 73-116.
7. Martin Lof, P, (1975) Constructive Mathematics and Computer programming, Cohen. Las, Pfieffer, Podewski (eds) Logic, Methodology and. Philosophy in Science.
8. Russell B.(1937) The Principles of Mathematics, London.
9. Bishop E. (1957) Foundations of Constructive Analysis, New York
10. Wittgenstein L. (1977) Philosophical Remarks, Oxford..