The Church-Turing Thesis*

Andrés Sicard-Ramírez
asicard@eafit.edu.co

Mario E. Vélez-Ruiz
mvelez@eafit.edu.co

Universidad EAFIT; Medellín, Colombia

October 11, 1999

1 Introduction

The Church-Turing thesis (CT-T) establishes a strong relation between effectively calculable (computable) functions and Turing computable (recursive) functions, actually, this is not just a relation, this is an equivalence between them. In section 2 we present a short history about the CT-T. We begin presenting Church's definition and Turing's definition for a computable function, then we mention that Kleene have named these definitions as Church's thesis and Turing's thesis and finally he decided name both of them as CT-T. We also introduce some the different philosophical interpretations allowed for CT-T. In section 3 we talk about the possibility of refuting CT-T in two ways: a weak refutation and strong one. After, we introduce some logic implications and some possible future considerations from paraconsistent logic point of view. In section 4 we introduce some misunderstandings in CT-T under perspective of what can be computed by a machine. Then we present a thesis stronger than CT-T, called Thesis M, so we can talk about Hypercomputation. Finally, in section 5 we work over CT-T from quantum computation perspective. We introduce other thesis stronger than CT-T called Church-Turing principle and some examples of possible relations between quantum computing and CT-T.

2 Historical and philosophical remarks

We believe that the Church-Turing thesis development has been: Church's definition, Turing's definition, Church's thesis, Turing's thesis and finally Church-Turing thesis. Obviously, the historical development has not been a linear history; its principal protagonists (Alonso Church, Kurt Gödel, Stephen Kleene, Jacques Herbrand, Emil Post, Alan Turing and others) exchanged letters, meetings, conversations and discussions. To present the CT-T in its current context

*The paper was financed by EAFIT University, under the research project number 817407.

we introduce a very simple history about it (we suggest papers [10, 12, 30, 36] for a detailed history about it).

In the decade of thirties, in contex of Hilbert's finitist program (in general) and looking for *Entscheidungsproblem*'s[1] solution (in particular), the mathematicians and logicians were looking for a formal meaning to informal notion of computable (calculable, effectively calculable, algorithmic) function of positive integers. In other words, they were looking for what is the meaning for the notion of effective or mechanical method in logic and mathematics.

From mathematical point of view as opposite to mechanical or algorithmic point of view, Church gave this definition for computable functions:

> **Church's definition:** "We now define the notion, already discussed, of an *effectively calculable* function of positive integers by identifying it with the notion of a recursive function of positive integers (or of a λ-definable function of positive integers). This definition is thought to be justified by considerations which follow, so far as positive justification can ever be obtained for the selection of a formal definition to correspond to intuitive notion." [7, p. 356]

From mechanical and algorithmic point of view, Turing built a Turing machine (called by him as automatic machine), he defined a function Turing computable if it can be calculated by a Turing machine, he also gave the following definition for computable functions[2]:

> **Turing's definition:** "The "computable" numbers[3] include all numbers which would naturally be regarded as computable." [39, p. 249]

Thanks to the agreement obtained between people who is working on the problem and the empirical evidence obtained [18], it means, *"(1) Every effectively calculable function that has been investigated in this respect has turned out to be computable by Turing machine; (2) All known methods or operations for obtaining new effectively calculable functions from given effectively calculable functions are paralleled by methods for constructing new Turing machines from given Turing machines; (3) All attempts to give an exact analysis of the intuitive notion of an effectively calculable function have turned out to be equivalent in the sense that each analysis offered has been proved to pick out the same class of functions, namely those that are computable by Turing machine."* [10, p. 4]; we think that Church's definition and Turing's definition earned its thesis's status:

> **Church's thesis and Turing's thesis:** "The thesis of Church and Turing were not even called "thesis" at all until Kleene (1943, p. 60)[4] referred

[1] "By the *Entscheidungsproblem* of a system of symbolic logic is here understood the problem to find an effective method by which, given any expression Q in the notation of the system, it can be determined whether or not Q is provable in the system." [6, p. 41]

[2] In reality, Turing paper's subject [39] was computable numbers, although his works is directly extended for computable function.

[3] The number whose decimal representation can be generating progressively by a Turing machine. [10, p. 9]

[4] Kleene, S.C. 1943. Recursive predicates and quantifiers. Transactions of the American Mathematical Society, vol 53. p. 60)

to Church's "definition" as "Thesis I" and the 1952 Kleene[5] referred to "Church's Thesis" and "Turing's Thesis"." [36, pp. 295-296]

Becoming a *definition* in a *thesis* implies a philosophical jump. This jump stimulates different interpretations for the Church's thesis and Turing's thesis (for example, Ramos [26] introduce Nelson's naturalistic interpretation [20], Shapiro's structuralist interpretation [38] and Schanker's conventionalist interpretation [27]). Every possible interpretation is supported by an ontological position about mathematical objects and its relation with reality. On the same direction, some people instead of giving the name of 'thesis' to the concept prefer give names as 'hypothesis', 'working hypothesis', or 'principle' to refer a Church's or Turing's thesis[6]; in some cases, this names has the same meaning but in other cases, is not in this way.

Without entering in the question about the ontology dimension of mathematical objects (because this is outside of limits of this paper) and independently of the matter of considering Church's and Turing's thesis as definitions (in any philosophical sense), or as empirical or mathematics thesis, we think that Church's thesis and Turing's thesis should have associated a true-value. This true-value is not known, and although is not possible a positive test, we thought that if should be possible a negative one.

On the other hand and because of the demonstrations about coexistence between λ-definable functions set, Turing computable functions set and recursive functions set, Church's thesis and Turing's thesis are presented usually as the Church-Turing thesis:

> **Church-Turing thesis:** "The term 'Church-Turing thesis' seems to have been first introduce by Kleene, with a small flourish of bias in favor of Church" [10, pp. 3-4]:
>
> > 'So Turing's and Church's thesis are equivalent. We shall usually refer to them both as *Church's thesis*, or in connection with that one of its ... version which deal with 'Turing machines' as *the Church-Turing thesis*.' (Kleene 1967: 232.)[7]

About the using of the name 'Church-Turing thesis' Soare says:

> "Here we also use the phrase "*Church-Turing thesis*" to refer to the amalgamation of the two theses (these and others)[8] where we identify all informal concepts of Definition 1.1[9] with one another we identify all the formal concepts of Definition 1.2[10], and their mathematical equivalents, with one another and suppress their intensional meanings." [36, p. 296]

[5] Kleene, S.C. Introduction to metamathematics. Wolters-Noordhoff Publishing. Groningen, 1952.

[6] This situation is the same in relation with the name 'Church-Turing thesis'.

[7] Kleene, S.C. 1967. Mathematical Logic. New York: Wiley.

[8] Church's thesis and Turing's thesis.

[9] Definition 1.1: A function is "*computable*" (also called "*effectively calculable*" or simply "*calculable*") if it can be calculated by a finite mechanical procedure. [36, p. 284]

[10] Definition 1.2: (i) A function is "*Turing computable*" if it is definable by a Turing machine, as defined by Turing 1936. [36, p. 285]

According to Soare explanation, the name 'Church-Turing thesis' suppress the motivation and the intuition behind the Church's thesis and the Turing's thesis. The justification to synthesize Church's thesis and Turing's thesis in Church-Turing thesis is because they have the same formal meanings, it means, the Turing computable functions and the recursive (λ-definable) functions are equivalents in the formal dimension.

The introduced situation between Church's thesis and Turing's thesis is not only situation in which, two definitions are formally equivalents but they have different intensional meanings. In the beginning of quantum mechanics formulations, there was two definitions with this characteristics: Werner Heisenber's matrix mechanics and Erwing Scrödinger's wave mechanics[11].

3 Possible refutations for Church-Turing thesis

Nowadays, agreement exists between almost all logicians and mathematicians that the Church Turing thesis is correct and even though almost all of them agree with the impossibility of proving the CT-T [13, 19, 36] because it establishes a relation between a formal concept (Turing computable functions, general recursive functions) and an intuitive concept (effectively calculable function); some persons believe that it is possible of refutation. If we think about CT-T as:

> A function f is effectively calculable iff function f is Turing computable;

we have two possible ways for refutation. One possibility for refutation is to eliminate equivalence in one direction, it means:

> If a function f is Turing computable do not imply that function f is effectively calculable.

First possibility of refutation means that we find a Turing computable function which is not effectively computable, it means that, formal notion of Turing computable functions is wider than intuitive notion of computable functions [19, p. 201]. We can call this way of refutation as, *the weak refutation* because from definition for Turing computable functions it seems evident that they are effectively calculable functions. Nevertheless, there are Turing computable functions that we do not believe it is very clear that they are effectively calculable functions.

[11] In classic mechanics in general and wave mechanics in particular, is frequently recognize set up modes of vibration on material objects (for example, modes of vibration for a rope tied in the ends or modes of vibration produced by the sound in a resonant cavity) and idea by Louis de Broglie of pilot wave associated all matter in motion, they made to germinate in Erwing Scrödinger toward 1923 the idea of a quantum mechanics in the who the important was that the electrons set up harmonics modes of vibration in the interior of the atoms. Toward 1926 in hands of Werner Heisenber and his colleagues, appear another quantum mechanics whit irreconcilable view with previous, for the who the truly important it was the novelty idea of the quantum jumps and of the discontinuities in the spectrum of some observables of the atoms, with an added difficulty additional, the impossibility of make to intuition about images of the atoms [4, 9]

For example, the Ackermann's function is a Turing computable function [17, 18][12], but its grows very fast. $A(4,2)$ has $19,728$ digits [35], so ¿it is possible calculate, $A(4,3)$? for instance. Let's see Harry Smith's answer:

> "I agree that the Ackermann function is "A function to end all functions". You asked if I tried A(4,3). Well yes, but my program VPCalc only handles a value up to about $10^{15,032,385,525}$ and can only handle $114,639$ decimal digits in the mantissa. A(4,2) fits my niche well, but A(4,3) is way out there. It would take more than A(4,2) seconds to compute A(4,3) and the computer would need more than A(4,2) bits of memory. This cannot be done in this universe. Remember $A(4,2) = 2.00352...E + 19728$." [35].

On the same way, Mendelson introduce the following example:

> "Porte [13] has a proved an interesting theorem which has, as a result, there are general recursive function $f(z)$[14] such that, for any general recursive function $g(x)$, there exist infinitely many numbers x in the range of f such that, for any argument z_0 with $f(z_0) = x$, the number of steps[15] necessary to compute $f(z_0)$ exceeds $g(x)$. Now, if g grows very fast, say $g(x) = 100^{100^{100^{100^x}}}$, then it will be impossible to carry out the computation of $f(z_0)$ within the life-span of a human being or probably within the life-span of the human race." [19, p. 202]

Then, there are some Turing-computable functions impossible to calculate in this universe. However, there is agreement in academic community of that these function are effectively calculable functions, in Mendelson's words:

> "A function is considered effectively computable if its value can be computed in an effective way in a finite number of steps, but there is no bound on the number of steps required for any given computation. Thus, the fact that there are effectively computable functions which may not be humanly computable has nothing to do with Church's thesis." [19, p. 202]

In general, we can say that problem of number of steps, there is not problem for computability, there is a problem for an area called computational complexity [16, 24].

Another possibility for refutation is to eliminate the equivalence in the other direction, it means:

> If a function f is effectively calculable do not imply that function f is Turing computable.

---

[12]This function is defined as:

$$\begin{aligned} A(0, y) &= y + 1, \\ A(x + 1, 0) &= A(x, 1), \\ A(x + 1, y + 1) &= A(x, A(x + 1, y)). \end{aligned}$$

[13]Porte, J. Quelques pseudo-paradoxes de la "calculabilité effetive," Actes du 2^{me} Congrès International de Cybernetique, Namur, Belgium, 1960, pp. 332-334

[14]Namely, any general recursive function with non-recursive range.

[15]Say, the number of steps in the calculation of the values of f from a system of equations for the computation of f.

Second refutation means that we can find an effectively computable function which can be showed as not a Turing computable function, it means, formal notion of Turing computable functions is narrow than intuitive notion of computable functions [19, p. 201]. We can call this way of refutation as, *the strong refutation* because from definition for Turing computable functions it does not seem evident that they are all effectively calculable functions.

The fact of finding an effectively computable function that does not be Turing computable function will have very important results in the current logic. We hereby present some extracts from the letter sent by Alonso Church to Jósez Pepis, in relation to Pepis' implicit proposal of a powerful definition for computability:

> "Therefore to discover a function which was effectively calculable but no general recursive would imply discovery of an utterly new principle of logic, not only never before formulated, but never before actually used in a mathematical proof - since all extant mathematics is formalizable within the system of Principia, or at least within one of its known extensions. Moreover this new principle of logic must be of so strange, and presumably complicated, a kind that its metamathematical expression as a rule of inference was not general recursive, for this reason, if such a proposal of a new principle of logic were ever actually made, I should be inclined to scrutinize the alleged effective applicability of the principle with considerable care." [30, pp. 175-176]

Church's answers is based over relation between effectively calculable functions and theorems for a system of symbolic logic. We can see this relation as:

> "A function $y = f(x)$ of the natural numbers is said to be representable in a formal system including the arithmetic, if there is a formula $\phi(x, y)$ in that system such that for any natural numbers m, n and the symbols μ, ν in that system representing m, n respectively, if $m = f(n)$ then the formula $\phi(\mu, \nu)$ is provable and if $m \neq f(n)$ then the negation of $\phi(\mu, \nu)$ is provable. Then, it is well-known that if the system is consistent, the class of representable functions coincides with the class of recursive functions." [22, p. 6]

In this case, it is clear Church's mention of a *new principle of logic*. However, we think that relation established (between effectively calculable functions and theorems of logic) is supported in the consistency of the system of logic used. Just like a working hypothesis, we believe that a good possibility for working on *the strong refutation* is not using a consistent system of logic, but a paraconsistent system of logic [3, 11, 25]. We think about a decidable inconsistent first order predicate calculus in which we can express and solve the *halting problem*[16]

[16] We can think the halting problem as: Let $\mathcal{MT}$ be a Turing machine and let α an input for machine $\mathcal{MT}$. Is there a Turing machine $\mathcal{H}$, such a:

$$\mathcal{H}(\mathcal{MT}, \alpha) = \begin{cases} 1 & \text{iff } \mathcal{MT} \text{ halt with } \alpha, \\ 0 & \text{iff } \mathcal{MT} \text{ do not halt with } \alpha. \end{cases}$$

From Turing's paper [39], do not exist Turing machine $\mathcal{H}$.

for example, nevertheless for paraconsistent first order predicate calculus known at the moment, they are undecidable calculus[17].

4 Thesis M: Misunderstanding of Church-Turing thesis

The Church-Turing thesis is a formal-theoretical thesis (as opposite to concrete-physical thesis). However, as Jack Copeland point out [10], there are some misunderstandings of CT-T because some people relation it with the limits of what can be computed by a machine[18].

We believe that this misunderstanding is because Turing used the term *computer* for his Turing machines in his initial paper [39]. Nevertheless, how some people have marked, Turing used the term *computer* for an *"idealized human calculating in a purely mechanical fashions"* [36, p. 292]. Following to Soare[19] we use the term *computor* for an idealized human calculating and we use the term *computer* for a machine (theoretical or physical). So, the Church-Turing thesis is about *computors* and not about *computers.* Copeland introduces this difference as:

> "Gandy (1980)[20] is one the few writers to distinguish explicitly between Turing's thesis and the stronger proposition that whatever can be calculated by a machine can be calculated by a Turing machine. Borrowing Gandy's terminology, I will call the stronger proposition 'Thesis M'.
>
> > **Thesis M:** Whatever can be calculated by a machine (working on finite data in accordance with a finite program of instructions) is Turing machine computable.
>
> Thesis M itself admits of two interpretations, according to whether the phrase 'can be calculated by a machine' is taken in the narrow sense of 'can be calculated by a machine that conforms to the physical laws (if not to the resource constrains) of the actual world', or in a wide sense that abstracts from the issue of whether or not the notional machine in question could exist in the actual world. The narrow version of the thesis M is an empirical proposition whose truth-value is unknown. The wide version of the thesis M is known to be false. Various notional machines have been described which can calculate functions that are not Turing machine computable ... [21]." [10, pp. 5-6]

---

[17] This was affirmed by Graham Priest and Chris Mortensen, in private electronic mail.

[18] Copeland does to mention about other misunderstandings of Chruch-Turing thesis in areas as the computational theory of the mind or in cases as process that can be scientifically explicable [10].

[19] Soare follows Robin Gandy (Gandy, R. Church's thesis and principles for mechanisms. The Kleene symposium, North-Holland, pp. 123 - 148) and Wilfred Sieg (Sieg W. Mechanical procedures and mathematical experience. Mathematics and mind (A. George, editor), Oxford University Press. 1994.) [36, p. 291-292]

[20] Gandy, R. Church's thesis and principles for mechanisms. The Kleene symposium, North-Holland, 1980, pg. 123-148)

[21] In this point, Copeland indicates some papers that describe this machines.

Models used to demonstrate that wide version of the thesis M is false are usually called hipercomputation models. This models are continuous models or infinity models for computation (as opposite to discrete and finite models used for narrow version of the thesis M). To build a hipercomputation $\mathcal{ST}$ model, usually first is constructed a model T equivalent for Turing machine and under some modifications, it comes to be a hipercomputation model. To demonstrate that model ST is a hipercomputation model there are two ways: (i) To demonstrate that ST is equivalent to other hipercomputation model well known or (ii) To demonstrate that ST can solve a problem that Turing machine can not (for example, the *halting problem*).

Nowadays, there are hipercomputation models from different theories as indicated in following examples. From neural networks theory, Hava Siegelmann and Eduardo Sontag [31, 33, 34] introduce a neural network model called *Analog Recurrent Neural Network (ARRN)*, if neurons's weights are rational numbers then ARRN is equivalent to Turing machine, but if neurons's weights are real numbers then ARRN is a hipercompuation model. From dynamic system theory, Hava Siegelman [32] introduces a hypercomputation model called *Analog Shift Map*. Mike Stannett [37] from *X-machine* model introduces a hipercomputation model called *Analogous X-Machine (AXM)*.

On the other hand, models used for working with narrow version of the thesis M are discrete and finite models. Turing required that computable function were calculable by finite means [39, p. 231]. Finite means for Turing machines means: (A) Turing machine's components are finite: (A1) A Turing machine has finite states, (A2) A Turing machine has finite alphabet; and (B) Turing machine's evolution is finite and local: (B1) Every evolution's step must depends only on the current state and the symbol in the observed cell, (B2) A Turing machine can change one and only one symbol from the tape, in every evolution's step and (B3) Turing machine can move only one cell to left or to right from the observed cell, in every evolution's step.

We believe that discrete components, finitess and locality are three necessary conditions to think about the narrow version for thesis M. In this context, we present the Gandy's result[22]:

> "Gandy's main result is that what can be calculated by a discrete deterministic mechanical device is Turing computable." [36, p. 296]

> "One of the arguments more solid to in favor of the Church's thesis has been proposed by R. Gandy, who had demonstrated that the mechanisms of all the machines building under newtonian mechanics could calculate only programmable functions[23]." [13, p. 81]

At this moment, we consider important to say that the digital computers was consider for Turing as *computors*, it means, digital computers satisfies the Church-Turing thesis. In Turing's words:

[22]Gandy, R. Church's thesis and principles for mechanisms. The Kleene symposium, North-Holland, 1980, pp. 123-148)

[23]We can think about programmable functions as Turing computable functions.

> “ ‘The idea behind digital computers may be explained by saying that these machine are intended to carry out any operations which could be done by human computer.’ (Turing 1950:436)[24].” [10, p. 9]

Some people can object us that current digital computers are not mechanical, because its is electrical. Nevertheless, Turing did not considered this important:

> “It gives up frequent importance to the fact that current digital computers are electrical .... It like Babbage’s machine was not electrical and like sure sense all digital computers are equivalent, we see that this use of the electricity could not be of theoretical importance.” [40, p. 21-22], [41, p. 41]

5 Church-Turing principle: Physical version for Church-Turing thesis

For Gandy’s result one possibility to work with the narrow version for thesis M is quantum computing theory. Although, quantum physics works naturally with continuous objects, current models for quantum computing should be regarded as a discrete models for computation [2]. Some models for quantum computation are: (i) Quantum circuits [1, 15, 28] and (ii) Quantum Turing machines [2, 14, 23, 29].

We consider David Deutsch’s paper “Quantum theory, the Church-Turing principle and the universal quantum computer” as a principal paper about quantum computing. In this paper, Deutsch introduces a personal version for the Church-Turing thesis called by him, Church-Turing hypothesis:

> **Church-Turing hypothesis:** “Every ‘function which would naturally be regarded as computable’ can be computed by the universal Turing machine.” [14, p. 99]

Deutsch proposes to reinterpret ‘function which naturally be regarded as computable’ as ‘function which may in principle be computed a real physical system’; he also proposes to generalize ‘universal Turing machine’ as a ‘universal model computing machine operating by finite means’ and he establishes that ‘finitely realizible physical systems’ must include any physical object upon any experimentation is possible. Then, he presents his physical version of Church-Turing hypothesis called by him, Church-Turing principle:

> **Church-Turing principle:** “Every finitely realizable physical system can be perfectly simulated by a universal model computing machine operation by finite means.” [14, p. 99]

Deutsch introduces a general quantum Turing machine, this general quantum Turing machine can be thought as universal quantum Turing machine with following considerations: Instructions set for a Turing machine determine its

[24] Turing, A.M. 1950. Computing Machinery and Intelligence. Mind 59, 433-460 (some Spanish version are [40] and [41]).

behavior, the same as, temporal evolution operator for a quantum Turing machine determines its behavior, then, an adequate [instructions set - temporal evolution operator] transforms a [Turing machine - quantum Turing machine] in a [universal Turing machine - universal quantum Turing machine]. Deutsch establishes that universal quantum Turing machine satisfies the Church-Turing principle:

> "... I present a general, fully quantum model form computation. I then describe the universal quantum computer $\mathcal{Q}$, which is capable of perfectly simulating every finite, realizible physical system. ... In computing strict functions from $\mathbb{Z}$ to $\mathbb{Z}$ it generates precisely the classical recursive functions $C(\mathcal{F})$ [25]." [14, p. 102]

Discrete components, finitess, and locality conditions for quantum Turing machines are given by: (A) quantum Turing machine's components are finite: (A1) quantum Turing machine has finite processor, (A2) quantum Turing machine has finite alphabet; and (B) quantum Turing machine's evolution is finite and local: "*(i) only a finite subsystem is in motion during any one step, and (ii) the motion depends only on the state of a finite subsystem, and (iii) the rule that specifies that motion can be given finitely in the mathematical sense.*" [14, p. 100]

Although quantum Turing machines can compute only Turing computable functions, Deutsch claims that Church-Turing principle is stronger than Church-Turing thesis:

> "The statement of the Church-Turing principle is stronger than what is strictly necessitated by (1.1)[26]. Indeed it is so strong that is *not* satisfied by Turing's machine in classical physics. Owing to the continuity of classical dynamics, the possible states of a classical system necessarily form a continuum. Yet there are only countable many ways of preparing a finite input for $\mathcal{F}$[27]. Consequently $\mathcal{F}$ can not perfectly simulate any classical dynamics system." [14, p. 100]

Because the Church-Turing principle is stronger than Church-Turing thesis and the quantum Turing machine satisfies this principle, should be one problem that is quantum Turing computable but its not Turing machine. One problem belong to this category is problem of generate truly random numbers. In relation to randomness and classical computers, Colin Williams and Scott Cleartwater say:

> "Although most modern programming languages contain some kind of command for generating a "random" number, in reality, they can only generate "pseudorandom" numbers. These are sequences of numbers that pass many of the test that a sequence of random numbers would also pass. But they are not true random numbers, because they are merely the completely predictable output from a definite function evaluation." [42, p. 155]

[25] $\mathcal{F}$ is Turing's universal computing machine and $C(\mathcal{F})$ is the Turing computable functions set.

[26] The Church-Turing hypothesis.

[27] $\mathcal{F}$ is Turing's universal computing machine.

Generation for random numbers is intrinsic for quantum computing. Deutsch does it explicit by:

> " Unlike $\mathcal{F}^{28}$, it[29] can simulate any finite classical discrete stochastic process perfectly." [14, p. 102]

This problem is very special because if we accept to generate random numbers per function, we have pseudorandom numbers. Quantum Turing machines allows to generate random numbers without the explicit use of a function. [30]

On the other hand, we have indicated some ideas of thinking about Church-Turing thesis from quantum computing point of view. Conversely, we now indicate some way of thinking about quantum physics from Church-Turing thesis point of view. We want to present M. A. Nielsen and Norman Bridge arguments about limitations for quantum observables, due to Church-Turing thesis.

Paper's abstract by Nielsen present directly the question:

> "We construct quantum mechanical observables and unitary operators which, if implemented in physical systems as measurements and dynamical evolution, would contradict the Church-Turing thesis which lies at the foundation of computer science. We conclude that either the Church-Turing thesis needs revision, or that only restricted classes of observables may be realized, in principle, as measurements, and that only restricted classes of unitary operators may be realized, in principle, as dynamics." [21, p. 1]

Initially, Nielsen defines a halting observable $\hat{h}$ using the halting function $h(x)$.[31] Agree with quantum mechanics definition, *"the Hermitian operator A is an observable if this orthonormal system of vectors forms a basis in the state space."* [8, p. 137]. The halting observable is a Hermitian operator on the state space of the system, then it is a quantum observable. If $\hat{h}$ is a quantum observable, its measure has two possibilities [21, p. 2]:

[28] $\mathcal{F}$ is Turing's universal computing machine.

[29] Universal quantum computer.

[30] We can see how generate a random number from quantum computing. Let $|\, x\rangle$ be a qubit in $|\, 0\rangle$ state and let H be Hadamard operator definite by:

$$H\,|\,0\rangle \to \frac{1}{\sqrt{2}}(|\,0\rangle + |\,1\rangle),$$
$$H\,|\,1\rangle \to \frac{1}{\sqrt{2}}(|\,0\rangle + |\,1\rangle).$$

Then, we apply Hadamard operator for qubit $|\,0\rangle$ and we obtain superposition state $\frac{1}{\sqrt{2}}\,|\,0\rangle + \frac{1}{\sqrt{2}}\,|\,1\rangle$. After, we observe the superposition state. *"The act of observation causes the superposition to collapse into either $|\,0\rangle$ or the $|\,1\rangle$ state with equal probability. Hence you can exploit quantum mechanical superposition and indeterminism to simulate a perfectly fair coin toss."* [42, p. 160]

[31] Halting observable $\hat{h}$ and function $h(x)$ are defined as [21, pp. 2-3]:

$$\hat{h} \equiv \sum_{x=0}^{\infty} h(x)|x\rangle\langle x| \quad \text{and} \quad h(x) \equiv \begin{cases} 1 & \text{if program } x \text{ halts on input } x; \\ 0 & \text{if program } x \text{ does not halt on input } x. \end{cases} \tag{1}$$

1. It is possible, in principle, to construct a measuring device capable of performing a measurement of the observable $\hat{h}$.
2. It is not possible, in principle, to construct a measuring device capable of performing a measurement of the observable $\hat{h}$.

Nielsen's analysis for this two possibilities is:

> "Suppose the first possibility is true. Then in order to compute the value of $h(x)$ one performs the following procedure: Construct the measuring apparatus to measure $\hat{h}$, and prepare the system to be measured in the state $|x\rangle$. Now perform the measurement. With probability one the result of the measurement will be $h(x)$. This gives a procedure for computing the halting function. If one accepts the Church-Turing thesis this is not an acceptable conclusion, since the halting function is not computable.
>
> Acceptance of the Church-Turing thesis therefore forces us to conclude that the second option is true, namely, that it is not possible, in principle, to construct a measuring device capable of performing a measurement of the observable $\hat{h}$. That is, only a limited class of observables correspond to measurements which may be performed, in principle, on quantum mechanical systems." [21, p. 2]

Then, Nielsen makes a similar analysis to perform an approximated measurement of $\hat{h}$ and make a similar analysis for the physical realization of unitary operators as dynamical evolution (under infinite and finite dimensional state space). In accordance with previous, there are two possibilities: (i) Modification of the Church-Turing thesis or (ii) Rethinking observables and evolution operators for realizible quantum system. Nielsen opinion is:

> "It is the author's conjecture that the Church-Turing thesis is essentially correct, and that a more satisfactory program is to address the problem of achieving a sharp characterization of the class of observables and unitary dynamics which may be realized in physical systems." [21, p. 3]

Although Masanao Ozawa [22] refutes Nielsen arguments satisfactorily[32] we want to emphasize principal idea behind Nielsen work for us. In this case, they used the Church-Turing thesis for think once again about some aspects of quantum computing in particular and quantum physics in general.

6 Final observation

Although we have mentioned some relations between (i) Church-Turing Thesis, (ii) Thesis M in narrow sense, (iii) Thesis M in wider sense and (iv) Church-Turing principle; we think it is necessary to establish more relations between (i),

[32] "It can be pointed out that Nielsen's formulation of the measurability of observables does not respect sufficiently the fact that every measuring apparatus has only finite precision as long as it can be constructed in a laboratory or it is finitely realizible. . . . . Under this formulation, it is proved that the halting observable is measurable, contrary to Nielsen's argument, without any contradiction with the Church-Turing thesis." [22, p. 2]

(ii), (iii) and (iv). This way of working allow to us a better comprehension of computability problem which has the simplicity and complexity we hope every good problems.

References

[1] Adriano Barenco, Charles H. Bennett, Richard Cleve David P. DiVincenzo, Norman Margolus Peter Shor Tycho Sleator, John Smolin, and Harald Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A **52** (1995), no. 5, 3457–3467.

[2] Ethan Bernstein and Umesh Vazirani, *Quantum complexity theory*, SIAM J. Comput. **26** (1997), no. 5, 1411–1473.

[3] M. Andrés Bobenrieth, *¿Inconsistencias, por qué no?*, Santafé de Bogotá: Tercer Mundo Editores, División Gráfica, 1996.

[4] David C. Cassidy, *Heisenber, uncertainty and the quantum revolution*, Scientific American (1992), 106–112.

[5] Alonso Church, *Correction to a note on the Entscheidungsproblem*, The Journal of Symbolic Logic **1** (1936), no. 3, 101–102.

[6] ———, *A note on the Entscheidungsproblem*, The Journal of Symbolic Logic **1** (1936), no. 1, 40–41, Correction [5].

[7] ———, *An unsolvable problem of elementary number theory*, American Journal of Mathematics **58** (1936), no. 2, 345–363.

[8] Claude Cohen-Tannoudji, Bernard Diu, and Franck Laloë, *Quantum mechanics*, vol. 1, Hermann and John Wiley and Sons, Inc., 1997.

[9] François Gieres, *Dirac's formalism and mathematical surprises in quantum mechanics*, Preprint: `arxiv.org/abs/quant-ph/9907069v1`, 1999.

[10] B. Jack Copeland, *The Church-Turing thesis*, Stanford Encyclopedia of Philosophy, 1996.

[11] Newton C. A. da Costa and Renato A. Lewin, *Lógica paraconsistente*, Enciclopedia IberoAmericana de Filosofía, vol. 7: Lógica, Editorial Trotta, S.A., 1995, pp. 185–204.

[12] Martin Davis, *Why Gödel didn't have Church's thesis*, Information and Control **54** (1982), no. 1/2, 3–24.

[13] Paul Delahaye, *Creaciones informáticas: un juego universal de herramientas de cálculo*, Investigación y Ciencia (1996), 80–83.

[14] David Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proc. R. Soc. Lond. A **400** (1985), 97–117.

[15] ———, *Quantum computational networks*, Proc. R. Soc. Lond. A **425** (1989), 73–90.

[16] Michael R. Garey and David S. Johnson, *Computers and intractability. A guide to the theory of NP-completeness*, W. H. Freeman and Company, 1979.

[17] Hans Hermes, *Enumerability · Decidability · Computability*, Berlin: Springer-Verlag, 1969.

[18] Stephen C. Kleene, *Introducción a la metamatemática*, Colección: Estructura y Función, Madrid: Editorial Tecnos, 1974.

[19] Elliot Mendelson, *On some recent criticism of Church's thesis*, Notre Dame Journal of Formal Logic **IV** (1963), no. 3, 201–205.

[20] R. J. Nelson, *Church's thesis and congnitive science*, Notre Dame Journal of Formal Logic **28** (1987), no. 4, 581–609.

[21] Michael A. Nielsen, *Computable functions, quantum measurements and quantum dynamics*, Phys. Rev. Lett. **79** (1997), no. 15, 2915–2918.

[22] Masanao Ozawa, *Measurability and computability*, Eprint: `arXiv.org/abs/quant-ph/9809048`, 1998.

[23] Masanao Ozawa and Haramichi Nishimura, *Local transition function of quantum Turing machines*, Eprint: `arXiv.org/abs/quant-ph/9811069`, 1999.

[24] Christos Papadimitriou, *Computational complexity*, Reading: Adisson-Wesley Publishing Company, 1994.

[25] Lorenzo Peña, *Introducción a las lógicas no clásicas*, Instituto de Investigaciones Filosóficas. Colección: Cuadernos, México D.F.: U. Autónoma de México, 1993.

[26] Jaime Ramos, *Sobre la naturaleza de la tesis de Church*, Ideas y Valores; U. Nacional, Santafé de Bogotá-Colombia **92-93** (1993), 157–167.

[27] S. G. Schanker, *Wittgenstein vs. Turing on the nature of Church's thesis*, Notre Dame Journal of Formal Logic **28** (1987), no. 4, 615–649.

[28] Andrés Sicard and Mario Vélez, *Algunos elementos introductorios acerca de la computación cuántica*, 1999, VII Encuentro ERM. U. de Antioquia, Medellín. Agosto 23-27.

[29] ———, *Some relations between quantum Turing machines and Turing machines*, (Draft), 1999.

[30] Wilfred Sieg, *Step by recursive step: Church's analysis of effective calculability*, The Bulletin of Symbolic Logic **3** (1997), no. 2, 154–180.

[31] Hava T. Siegelmann, *Computation beyond the Turing limit*, Science **268** (1995), 545–548.

[32] ______, *The simple dynamics of super Turing theories*, Theoretical Computer Science **168** (1996), no. 2, 461–472.

[33] Hava T. Siegelmann and Eduardo D. Sontag, *On the computational power of neural nets*, Proc. Fifth ACM Workshop on Computational Learning Theory, Pittsburg, July 1992, 1992.

[34] ______, *Analog computation via neural networks*, Theoretical Computer Science **131** (1994), no. 2, 331–360.

[35] Harry Smith, *The Modulator forum: letters to the editor*, The ModulaTor Technical Publication **11** (1991), Eprint: `www.modulaware.com/mdlt17.htm` [15-Feb-1999].

[36] Robert I. Soare, *Computability and recursion*, The Bulletin of Symbolic Logic **2** (1996), no. 3, 284–321.

[37] Mike Stannett, *X-machines and the halting problem: Building a super-Turing machine*, Formal Aspects of Computing **2** (1990), 331–341.

[38] Shapiro Stewart, *Understanding Church's thesis*, J. Philos. Logic **10** (1981), 353–365.

[39] Alan M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, Proc. London Math. Soc. **42** (1936-7), 230–265, A correction, ibid, vol. 43, no. 2198, p. 544–546, 1937.

[40] ______, *Maquinaria computadora e inteligente*, Controversias sobre mentes y máquinas (Alan Ross, ed.), Muy Interesante, vol. 32, Editorial Orbis S.A., 1986, pp. 13–52.

[41] ______, *¿Puede pensar una máquina?*, SIGMA. El mundo de las matemáticas (James R. Newman, ed.), vol. 6, Editorial Grijalbo, 1994, pp. 36–60.

[42] Colin P. Williams and Scott H. Clearwater, *Explorations in quantum computing*, Springer-Telos, 1997.