Fuzzy time and possible impacts of it on science [13]

Theory of Fuzzy Time Computation(TC^* vs TC & TQC)

Farzad Didehvar

didehvar@aut.ac.ir

Amir Kabir University of Technology (Tehran Polytechnic)

January 1 2023

Abstract. One of the possible hypotheses about time is to consider any instant of time as fuzzy number, so that two instants of time could be overlapped. Historically, some Mathematicians and Philosophers have had similar ideas like Brouwer and Husserl [5].

Throughout this article, the impact of this change on Theory of Computation and Complexity Theory are studied. In order to rebuild Theory of Computation in a more successful and productive approach to solve some major problems in Complexity Theory, the present research is done. This novel theory is called here, the fuzzy time theory of computation, TC^*. Here, we show the situation of some major problems in Complexity theory and Quantum complexity theory in the new model, and how some major problems would be solved.

Keywords. $\mathrm{P} \neq \mathrm{NP}$, $\mathrm{P} = \mathrm{PBB}$, $\mathrm{MA} = \mathrm{AM}$, , $\mathrm{PH} \subsetneq \mathrm{PSPACE}$, QIP=PSPCE, $\mathrm{QMA}^* = \mathrm{MA}^* = \mathrm{AM}^* = \mathrm{QAM}^*(= \mathrm{QMA} = \mathrm{QAM})$ **,**Fuzzy Time, TC^*, Reducibility, Complexity Theory Problems, Fuzzy time particle interpretation of quantum mechanics

1. Introduction

Throughout this article, the author presents the Theory of Computation by applying Fuzzy Time. More specifically, the author tries to rebuild the structure of the Theory of computation based on considering time as a fuzzy concept.

In fact, there are reasons to believe time as a fuzzy concept. More precisely, here, given the classical definition of Turing Machine, the concept of Time is changed to be Fuzzy. This new theory is called Theory TC^* and this type of computation “Fuzzy time Computation”. We have relatively large number of fundamental unsolved problems in Complexity Theory. In the new theory, some of the major obstacles and unsolved problems have been solved. It should be noted that in this article, the author considers fuzzy number associated to instants of time as a symmetric one. The point is about applying the symmetricity of fuzzy time function in the proof of Lemma 3.

In particular, the new classes of complexity Theory, P^*, NP^*, BPP^* in the TC^* are defined similar to the definitions of P, NP and BPP as their natural alternative definition. Here, firstly we will see, $\mathrm{P}^* = \mathrm{BPP}^*,\ \mathrm{MA}^* = \mathrm{AM}^*$. Later we have some results about P vs NP problem and PH vs PSPACE problem in the new model. Finally, we discuss about Quantum Complexity classes.

Considering instants of time as fuzzy concept has some excuses which we list it as following

1. Brower and Husserl had similar ideas [5]. Also, some physicists introduced the operator of time.
2. There is no reason right now, to prefer Classical time
   To a model of time which fuzzy time function is a narrow function.
3. By a new interpretation of Quantum Mechanics we are able to compute the fuzzy instants of time.[6]

   Moreover, there is a possibility to understand experimentally whether time Is a fuzzy concept.[7]
   This means this fuzzy time is in harmony with the modern concepts of Physics.
4. Besides all, we have a more easygoing and feasible theory based on this hypothesis (TC^*), as we explain in this article.
5. We have a solution for some paradoxes like "Unexpected hanging paradox".[8]
6. In this paper we will see, even in the case that considering fuzzy time as a physical concept doesn't seem appropriate, this consideration theoretically would be useful. Roughly speaking, by applying this technic we are capable to solve some problems in TC.

   We mention the main theoretical streamline in this article as, considering the contrast betweenTC, TC^* is useful to solve some major questions in the field setting of Complexity theory. However, it is possible that physically TC^* would be a better choice. In general, this provides a novel angel and approach to start studying theory of Computation.

Some part of this theory was introduced in [17], [18], [19].

1. The spaces and model of computation

To give the proof more exactly, first we define $\mathrm{W_m}$. Let .

$$\mathrm{W_m} = \{C_{i,t} \overset{m}{\Rightarrow} C_{j,t} : C_{i,t} \ \& \ C_{j,t} \text{ are configurattions for } M_t\}$$

Now, we define $\mathrm{C} \subsetneq \mathrm{W_m}$ as follows

$\mathrm{C} = \{C_{i,t} \overset{m}{\Rightarrow} C_{j,t} : \mathrm{C_{i,t}} \ \& \ \mathrm{C_{j,t}}$ are configurattions for $\mathrm{M_t}$ and in m steps by transition functtion associatted to $\mathrm{M_t}$, we reach from $\mathrm{C_{i,t}}$ to $\mathrm{C_{j,t}}\}$ (Classical computational world)

We define $\circ$ over $\mathrm{W_m}$

$\circ$: $\mathrm{W_m} \times \mathrm{W_m} \to \mathrm{W_m}$

$(C_{i,t} \overset{m}{\Rightarrow} C_{j,t}) \circ (C_{j,t} \overset{n}{\Rightarrow} C_{k,t}) = (C_{i,t} \overset{m+n}{\Longrightarrow} C_{k,t})$

Furthermore, C induces a directed graph on the space of all configurations, like $\vec{\mathrm{G}}$. We call the underlying graph of this undirected graph, G.

(Remark1. In the case of nondeterministic Turing Machines, we define the concepts in the same way.)

By considering, fuzzy time, we have the possibility of turning back in time. So, any path in the graph G, is a path of possible computation, when in our model we consider instants of time as fuzzy number.

In the definition of path here, the nodes could be repeated but the lengths of paths are finite. $\mathrm{P(G)}$ is the set of all paths of G.

$W_{\mathrm{FUZZY}} = \left\{\mathrm{C_{i,t}} \overset{\mathrm{m}}{\Rightarrow} \mathrm{C_{j,t}} : m \in N, \mathrm{C_{i,t}} \ \& \ \mathrm{C_{j,t}} \text{ belong to a path in } \mathrm{P(G)}\right\}.$

We call $\mathrm{W_{FUZZY}}$ the Fuzzy world. In the fuzzy world, all of these paths are possible.

Here, any instant of time is a fuzzy number, which its support is R, the set of real numbers.

Remark 1. Here, we are able to define the "fuzzy computational model" more exactly, in the case that the area under the instant of time is finite.

Computational-Model= $\{(C_{i,t} \overset{m}{\Rightarrow} C_{j,t}, \eta(C_{i,t} \overset{m}{\Rightarrow} C_{j,t})) : \mathrm{m} \in \mathrm{N}, \mathrm{C}_{\mathrm{i,t}}\ \&\ \mathrm{C}_{\mathrm{j,t}}$ blong to a path in $\mathrm{P(G)}, \eta(\mathrm{C}_{\mathrm{i,t}} \overset{\mathrm{m}}{\Rightarrow} \mathrm{C}_{\mathrm{j,t}})$ is the probability

of reching from $\mathrm{C}_{\mathrm{i,t}}$ to $\mathrm{C}_{\mathrm{j,t}}$ in m steps$\}$

η could be computed by fuzzy function

Corollary 1. If we consider quantum computers instead of non deterministic Turing machines we have the same definitions. It will lead us to a unification of these two theories. In the other words, it is easy to see that, by considering Fuzzy time-particle Interpretation of quantum Mechanics, TC^*, TQC and

TQC^* are the same theories, and the respected Complexity classes are equal sets. In the last chapter, we will show some results by considering this interpretation of quantum mechanics.

2. Reducibility

In this section, firstly, we define a quasi-order relation in TC^* analogous with the m-reducibility in TC .

It should be reminded that a fuzzy time Turing Machine is a Turing Machine which works with fuzzy time, as it is defined in the previous section.

In addition, here, the Turing Machine is considered as a two tuple (M, S). Whereas, M is a Turing machine in the usual sense and S is a polynomial function. Meanwhile, M runs in bounded time by S , equivalently, M(x) in less than S([x]) steps is computed.

First, we remind the Classical definition of m-reducibility:

$\mathrm{Y} >_{\mathrm{m}} \mathrm{X}$, if there is a polynomial time computable function f such that:

$$\mathrm{x} \in \mathrm{X} \leftrightarrow \mathrm{f(x)} \in \mathrm{Y}$$

The parallel definition in TC^* is introduced as following

Definition 1: For $\alpha > \frac{1}{2}$, $\mathrm{Y} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{X}$ if there is a polynomial time computable* function f such that:

1. $\mathrm{x} \in \mathrm{X}\ \&\ \mathrm{f(x)} \downarrow$ in polynomially bounded time respect to $|\mathrm{x}| \leftrightarrow (\mathrm{f(x)} \in \mathrm{Y})$

2. Pr $(\mathrm{f(x)} \downarrow$ in polynomially bounded time respect to $|\mathrm{x}|) > \alpha$

A Computable* function f is a function that is computable by a fuzzy time Turing machine.

Here, by bounded time, we mean that for the function f there exists a Polynomial function h such that $\mathrm{f(x)} \downarrow$ in less than $\mathrm{h}\ (\mathrm{length(x)})$ steps.

$\mathrm{Y} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{X}$ can be represented by a 5-tuple, $(\mathrm{Y}, \mathrm{X}, \mathrm{f}, \mathrm{S_f}, \alpha)$, $\mathrm{S_f(x)}$ is the number of steps that f(x) is computed. The definition is as follows

$$\mathrm{Y} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{X} \leftrightarrow (\mathrm{Y}, \mathrm{X}, \mathrm{f}, \mathrm{S_f}, \alpha) \text{ is an acceptable 5-tuple}$$

One of the major question here is about the independence of the definition from the value of α? $(\alpha > \frac{1}{2})$

In the first step, to answer the above question, we need the following simple lemma.

Lemma 1 Let for $1 > \alpha > \frac{1}{2}$, $(\mathrm{Y}, \mathrm{X}, \mathrm{f}, \mathrm{S_f}, \alpha)$ is an acceptable 5-tuple then for any $1 > \beta > \frac{1}{2}$ there is a computable function g in which $(\mathrm{Y}, \mathrm{X}, \mathrm{g}, \mathrm{S_g}, \beta)$ is an acceptable 5-tuple.

Proof. Actually, there is a natural number k, so that the function g is equivalent to, k times repeating f , till we reach a solution with probability less than β. It is easy to understand that such k exists. □

Lemma 1 indicates for $1 > \alpha > \frac{1}{2}$, the relation $\mathrm{Y} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{X}$ would be independent of α. So, we define $\mathrm{Y} >_{\mathbf{m}}^{*} \mathrm{X}$ as follows

Definition 2. $\mathrm{Y} >_{\mathbf{m}}^{*} \mathrm{X}$ if for some α $(1 > \alpha > \frac{1}{2})$, $\mathrm{Y} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{X}$.

Lemma 2. $>_{\mathrm{m}}^{*}$ is a quasi-order relation.

Proof. $\mathrm{X} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{Y}$ implies $\forall \frac{1}{2} > \varepsilon > 0 \quad \mathrm{X} >_{\mathbf{m}}^{\mathbf{1}-\varepsilon} \mathrm{Y}$ (*)

$\mathrm{Y} >_{\mathbf{m}}^{\boldsymbol{\alpha}} \mathrm{Z}$ implies $\forall \frac{1}{2} > \varepsilon > 0 \quad \mathrm{Y} >_{\mathbf{m}}^{\mathbf{1}-\varepsilon} \mathrm{Z}$ (**)

From (*), (**), we have $\forall \frac{1}{2} > \varepsilon > 0 \quad \mathrm{X} >_{\mathbf{m}}^{(\mathbf{1}-\varepsilon)^2} \mathrm{Y}$ (***). □

Lemma 3. $\mathrm{Y} >_{\mathrm{m}} \mathrm{X}$ implies $\mathrm{Y} >_{\mathbf{m}}^{*} \mathrm{X}$.

Proof. …

Remark 2. Using lemma 3, suppose we have a computation by Turing Machine $(\mathrm{M}, \mathrm{S_f})$ and the input x in classical time and $(\mathrm{M}, \mathrm{S_f})(\mathrm{x}) \downarrow$. If we change the classical time to the symmetric fuzzy time, the probability of reaching to the final state is more than $\frac{1}{2}$. As a conclusion, if we consider the computation $(\mathrm{M}, \mathrm{k}\, \mathrm{S_f})(\mathrm{x}) \downarrow$, the probability of reaching to the final state is more than $1 - \frac{1}{2^k}$.

2.2 $\mathbf{P}^*, \mathbf{NP}^*, \mathbf{NP}^* - \mathbf{Hard}, \mathbf{NP}^* - \mathbf{Complete}$

One of the main questions throughout this article is, how to redefine the most important classes of Complexity Theory in the new theory? As a first attempt, let we try to define P^*as follows:

P^* is the class of all problems that can be determined by a Fuzzy Turing Machine (M, S).

But what exactly do we mean by determined? Since it is possible that we do not reach to the final state, we should consider the possibility associated with $\mathrm{x} \in \mathrm{p}$ for any $\mathrm{p} \in \mathrm{P}^*$ when x belongs to p, and the possibility associated with $\mathrm{x} \notin \mathrm{p}$ when x belongs to p^{c}. Hence, by the above consideration, we are able to modify the definition of P^*, as follows

Definition 3. P^* is a class of problems such that, for any $p\epsilon P^*$ and the probability α, we have a polynomial $Q_{\alpha,p}$ and an associated algorithm $A_{\alpha,p}$ to solve p by probability α such that $Q_{\alpha,p}$ is upper bound of the computation time. Equivalently, for any $p\epsilon P^*$ (p as a language) and probability α we have an associated algorithm $B_{\alpha,p}$ and a polynomial $Q_{\alpha,p}$ as an upper bound of the computation time.
$x\epsilon p$ → By probability α , $B_{\alpha,p} = 1$
$x \notin p$ → By probability α, $B_{\alpha,p} = 0$

This is similar to the definition of the class BPP. Equivalently, by considering time as a Fuzzy concept we have BPP^*.

By the above considerations, it is easy to see:

Theorem 1. $P^* = BPP^*$.

The next natural question in TC^* is the situation of the problem P vs NP, more exactly P^* vs NP^*. Firstly, we are going to prove the following proposition about random generators.

Proposition1. By considering time as a fuzzy concept, random Generators exist.

Proof. …
□

Now, let we consider the following definition of NP problems.

Definition 4 The Complexity class **NP** is the set of decision problems like D such that there is a deterministic polynomial time Turing machine M_D and polynomials p_D, q_D in order that for every input x with length x' ($l(x)=x'$)

1. x belongs to D implies there exists string z with length $q_D(x')$ such that for all string y with length $p_D(x')$, $Pr(M_D(x,y,z) = 1) = 1)$
2. x does not belong to D implies for all string z with length $q_D(x')$ such that for all string y with length $p_D(x')$ $Pr(M_D(x,y,z) = 0) = 1$ (The definition is Quoted in [4])

By considering the above definition and by fuzzifying time we have the definition of NP^*.

We define NP^*--hard, NP^*-Complete likewise in below

Definition 5 X is NP^*-hard if for any $\mathrm{Y} \in \mathrm{NP}^*$, $\mathrm{X} >_{\boldsymbol{m}}^* \mathrm{Y}$.

Definition 6 X is NP^*-Complete if X is NP^*-hard and $\mathrm{X} \in \mathrm{NP}^*$.

Theorem 2 SAT is NP^*-Complete.

Proof. SAT belongs to NP, hence $\mathrm{SAT} \in \mathrm{NP}^*$, by definition. The analogues proof of Cook-Levin's theorem works here. More exactly, by employing the reduction associated with the reduction function f in Cook-Levin theorem with this difference that time is fuzzy, we have the analogous function f^*in the new proof, also here, we consider $>_{\boldsymbol{m}}^*$ instead of m -reducibility. Lemma 3 guarantees the proof of the theorem. □

Theorem3. $\mathrm{P}^* \neq \mathrm{NP}^*$ implies $\mathrm{P} \neq \mathrm{NP}$.

Proof. To prove $\mathrm{P} \neq \mathrm{NP}$, we apply Theorem 2 and lemma 3.

Suppose $\mathrm{P} = \mathrm{NP}$ and we remind that SAT is a NP-Complete problem. Hence, there is an algorithm A which solves SAT in Polynomial time.

Considering Fuzzy time, A also solves SAT in polynomial time, hence SAT belongs to P^*. SAT is NP^*-Complete, so $\mathrm{P}^* = \mathrm{NP}^*$, A contradiction. Consequently, $\mathrm{P} \neq \mathrm{NP}$. □

Lemma 4. $\mathrm{SAT} \notin \mathrm{P}$ implies $\mathrm{SAT} \notin \mathrm{P}^*$, unless $\mathrm{P} = \mathrm{NP}$.

Proof. SAT is NP^*-Complete. Suppose $\mathrm{SAT} \notin \mathrm{P}$. If $\mathrm{SAT} \in \mathrm{P}^*$ then $\mathrm{P}^* = \mathrm{NP}^*$. In brief, $\mathrm{P} \neq \mathrm{NP}$ implies $\mathrm{P}^* = \mathrm{NP}^*$, which contradicts Theorem 4. □

Theorem 4. $\mathrm{P} \neq \mathrm{NP}$ implies $\mathrm{P}^* \neq \mathrm{NP}^*$.

Proof. Suppose $\mathrm{P} \neq \mathrm{NP}$**.** By above lemma, $\mathrm{P} \neq \mathrm{NP}$ implies $\mathrm{SAT} \notin \mathrm{P}^*$. But $\mathrm{SAT} \in \mathrm{NP}^*$, so $\mathrm{P}^* \neq \mathrm{NP}^*$. □

Chapter3. $\mathbf{MA}^*$, $\mathbf{AM}^*$

In the previous chapter, by defining the concepts of P, BPP in the new framework, we define the new classes $\mathrm{P}^*, \mathrm{BPP}^*$. It is shown that the new classes P^*, BPP^* are both equal to each other. In contrast, what is the alternative definition for the NP class in this new framework? To illustrate NP problems in the Theory of Algorithm, it is required to define a new class for it. Possibly MA is the best choice in probabilistic classes [1], [4] (introduced by Laszlo Babai, Shafi Goldwasser, Micheal Sipser).

Indeed, the MA complexity class is known as an alternative for NP problems in probabilistic classes, we also have a theorem states [2], [3]

$$\mathrm{P} = \mathrm{BPP} \rightarrow \mathrm{MA} = \mathrm{NP}$$

The last point, besides $\mathrm{P}^* = \mathrm{BPP}^*$ confirms our choice. So, let we define the concept of NP problems in fuzzy time by applying and similar to the definition of MA. On the other hand in the previous chapter we defined NP^*, as the second way to define an alternative definition for NP in TC^*. It is easy to see, these two ways of defining a parallel concept for NP in TC^*, leads us to the equivalent definitions.

Here, we mention the complexity class Merlin-Arthur MA, in Two-sided version definition[4].

Definition 7. The Complexity class **MA** is a set of decision problems like D such that there are

deterministic polynomial time Turing machine $\mathrm{M_D}$ and polynomials $\mathrm{p_D}$, $\mathrm{q_D}$ in order that for every input x with length x' $(\mathrm{l(x)=x'})$

1. x belongs to D implies there exists string z with length $\mathrm{q_D(x')}$ such that for all string y with length $\mathrm{p_D(x')}$ $\mathrm{Pr}\ (\mathrm{M_D(x,y,z)} = 1) \geq {}^{2}/_{3})$
2. x does not belong to D implies for all string z with length $\mathrm{q_D(x')}$ such that for all string y with length $\mathrm{p_D(x')}$ $\mathrm{Pr}\ (\mathrm{M_D(x,y,z)} = 0) \geq$ ⅔ (The definition is Quoted in [4])

Likewise, we remind the complexity class Arthur-Merlin AM in Two-sided version definition [4].

Definition 8. The Complexity class **AM** is a set of decision problems like D such that there are deterministic polynomial time Turing machine $\mathrm{M_D}$ and polynomials $\mathrm{p_D}$, $\mathrm{q_D}$ in order that for every input x with length $\mathrm{x'}$ $(\mathrm{l(x)=x'})$

1. x belongs to D implies there exists string z with length $\mathrm{q_D(x')}$ such that for all string y with length $\mathrm{p_D(x')}$ $\mathrm{Pr}\ (\mathrm{M_D(x,y,z)} = 1) \geq {}^{2}/_{3})$
2. x dose not belong to D implies for all string z z with length $\mathrm{q_D(x')}$ such that Pr (for all string y with length $\mathrm{p_D(x'), M_D(x,y,z)} = 0) \geq$ ⅔ (The definition is Quoted in [4])

By considering time as a fuzzy concept, we define $\mathrm{MA^*}$. $\mathrm{AM^*}$ is defined similarly, by considering two sided definition of AM in above.

The list of new possible classes which we study here, is
$\mathrm{P^*, NP^*, BPP^*}$,$\mathrm{MA^*}$ $\mathrm{AM^*}$and $\mathrm{AM^*}$.

Instead of $\mathrm{P = NP}$ problem and in parallel to it, we have the following problems

$$\mathrm{BPP^* = MA^*}$$

$$\mathrm{BPP^* = AM^*}$$

$$\mathrm{MA^* = AM^*}$$

Theorems 3&4 shed a light on the above problems.

It is easy to see:

1. $\mathrm{P^* = BPP^*}$ (Theorem 1)
2. $\mathrm{NP^* = MA^*}$ (Considering certificate definition of NP)

It is notable to remind, by proposition 1, we have random generators in the new Theory. So, the pseudo-random generators exist too. In addition, we have $\mathrm{P^* = BPP^*}$ (Theorem 1). In this theory the third major conclusion is about the classes $\mathrm{MA^*, AM^*}$.

Theorem 5. $\mathrm{MA^* = AM^*}$.

Proof. MA is the nondeterministic version of BPP, AM is the probabilistic version of NP.

So, clearly $\mathrm{AM}^* = \mathrm{NP}^*$and MA^* is the nondeterministic version of BPP^*.

By the way, $\mathrm{P}^* = \mathrm{BPP}^*$. Consequently, MA^* is the nondeterministic version of P^*. By definition, $\mathrm{MA}^* = \mathrm{NP}^*$. In sum, $\mathrm{AM}^* = \mathrm{MA}^* = \mathrm{NP}^*$. □

Moreover, by above we have

Theorem 6. The following statements are equivalent

1. $\mathrm{P} \neq \mathrm{NP}$
2. $\mathrm{P}^* \neq \mathrm{NP}^*$
3. $\mathrm{BPP}^* \neq \mathrm{MA}^*(= \mathrm{AM}^*)$

Proof. By Theorems 2, 3, 4, 5. □

Conclusion 1. Throughout this Chapter, it is shown that by considering time as a fuzzy concept, we have random generators. Under this condition, TC^* as a new theory in the field setting of computation is introduced. Hereafter, in the new theory, some problems in parallel to some of the famous problems in Complexity Theory are solved. In brief, $\mathrm{P}^* = \mathrm{BPP}^*$, $\mathrm{MA}^* = \mathrm{AM}^*$.

5. Complexity Classes, $\mathbf{TC} + \mathbf{CON}(\mathbf{TC}^*) \vdash \mathbf{P} \neq \mathbf{NP}, \mathbf{P} = \mathbf{BPP}$

In this section first we present some definitions mostly based on the concepts introduced in the first section, in the second step we define some complexity classes, in the third step we give the proof of the above claims.

Now we define $\mathrm{R}(\mathrm{M}_t)$ as the set of possible computational worlds for M_t.

In section 1, we define $\mathrm{W}_{\mathrm{FUZZY}}$, Computational Model, here we continue to introduce the definitions related to the possible worlds associated to fuzzy time.

Let $\mathrm{R}(\mathrm{M}_t) = \{ \mathrm{W}_{i,t}\}_{i \in \mathrm{I}}$, which the following four conditions hold

1. $\mathrm{W}_{i,t} \subset \mathrm{W}_{\mathrm{FUZZY}}$

2. $\forall m(\mathrm{C}_{l,t} \overset{m}{\Rightarrow} \mathrm{C}_{j,t} \in \mathrm{W}_{i,t})$ implies there is a path between $\mathrm{C}_{i,t}$ &$\mathrm{C}_{j,t}$ in $\mathrm{W}_{\mathrm{FUZZY}}$

3. $\mathrm{C}_{l,t} \overset{m}{\Rightarrow} \mathrm{C}_{j,t} \in \mathrm{W}_{i,t}$ &$\mathrm{C}_{l,t} \overset{m}{\Rightarrow} \mathrm{C}_{k,t} \in \mathrm{W}_{i,t}$ implies $k = j$

4. $\mathrm{W}_{i,t}$ is closed by $\circ$.

Examples:

1. $\{C_{0,t} \overset{m}{\Rightarrow} C_{0,t} : m \in N\} \in R(M_t)$, void world.

2. $C_{l,t} \overset{m}{\Rightarrow} C_{j,t} \in W_{c,t}$ iff $C_{l,t} \overset{m}{\Rightarrow} C_{j,t}$ in classical time in Turing machine M_t. $W_{c,t}$ is the classical world associated to M_t.

2. In the case of Non determinism, we do not consider the third condition in above.

Now, we define $S = \{(w_{i,t})_{t\in N} : W_{i,t} \in R(M_t)\}$, and we recall it the possible worlds of computation. We give here two members of S as examples.

1. Void world of computation. $V \in S$ is Void world of computation by definition if any component of V is a void world.
2. Classical world of computation. $W_{\text{classical}} \in S$ is classical world by definition, if any component of V is a classical world.

Definition 9. The problem X is solved by $W_{k,t}$ in polynomial time means, for some polynomial function p and in less than $p(|a|)$ steps we have either 1 as output if a belongs to X or we have 0 as output if a does not belong to X.

Definition 10. The problem X is solved_1 by $W_{k,t}$ in polynomial time means, for some polynomial function p and in less than $p(|a|)$ steps,

if a belongs to X we have either 1 as output

with a probability greater than or equal $2/3$

or if a does not belong to X

we have 0 as output with probability less than or equal $1/3$.

Definition 11. For $W_k \in S$, $X \in (P, W_k)$ or X is a (P, W_k) problem if X is solved by $W_{k,t}$ in polynomial time, which $W_{k,t}$ is a component of W_k.

Definition 12. For $\mathrm{W_k} \in \mathrm{S}$, $\mathrm{X} \in (\mathrm{BPP}, \mathrm{W_k})$ or X is a $(\mathrm{BPP}, \mathrm{W_k})$ problem if X is solved_1, by $\mathrm{W_{k,t}}$ in polynomial time, which $\mathrm{W_{k,t}}$ is a component of $\mathrm{W_k}$.

Definition 13. $\mathrm{X} \in (\mathrm{NP}, \mathrm{W_k})$, if for some polynomial function Q there is a set $\mathrm{Y} = \{(x, a) : x \in X,\ |a| \text{ is less than } \mathrm{Q}(|x|)\}$, such that $\mathrm{Y} \in (\mathrm{P}, \mathrm{W_k})$.

Remark 3. In the case $\mathrm{W_k} = \mathrm{W_{classical}}$, it is easy to see that, the above definition is equivalent to the following definition

Definition 14. $\mathrm{X} \in (\mathrm{NP}, \mathrm{W_{classical}})$, if X is solvable by non deterministic Turing machine in polynomial time.

Actually, $\mathrm{X} \in (\mathrm{NP}, \mathrm{W_{classical}})$ if and only if $\mathrm{X} \in \mathrm{NP}$ and $\mathrm{X} \in (\mathrm{P}, \mathrm{W_c})$ iff $\mathrm{X} \in \mathrm{P}$.

Poposition 2. $\mathrm{X} \in (\mathrm{P}, \mathrm{W_{classical}})$ iff $\mathrm{X} \in \mathrm{P}$.

Poposition 3. $\mathrm{X} \in (\mathrm{BPP}, \mathrm{W_{classical}})$ iff $\mathrm{X} \in \mathrm{BPP}$.

Poposition 4. $\mathrm{X} \in (\mathrm{NP}, \mathrm{W_{classical}})$ iff $\mathrm{X} \in \mathrm{NP}$.

Poposition 5. $\mathrm{SAT} \in (\mathrm{P}, \mathrm{W_{classical}})$ then $\mathrm{P} = \mathrm{NP}$.

The concepts $\mathrm{m - reucibility}$ and $(\mathrm{NP}, \mathrm{W_{classicl}}) - \mathrm{compelte}$ is defined similar to the classical case.

Poposition 6. $X \in (\mathrm{NP}, \mathrm{W_{classical}}) - \mathrm{complete}$ iff $\mathrm{X} \in \mathrm{NP} - \mathrm{compelete}$.

The concepts like seed and pseudorandom generator are defined analogous with the classical definition.

Corollary 2. If pseudo random generator exists, $(\mathrm{P}, \mathrm{W}_{\text{classical}}) \neq (\mathrm{NP}, \mathrm{W}_{\text{classical}})$, (i.e $\mathrm{P} \neq \mathrm{NP}$).

Proof. If $\mathrm{P} = \mathrm{NP}$ we are able to guess the seeds non deterministically, so pseudo random generator does not exist. □

Corollary 3. If pseudo random generator exists, $(\mathrm{P}, \mathrm{W}_{\text{classical}}) = (\mathrm{BPP}, \mathrm{W}_{\text{classical}})$.

Proof. In above, $\mathrm{X} \in (\mathrm{BPP}, \mathrm{W}_{\text{classical}})$ iff $\mathrm{X} \in \mathrm{BPP}$.

$\mathrm{W}_{\text{classical}}$ is similar to the classical world of computation, nevertheless time is a fuzzy concept. Due to fuzziness of time in this model of computational world, we have random generator, consequently $(\mathrm{P}, \mathrm{W}_{\text{classial}}) \neq (\mathrm{NP}, \mathrm{W}_{\text{classical}})$ **,** $(\mathrm{P}, \mathrm{W}_{\text{classical}}) = (\mathrm{BPP}, \mathrm{W}_{\text{classical}})$ **.** □

By the above proposition we have the following corollary.

Corollary 4. $\mathbf{P} \neq \mathbf{NP}$.

Corollary 5. $\mathbf{P} = \mathbf{BPP}$.

Conclusion 2. In the above proof, our presumption is the existence of a model for TC^*. So we have, $\mathrm{TC} + \mathrm{CON}(\mathrm{TC}^*) \vdash \mathrm{P} \neq \mathrm{NP}, \mathrm{P} = \mathrm{BPP}$.

Therefore, we have $\mathrm{P} \neq \mathrm{NP}$ so we have $\mathrm{P}^* \neq \mathrm{NP}^*$, $\mathrm{P}^* = \mathrm{BPP}^*$ [17], [2] under the same assumption.

Generlization 1. In above, for some specific formulas φ, (here $\mathrm{P} \neq \mathrm{NP}$ and $\mathrm{P} = \mathrm{BPP}$),

$\mathrm{TC} + \text{Existence of random generator} \vdash \varphi$ implies

$\mathrm{TC} + \mathrm{CON}(\mathrm{TC}^*) \vdash \varphi$.

In the first type of generalization, we try to show the type of sentences has the above property.

Firstly, we modify slightly our definitions, we used

$$C_{i,t} \overset{m}{\Rightarrow} C_{j,t}$$

As, possibility of reaching from on configuration to another in m steps. Now we write a similar and more complete form of this definition by

$$C_{i,t}(< x_{i,t} >) \overset{m}{\Rightarrow} C_{j,t}(< y_{j,t} >)$$

$x_{i,t}, y_{j,t}$ are two vectors which shows the data on the strip of Turing Machine, in configurations $C_{i,t}, C_{j,t}$.

Indeed, we define the usual Complexity classes in four groups, as following

1. Computably numerable set, Computable set
2. P,NP,… .
3. Probability Classes.
4. Interactive proofs
   We should enrich the language little by little to capture the concepts in the above four levels. Hence, we create the languages $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4$, associated to these four levels.

1. Σ_1:Computably enumerable set, Computable set

$$x \in L \rightarrow \exists m\, C_{i,t}(< x_{i,t} >) \overset{m}{\Rightarrow} C_{j,t}(< y_{j,t} >) \& (x_{i,t} = x) \& (C_{i,t}(< x_{i,t} >)\ is\ a \text{ start configuration}) \& (C_{j,t}(< y_{j,t} >) \text{ is a final configurattion}) \& (< y_{j,t} >= 1)$$

$$x \notin L \rightarrow \exists m\, C_{i,t}(< x_{i,t} >) \overset{m}{\Rightarrow} C_{j,t}(< y_{j,t} >) \& (x_{i,t} = x) \& (C_{i,t}(< x_{i,t} >) \text{ is } a \text{ start configuration}) \& (C_{j,t}(< y_{j,t} >) \text{ is a final configurattion}) \& (< y_{j,t} >= 0)$$

We need three relations, start configuration, final configuration, $C_{i,t}(< x_{i,t} >) \overset{m}{\Rightarrow} C_{j,t}(< y_{j,t} >)$ in our language.

It shapes first type of languages.

1. Σ_2:P,NP,… .

For P we are able to add a new quantifier bounded one…$\exists m < p(|x|)$, so we need quantifiers like
For NP,… $C_{i,t}(< x_{i,t} >) \overset{m}{\Rightarrow} C_{j,t}(< y_{j,t} >)$, we should add existential quantifiers on configurations. In general, …

2. Σ_3:Probability Classes. For each formula, Ψ we add $\Pr(\Psi) > \alpha$, $\Pr(\Psi) < \alpha$ to the set of formulas.
3. Σ_4:Interactive proofs. Quantifiers on machines and

Quntifiers on V, P and sentences like $V \leftrightarrow P$.

To check the truth of the following

$\text{TC} + \text{Existence of random generator} \vdash \varphi$ implies

$\text{TC} + \text{CON}(\text{TC}^*) \vdash \varphi$.

We should consider the language which φ is defined in it.

…
…

Generlization 2.

Here, we try to answer the following question

Do the above theorems and conclusions remain true if we consider oracle Turing machines?

The major problem in this discussion, is the existence of random generator by considering oracle Turing machines. More exactly,

Definition 15. X-random generator is an oracle Turing machine with oracle X, which feeds by random seeds we have normal distribution as output.

Definition 16. A random generator is independent of oracle X, if it is a X-random generator.

A random generator is independent of oracle, if it is a X-random generator for any X.

(Random generator independent of oracle A).

Theorem 7. If there exists random generator independent of oracle X, $P^X \neq NP^X$.

Proof. Analogues to the above proof (relative version of it). □

Corollary 6. If $X \in \mathrm{Pspace} - \mathrm{Complete}$, we have no X-random generator. (As a theory in TC).

We use this result in the next chapter.

6. Polynomial Hierarchy

In this chapter, we try to generalize the previous results to the classes of polynomial hierarchy. Firstly, PSPACE^* is defined similar to P^*. In the next step, Σ_n^*, Π_n^*

$\Sigma_n^* - \mathrm{Compelete}\ \Pi_n^* - \mathrm{Compelete},$ are defined similar to

$\mathrm{NP}^*, \quad \mathrm{Co} - \mathrm{NP}^*, \quad \mathrm{NP}^* - \mathrm{Compelete}, \quad \mathrm{Co} - \mathrm{NP}^* - \mathrm{Compelete}.$

Definition 17. PSPACE^* is a class of problems such that, for any $p \epsilon \mathrm{PSPACE}^*$ and the probability α, we have a polynomial $Q_{\alpha,p}$ and an associated algorithm $A_{\alpha,p}$ to solve p by probability α such that $Q_{\alpha,p}$ is upper bound of the computation space.
Equivalently, for any $p \epsilon \mathrm{PSPACE}^*$ (p as a language) and probability α we have an associated algorithm $B_{\alpha,p}$ and a polynomial $Q_{\alpha,p}$ as an upper bound of the computation space.
$x \epsilon p$ → By probability α , $B_{\alpha,p} = 1$
$x \notin p$ → By probability α, $B_{\alpha,p} = 0$

Theorem 8. $\mathrm{PSPACE}^* = \mathrm{NSPACE}^*$.
Theorem 9. $\mathrm{PSPACE}^* \subset \mathrm{NSPACE}$.

Proof. Let $\boldsymbol{X} \in \mathrm{PSPACE}^*$, so there is a fuzzy deterministic Turing Machine which solve this problem by using finite memory and decision takes place in bounded polynomial time. Equivalently, there exists a deterministic Turing Machine which its transitive closure solves this problem by using finite memory decision takes place **,** in bounded polynomial time. The transitive closure of this deterministic Turing Machine is a non-deterministic Turing Machine. Hence, $\boldsymbol{X} \in \mathrm{PSPACE}^*$. □
Analogous proof, shows $\mathrm{NSPACE}^* \subset \mathrm{NSPACE}$.

Definition 18. *Transitive closure of a deterministic Turing Machine with transition function δ, is transitive closure of the following relation ≳*

$C_2 \succ C_1$ *if* $\delta(C_1) = C_2$ *among configurations of this Turing Machine.*

Similarly, we have the following definition

Definition 19. *Transitive closure of a deterministic Turing Machine with transition function δ, is transitive closure of the following relation*

$C_2 \gtrsim C_1$ *if* $C_2 \in \delta(C_1)$ *among configurations of this Turing Machine.*

Theorem 10. $\mathrm{NSPACE}^* \subset \mathrm{NSPACE}$.

Proof. Let $\boldsymbol{X} \in \mathrm{PSPACE}^*$, so there is a fuzzy non-deterministic Turing Machine which solve this problem by using finite memory and decides**,** in bounded polynomial time. Equivalently, there exists a non-deterministic Turing Machine which its transitive closure solve this problem by using finite memory decides , in bounded polynomial time. The transitive closure of this non-deterministic Turing Machine is a non-deterministic Turing Machine, again. Hence, $\boldsymbol{X} \in \mathrm{NPSPACE}^*$. □
Corollary 7. $\mathrm{PSPACE} \subset \mathrm{PSPACE}^* \subset \mathrm{NSPACE}^* \subset \mathrm{NSPACE}$, by savitch theorem we have, $\mathrm{PSPACE} = \mathrm{PSPACE}^* = \mathrm{NSPACE}^* = \mathrm{NSPACE}$.

Definition 20 . $\boldsymbol{X} \in \mathrm{PSPACE}^* - \mathrm{Compelete}$, if

1. $\mathrm{X} \in \mathrm{PSPACE}^*$.

2. $\forall Y \in \mathrm{PSPACE}^* \quad X >_{\mathbf{m}}^{*} Y$

Theorem 11 . Every $\mathrm{PSPACE} - \mathrm{Compelete}$ problem is a $\mathrm{PSPACE}^* - \mathrm{Compelete}$ problem. (Page 7)

Proof. Remind that, $Y >_{\mathrm{m}} X$ implies $Y >_{\mathbf{m}}^{*} X$. □

In below, we generalize the concepts and definitions in the previous chapter, first we define PH, PH^*.

Definition 21.

$\Sigma_0^{\mathrm{P}} = \Delta_0^{P} = \Pi_0^{P} = \mathrm{P}$

$\Delta_{i+1}^{P} = \mathrm{P}^{\Sigma_i^{\mathrm{P}}}$

$\Sigma_{i+1}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$,

$\Pi_{i+1}^{P} = \mathrm{Co} - \mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$

$\mathrm{PH} = \bigcup \Delta_{\mathrm{k}}^{\mathrm{p}}$

$\mathrm{P}^{\Sigma_i^{\mathrm{P}}}$, $\mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$, $\mathrm{Co} - \mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$.

Definition 22.

$\Sigma_0^{\mathrm{P},*} = \Delta_0^{P,*} = \Pi_0^{P,*} = \mathrm{P}^*$

$\Delta_{i+1}^{P,*} = \mathrm{P}^{\Sigma_i^{\mathrm{P}},*}$

$\Sigma_{i+1}^{\mathrm{P},*} = \mathrm{NP}^{\Sigma_i^{\mathrm{P}},*}$,

$\Pi_{i+1}^{P,*} = \mathrm{Co} - \mathrm{NP}^{\Sigma_i^{\mathrm{P}},*}$

$\mathrm{PH}^* = \bigcup \Delta_{\mathrm{k}}^{\mathrm{p}}$.

Definiion 23. $\mathrm{sat}_{\mathrm{n}} = \{\boldsymbol{\Phi}: \exists \mathrm{X}_1 \forall \mathrm{X}_2 \exists \mathrm{X}_3 \ldots \mathrm{AX}_{\mathrm{i}} \;\; \boldsymbol{\Phi} = \mathbf{1}\}$, $\mathrm{A} \epsilon \{\exists, \forall\}$ such that $\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3 \ldots, \mathrm{X}_{\mathrm{i}}$ is a partition of variables of $\boldsymbol{\Phi}$.

Theorem 12. The existence of Random generator independent of oracle $\mathrm{sat}_{\mathrm{i+1}}$, implies $\Sigma_{i+1}^{\mathrm{P}} \neq \Delta_{i+1}^{P}$.

Proof . Similar to the case $\mathrm{i} = 0$, when we feed the generator algorithm with oracle $\mathrm{sat}_{\mathrm{i+1}}$ by the random seeds we have stream of random numbers. By choosing Non-deterministically (by using $\mathrm{sat}_{\mathrm{i+1}}$ oracle-non deterministic Turing machine) the seeds, we have all the given strings. Equivalently, the generator is in $\Sigma_{i+1}^{\mathrm{P}}$.

If $\Sigma_{i+1}^{\mathrm{P}} = \Delta_{i+1}^{P}$, the generator algorithm is in Δ_{i+1}^{P}, so the generator is not pseudorandom. □

Definition 24. We call the existence of Random generator independent of oracle **X**, **EX.**

Theorem 13 . $\Sigma_{i+1}^{\mathrm{P}} = \Delta_{i+1}^{P}$ iff $\Sigma_{i+1}^{\mathrm{P},*} = \Delta_{i+1}^{P,*}$.

This is the result of Thorem 16 & 17 in below.

By considering the above definition and by fuzzifying time we defined NP^{*}.

We define $\Sigma_{i}^{\mathrm{P},*}$ --hard, $\Sigma_{i}^{\mathrm{P},*}$ -Complete likewise in below

Definition 25 X is $\Sigma_{i}^{\mathrm{P},*}$ --hard if for any $\mathrm{Y} \in \Sigma_{i}^{\mathrm{P},*}$ --hard, $\mathrm{X} >_{\boldsymbol{m}}^{*} \mathrm{Y}$.

Definition 26 X is $\Sigma_{i}^{\mathrm{P},*}$ -Complete if X is $\Sigma_{i}^{\mathrm{P},*}$ --hard and $\mathrm{X} \in \Sigma_{i}^{\mathrm{P},*}$.

Theorem 14 $\mathrm{SAT_i}$ is Σ_{i}^{P} -Complete.

Theorem 15 $\mathrm{SAT_i}$ is $\Sigma_{i}^{\mathrm{P},*}$ -Complete.

Proof. $\mathrm{SAT_i}$ belongs to Σ_{i}^{P}, hence $\mathrm{SAT_i} \in \Sigma_{i}^{\mathrm{P},*}$, by definition. The analogus proof of Cook-Levin's theorem works here. More exactly, by employing the reduction associated with the above reduction function h in the above theorem, with this difference that time is fuzzy, we have the analogous function h^{*}in the new proof, also here, we consider $>_{\boldsymbol{m}}^{*}$ instead of m -reducibility. Lemma 3 guarantees the proof of the theorem. □

Theorem 16. $\Sigma_{i+1}^{\mathrm{P},*} \neq \Delta_{i+1}^{\mathrm{P},*}$ implies $\Sigma_{i+1}^{\mathrm{P}} \neq \Delta_{i+1}^{P}$.

Proof. To prove $\Sigma_{i+1}^{\mathrm{P}} \neq \Delta_{i+1}^{P}$, we apply Theorem 2 and lemma 3.

Suppose $\Sigma^{\mathrm{P}}_{i+1} = \Delta^{P}_{i+1}$and we remind that SAT_{i+1} is a $\Sigma^{\mathrm{P}}_{i+1}$-Complete problem. Hence, there is oracle Turing Machine $\mathrm{A}^{\mathrm{SAT}_i}$ which solves SAT_{i+1}in Polynomial time.

Considering Fuzzy time, $\mathrm{A}^{\mathrm{SAT}_i}$also solves SAT_{i+1} in polynomial time, hence SAT_{i+1} belongs to $\Delta^{\mathrm{P},*}_{i+1}$. SAT_{i+1} is $\Sigma^{\mathrm{P}}_{i+1}$-Complete, so $\Sigma^{\mathrm{P},*}_{i+1} = \Delta^{\mathrm{P},*}_{i+1}$, a contradiction. Consequently, $\Sigma^{\mathrm{P}}_{i+1} \neq \Delta^{P}_{i+1}$. □

Lemma 5. $\mathrm{SAT}_{i+1} \notin \Delta^{P}_{i+1}$ implies $\mathrm{SAT}_{i+1} \notin \Delta^{P,*}_{i+1}$, unless $\Sigma^{\mathrm{P}}_{i+1} = \Delta^{P}_{i+1}$.

Proof. SAT_{i+1} is $\Sigma^{\mathrm{P},*}_{i+1}$-Complete. Suppose $\mathrm{SAT}_{i+1} \notin \Delta^{P}_{i+1}$. If $\mathrm{SAT}_{i+1} \in \Delta^{P,*}_{i+1}$ then $\Sigma^{\mathrm{P},*}_{i+1} = \Delta^{P,*}_{i+1}$. In brief, $\Sigma^{\mathrm{P}}_{i+1} \neq \Delta^{P}_{i+1}$ implies $\Sigma^{\mathrm{P},*}_{i+1} = \Delta^{P,*}_{i+1}$, which contradicts last theorem. □

Theorem 17. $\Sigma^{\mathrm{P}}_{i+1} \neq \Delta^{P}_{i+1}$ implies $\Sigma^{\mathrm{P},*}_{i+1} \neq \Delta^{\mathrm{P},*}_{i+1}$.

Proof. Suppose $\Sigma^{\mathrm{P}}_{i+1} \neq \Delta^{P}_{i+1}$. By above lemma, $\Sigma^{\mathrm{P}}_{i+1} \neq \Delta^{P}_{i+1}$ implies $\mathrm{SAT}_{i+1} \notin \Delta^{P,*}_{i+1}$. But $\mathrm{SAT}_{i+1} \in \Sigma^{\mathrm{P},*}_{i+1}$, so $\Sigma^{\mathrm{P},*}_{i+1} \neq \Delta^{\mathrm{P},*}_{i+1}$. □

Now, we have our major result in this section

Remark 4. Analogues to the previous chapter we have, $\mathrm{TC{+}CON(TC^*)} + \forall \mathrm{i} \in \mathrm{N}\ \mathrm{ESAT}_i \vdash \Sigma^{\mathrm{P}}_{i+1} \neq \Delta^{P}_{i+1}$. So, under this assumption PH does not collapse.

Theorem 18. $\mathbf{TC} + \mathbf{CON}(\mathbf{TC}^*) + \forall \mathrm{i} \in \mathrm{N}\ \mathbf{E}\mathrm{SAT}_i \vdash \mathrm{PH}^* \subsetneq \mathrm{NSPACE}^*$.

Proof . Easy by above.

Conjecture . For any $\mathrm{i} \in \mathrm{N}$, there exists Random generator independent of oracle SAT_i, $\mathbf{E}\mathrm{SAT}_i$.

Discussion. Intuitively, the following hypothesis seems true,

“The random generator, made by fuzzy time is independent of oracle $\mathrm{X} \in \mathrm{PH}$”.

The author has no proof for that, but this hypothesis and above theorems concludes PH^*& PH do not collapse. We call the above hypothesis ftrg-hypothesis (fuzzy time random generator hypothesis). Under this hypothesis we have

$P \subsetneq NP \subsetneq PH$ and $P \subsetneq NP \subsetneq PSPACE$

$(P^* \subsetneq NP^* \subsetneq PH^*$ and $P^* \subsetneq NP^* \subsetneq PSPACE^*)$.

So, $PH \subsetneq PSPACE$ a parallel proof shows, $PH^* \subsetneq PSPACE^*$.

To do more exactly, we show, there exist $PSPACE^* - \text{Compelete}, {\Sigma_n}^* - \text{Compelete}$, ${\Pi_n}^* - \text{Compelete}$ problems. Actually, it is easy to prove them by theorems in [2]

Poposition 7. $X \in PSPACE - \text{Compelete}$ then $X \in PSPACE^* - \text{Compelete}$.

Poposition 8. $X \in \Sigma_n - \text{Compelete}$ $(\Pi_n - \text{Compelete})$ then $X \in {\Sigma_n}^* - \text{Compelete}$ $({\Pi_n}^* - \text{Compelete})$.

Remark 5., At first glance, the above conclusion seems to be theorems in TC but actually, we need $CON(TC^*)$ and existence of a model for TC^* to prove it. It is noticeable that, our language is not first order. More exactly, we have

3. $TC + CON(TC^*) + \text{ftrg} \vdash P \neq NP, P \subsetneq NP \subsetneq PH \subsetneq PSPACE$

The second type of conclusions, needs TC^* as premises too,

2. $TC + CON(TC^*) + TC^* + \text{ftrg} \vdash P^* \neq NP^*, P^* \subsetneq NP^* \subsetneq PH^* \subsetneq PSPACE^*$

In above, by $CON(T)$ we mean theory T is consistent and has a model.

As a corollary, $TC + CON(TC^*) + TC^* + \text{ftrg}$ deduces graph isomorphism is not a NP-Complete problem.

Quantum Complexity Classes

Here, we consider Fuzzy time particle interpretation of quantum mechanics [6], [7], [8] as a true theory and the basis for theory of Quantum Computation (we call this hypothesis f -hypothesis). Based on this hypothesis and novel interpretation, time is considered as a fuzzy concept. Under the above considerations, we have the following results, immediately by definition

1. $\mathrm{BQP} = \mathrm{BQP}^*, \mathrm{BQP}^* = \mathrm{BPP}^* = \mathrm{P}^*$.
2. $\mathrm{QIP} = \mathrm{QIP}^*$.
3. $\mathrm{QMA}^* = \mathrm{MA}^* = \mathrm{AM}^* = \mathrm{QAM}^*(= \mathrm{QMA} = \mathrm{QAM})$ (By employing theorem ($\mathrm{MA}^* = \mathrm{AM}^*$)).

Results of 2. We have $\mathrm{QIP} = \mathrm{PSPACE}$ [9], [10](The proof should be checked again in the new interpretation, unless f hypothesis considered a true hypothesis). By considering above, we have $\mathrm{PSPACE} = \mathrm{PSPACE}^*$. As a result we have, $\mathrm{IP} = \mathrm{IP}^*$.
Results of 3. Since $\mathrm{MA} \subset \mathrm{QCMA} \subset \mathrm{QMA}$, we have $\mathrm{MA}^* \subset \mathrm{QCMA}^* \subset \mathrm{QMA}^*$. By 3, we have, $\mathrm{MA}^* = \mathrm{QCMA}^* = \mathrm{QMA}^*$.
Results of 1&3. By $\mathrm{TC} + \mathrm{CON}(\ \mathrm{TC}^*)$ we have $\mathrm{P}^* \neq \mathrm{NP}^*(= \mathrm{MA}^* = \mathrm{QCMA}^* = \mathrm{QMA}^*)$. As an important result, $\mathrm{BQP} \neq \mathrm{QMA}(= \mathrm{QAM} = \mathrm{QCMA})$.

The above results are based on a Physical assumption, means different interpretations of quantum Mechanics give the same classes of computational Complexity. It is valuable trying to find some more exact Mathematical proofs.

To complete the above to have a Mathematical proof, we should prove and check the proofs of the below claims in the new interpretation:

$\mathrm{QIP} = \mathrm{PSPACE}$

$\mathrm{MA} \subset \mathrm{QCMA} \subset \mathrm{QMA}$

If the above results are being refuted our Physical assumption would be in danger! It seems unlikely.

Note. Seemingly in the proof of $\mathrm{QIP} = \mathrm{PSPACE}$ [9], [10] we employ superposition of the states in quantum mechanics and if we show how superposition of the states is depicted in this new interpretation, we have a similar proof as it exists in the article.

The explanation would be as following

In any fuzzy instant of time, we have a linear combination of different states. This instant can be depicted as function and the x axis of graph of this function, is abstract time (similar to classical time). In any point of x-axis the system is in one of the states, so the system in this fuzzy instant of time is a combination of states.

We face a new question again: How we find the associated state to a point (abstract time) in x-axis for any fuzzy instant of time?

Although this question is a plausible question for our model, nevertheless it is not a physical question, since in this theory, abstract time is not a Physical concept.

Refrences

1. L.Babai "TRADING Group Theory for Randomness", STOC'85: Proceedings of the seventeenth annual ACM symposium on Theory of Computing, ACM, pp.421-429, 1985

2. O.Goldreich, In a world of P=BPP

3. O.Goldreich, Studies in Complexity and Cryptography: Miscellanea on the interplay

   between Randomness and Computation , Vol 6650 of Lecture Notes in Computer

   Science, Springer 2011, P 43.

4. S.Goldwasser; M.Sipser "Private coins versus public coins in interactive proof systems", STOC'86: Proceedings of the eighteenth annual ACM symposium on Theory of Computing, ACM, PP.59-68, 1986

5. Van Aten M, On Brouwer, Wadsworth Philosopher's Series, 2004
6. F.Didehvar, Computing Fuzzy Time,
7. F.Didehvar, …checkable experimentally …
8. F.Didehvar, ….Fuzzy time a solution for unexpected hanging Paradox, …
9. Rahil Jain, Zhengfeng Ji, Sarvagya Upadyay, John Watrus, QIP=PSPACE, arxiv: 0907.4737, 27 July 2009
10. Rahil Jain, Zhengfeng Ji, Sarvagya Upadyay, John Watrus, QIP=PSPACE, Journal of the ACM, Volume 58, Issue 6, NO:30, pp 1-27

11. L. Blum, F.Cucker, M.Shub, S. Smale, Complexity and Real Computation, Spinger, 1998

12.L. Blum, M.Shub, S. Smale, "On a Theory of Computation and Complexity over the real numbers: NP-Comleness, Recursive Functions and Universal Machines, Bull.Amer.Mth.Soc.(N.S.) 21 (1): 1-46 (July 1989).

13. B.Poizat, Les petit cailloux. Une approach modele-theorique de l'Algorithmie , Nur al-Mantiq wal-Ma'rifah, no.3.Ale'as, Lyon 1995, 2 pp.
Under changing Physical rules and models, Aranson's work

14. S.Aaranson, Guest Column: NP-Complness problem and physical reality ACM Sigact News, 36(1), 30-52, 2005
15. S.Aaranson, Bavarian. M, Gueltrini. G, Compuaabiliy theory of closed time like curves, arXiv preprint arXiv: 1609.05507, 2016
16. S.Aaranson, J.Wattrous, Closed timelike curves make quantum and classical computing equivalent, Proceedings of the Royal Society: Mathematical, Physical and Engineering Sciences, 465, 631-647, 2012.(?)
17. F. Didehvar, Theory of Fuzzy Time Computation(TC*), Philpapers, ssrn2023

18. F. Didehvar, Theory of Fuzzy Time Computation (2) , $(\mathrm{TC} + \mathrm{CON}(\mathrm{TC}^{*}) \vdash \mathrm{P} \neq \mathrm{NP})$
F.Didehvar, Philpapers, ssrn2023

19. F. Didehvar, Theory of Fuzzy Time Computation (3) , $(\mathrm{TC} + \mathrm{CON}(\mathrm{TC}^{*}) \vdash \mathrm{P} \neq \mathrm{NP})$
F.Didehvar, Philpapers, ssrn2023