Designing Quantum Turing Machine is uncomputable

Takayuki Miyadera* and Masanori Ohya
Department of Information Sciences, Tokyo University of Science, Noda City, Chiba 278-8510, Japan

We prove that there is no algorithm to say if an arbitrarily constructed Quantum Turing Machine has same time steps for different branches of computation. Our result suggests that halting scheme of Quantum Turing Machine sholud be analyzed with more attention.

PACS numbers: 03.67.Lx

In [1] Myers pointed out that there may be a problem if on a Quantum Turing Machine (QTM) different branches of quantum computation take different numbers of steps to complete their calculation. That is, in such a case, observation of halting qubit may destroy the computation result since it selects a branch of computation and the quantum interference can not take place after the selection. Subsequently a series of papers [2–4] on the halting process of QTM were published. In [2] Ozawa proposed a possible solution by use of quantum nondemolition measurement scheme. His proposal is restricting QMTs to ones which do not change their halting bit and data slots after the branch drops into the halting state and then the probability to obtain a result by a given time is invariant. In any case, the notion of halting is ambiguous since the halting is probabilistic. A QMT sometimes halts and sometimes does not. Can we say anything valuable with one-time experiment? Bernstein and Vazirani [5] argued that there exists no problem if for each input the different branches of computation always halt with the same time or none of them halt. We here call a QTM with such a condition as properly halting QTM (PHQTM). In the present paper, we discuss on the halting process of QTM from another point of view. A question we want to address here is the following: When we construct a QTM *can we decide whether it is a PHQTM?* We prove that the answer is negative. Our result should suggest that designing PHQTM is difficult in general and thus QTMs which have different computation time for the different branches naturally appears and need to be analyzed with more attention.

A QTM consists of an infinite two-way tape with data slots and other (working) slots, a head and a processor. The total Hilbert space is spanned by a complete orthonormalized set $\{|x\rangle \otimes |\xi\rangle \otimes |q_j\rangle\}$, where x is an infinite sequence of the alphabets $\{B, 0, 1\}$ (B is called as blank) with condition that the number of non-blank cells is finite and $\xi \in \mathbf{Z}$ represents the head position and $q_j \in \{q_0, q_1, \cdots, q_N, q_f\}$ is an internal state. Here q_0 denotes a beginning state and q_f a halting state. A QTM is constructed by assigning complex probability amplitudes (components of a unitary matrix) which satisfy local rule condition. According to [5], the components are assumed to be computable complex number, since otherwise we can not construct the QTM. The halting scheme should be slightly changed from classical Turing machine due to reduction of wave packet. For every step, we observe whether the internal state is q_f or not (i.e., on each step $|q_f\rangle\langle q_f|$ is measured). If the outcome is 1, we measure the data slots in the tape and recognize the computation result. All the known effective computation shemes [6,7] halt with probability one at some time and never halt before then. For an arbitrarily constructed QTM, however, the different branches of computation have different numbers of computation steps in general. In such a case the halting process or the notion of halting itself may have problems. One way to avoid such a difficulty is considering only special type of QTM. Bernstein and Vazirani defined a stationary QTM as a QTM which has same computation steps for different branches of none of the branches halt for each input x. The problem we here address is if we can check an arbitrary QTM is a PHQTM. In other words, we ask if there exists an algorithm [8] to say whether each QTM [9] is a PHQTM or not.

To answer the above question negatively, we assuume the existence of such an algorithm, classical Turing Machine (TM) T_0 and lead contradiction. TM T_0 reads input Q where Q is a QTM and determine whether Q is a PHQTM or not. Let us define a special type of QTM $Q(T_1, T_2)$, where T_1 and T_2 are reversible TMs. The internal state of $Q(T_1, T_2)$ consists of a doubly indexed set $\{(q_*, j), (q_0, j), (q_1, j), \cdots, (q_N, j), (q_f, j), (q_{*f}, j)\}$, where $j = 1, 2$ and N is a sufficiently large number. That is, the Hilbert space of the internal states holds tensor product structure, $\mathbf{C}^{N+4} \otimes \mathbf{C}^2$. The internal state is initialized with $|q_*, 1\rangle$ and a halting state is $|q_{*f}, 1\rangle$. $Q(T_1, T_2)$ with an input x (finite string) behaves as follows:
1) change the internal state from initial state $|q_*, 1\rangle$ to $\frac{3}{5}|q_0, 1\rangle + \frac{4}{5}|q_0, 2\rangle$
2) for the branch with the second qubit of internal state $|1\rangle$, execute the TM T_1 and for the branch of $|2\rangle$, execute T_2.
3) If the internal state is $|q_f, j\rangle$ ($j = 1, 2$), change the internal state plus a fixed tape working cell into $|q_{*f}, 1\rangle \otimes |j\rangle$. (i.e., To satisfy unitarity, an information which branch was lived in is transferred to the tape cell.)
Put a set of all the QTMs of above type as S, i.e.,

$$S := \{Q(T_1, T_2)|\ T_1, T_2 \text{ are reversible TMs}\}.$$

Since S is a subset of whole set of QTMs, TM T_0 could determine whether or not $Q(T_1, T_2)$ is a PHQTM. Then we can determine that for any given reversible TM T_1 and T_2 their computing time for any inputs (including non-halting case) are the same or not. Since we can construct the QTM $Q(T_1, T_2)$ from T_1 and T_2, we obtain a TM T_0' which reads input T_1 and T_2 to compare their computing times, whose output is "Yes" if their computing times are same for each input and otherwise "No".

By use of T_0', we can construct the following TM T_f with its input (T_1, x) where T_1 is a reversible TM and x is its input.

i) Read T_1 and x

ii) Construct the following TM T_2

T_2 with its input y behaves

a) read y

b) if $y \neq x$ execute T_1 with the input y

c) if $y = x$ make a loop and do not halt

Remark the number of steps to complete a) is $c_1 l(y) + c_2$ where c_1 and c_2 are sufficiently large numbers and $l(y)$ is a length of y.

iii) Construct the following TM T_1' with its input z:

A) read z (consume $c_1 l(z) + c_2$ steps exactly)

B) execute T_1 with the input z

iv) input (T_1', T_2) to T_0'

v) Write the output of vi)

We can see that if the outcome is "Yes" TM T_1 with the input x does not halt and if the outcome is "No" TM T_1 with the input x halts. It contradicts the undecidability of halting problem [10] of classical TM. Thus our assertion was proved.

Here we proved that for arbitrarily constructed QTM we can not say if it is a PHQTM. The result would suggest that to consider QTMs with diffierent computation steps for each branches is necessary. The notion of halting in QTM should be discussed with more attention.

T.M. thanks Fumihiko Yamaguchi and Satoshi Iriyama for helpful discussions.

---

* E-mail: Tmdella@aol.com

[1] J. M. Myers, Phys. Rev. Lett. **78** (1997) 1823.

[2] M. Ozawa, Phys. Rev. Lett. **80** (1998) 631. Theoret. Informatics and Appl. **34** (2000) 379.

[3] N. Linden and S. Popescu, quant-ph/9806054

[4] Y. Shi, Phys. Lett. A **293** (2002) 277.

[5] E. Bernstein and U. Vazirani, SIAM Journal on Computing **26**, (1997) 1411

[6] P. W. Shor, SIAM J. Computing, **26** (1997) 1484

[7] L. Grover, Phys. Rev. Lett. **79** (1997) 325

[8] Here the algorithm means classical algorithm.

[9] Of course some special QTMs can be shown to be PHQTMs. What we want to know is the existence of some universal algorithm independent of QTMs.

[10] R. Penrose, *The Emperor's New Mind*, Oxford University Press, (1989)