Math. Struct. in Comp. Science (2006), *vol.* 16, *pp.* 601–620. © 2006 Cambridge University Press
doi:10.1017/S096012950600538X Printed in the United Kingdom

Classically controlled quantum computation

SIMON PERDRIX and PHILIPPE JORRAND

IMAG, Universities of Grenoble, France
Email: {simon.perdrix,philippe.jorrand}@imag.fr

Received 15 October 2005; revised 27 March 2006

It is reasonable to assume that quantum computations take place under the control of the classical world. For modelling this standard situation, we introduce a Classically controlled Quantum Turing Machine (CQTM), which is a Turing machine with a quantum tape for acting on quantum data, and a classical transition function for formalised classical control. In a CQTM, unitary transformations and quantum measurements are allowed. We show that any classical Turing machine can be simulated by a CQTM without loss of efficiency. Furthermore, we show that any k-tape CQTM can be simulated by a 2-tape CQTM with a quadratic loss of efficiency. In order to compare CQTMs with existing models of quantum computation, we prove that any uniform family of quantum circuits (Yao 1993) is efficiently approximated by a CQTM. Moreover, we prove that any semi-uniform family of quantum circuits (Nishimura and Ozawa 2002), and any measurement calculus pattern (Danos *et al.* 2004) are efficiently simulated by a CQTM. Finally, we introduce a Measurement-based Quantum Turing Machine (MQTM), which is a restriction of CQTMs in which only projective measurements are allowed. We prove that any CQTM is efficiently simulated by a MQTM. In order to appreciate the similarity between programming classical Turing machines and programming CQTMs, some examples of CQTMs are given.

1. Introduction

Quantum computations operate in the quantum world. For their results to be useful in any way, by means of measurements for example, they operate under the control of the classical world. Quantum teleportation (Bennett *et al.* 1993) illustrates the importance of classical control: the final correcting Pauli operation is classically controlled by the outcome of a previous measurement. Another example of the importance of classical control is measurement-based quantum computation (Leung 2004; Nielsen 2003; Perdrix 2005; Raussendorf and Briegel 2003; Danos *et al.* 2004), where classical conditional structures are required to control the computation. This classical control may be described as follows: *'if the classical outcome of measurement number i is r_0, then measurement number $i+1$ is on qubit q_a according to observable O_a, otherwise measurement number $i+1$ is on qubit q_b according to observable O_b'*. A particularly elegant formalisation of measurement-based quantum computation is the measurement calculus (Danos *et al.* 2004).

The need to integrate classical control into the description of quantum computations is a now well-understood requirement in the design of high-level languages for quantum programming (Jorrand and Lalire 2004; Selinger 2004). There are also some proposals for lower-level models of quantum computation integrating classical control, such as the

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

quantum random access machines (QRAM) (Knill 1996; Bettelli *et al.* 2003). However, there are no formal and abstract models of quantum computation integrating classical control explicitly. This paper aims to define such an abstract model of classically controlled quantum computation.

One of the main existing abstract models of quantum computation is the Quantum Turing Machine (QTM) introduced by Deutsch (Deutsch 1985), which is an analogue of the classical Turing machine (TM). It has been extensively studied by Bernstein and Vazirani (Bernstein and Vazirani 1997): a quantum Turing machine is an abstract model of quantum computers, which expands the classical model of a Turing machine by allowing a *quantum* transition function. In a QTM, superpositions and interferences of configurations are allowed, but the classical control of computations is not formalised, and inputs and outputs of the machine are still classical. This last point means that the QTM model explores the computational power of quantum mechanics for solving classical problems, but without considering *quantum* problems, that is, quantum input/output.

While models dealing with quantum states like quantum circuits (Kitaev *et al.* 2002; Yao 1993) and QRAM are mainly used for describing specific algorithms, the development of complexity classes, like QMA (Watrous 2000), that deal with quantum states, points out the need for theoretical models of quantum computation acting on quantum data.

The recently introduced Linear Quantum Turing Machine (LQTM) by S. Iriyama, M. Ohya, and I. Volovich (Iriyama *et al.* 2004) is a generalisation of QTMs dealing with mixed states and irreversible transition functions, which allow the representation of quantum measurements without classical outcomes. A consequence of this lack of classical outcome is that classical control is not formalised in LQTMs and, amongst others, schemes like teleportation cannot be expressed. Moreover, as with QTMs, LQTMs deal with classical input/output only.

In this paper, we introduce a Classically controlled Quantum Turing Machine (CQTM), which is a TM with a quantum tape for acting on quantum data, and a classical transition function for formalised classical control. In a CQTM, unitary transformations and quantum measurements are allowed. Theorem 1 shows that any TM can be simulated by a CQTM without loss of efficiency. In Section 5, a CQTM with multiple tapes is introduced. Theorem 2 shows that any k-tape CQTM can be simulated by a 2-tape CQTM with a quadratic loss of efficiency. Moreover, we point out a gap between classical and quantum computations. In Section 6, we compare the CQTM model to two different models of quantum computation, the quantum circuit model (Yao 1993) and the measurement calculus (Danos *et al.* 2004), both of which are efficiently simulated by CQTMs. In Section 8, we present a restriction of CQTMs to measurement-based quantum Turing machine. In an MQTM, only projective measurements are allowed. Theorem 6 shows that any CQTM can be simulated by an MQTM without loss of efficiency. To illustrate the similarity between programming a TM and programming a CQTM, we give some examples of CQTMs for solving problems such as the recognition of quantum palindromes and the insertion of a blank symbol in the input data. Our purpose is to make the CQTM not only a well-defined theoretical model but also a bridge to practical models of quantum computations, such as QRAM, by relying on the fact that natural models of quantum computations are classically controlled.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

2. Quantum computing basics

2.1. *Quantum states*

The basic carrier of information in quantum computing is a 2-level quantum system (*qubit*), or, more generally, a d-level quantum system (*qudit*). The state of a single qudit is a normalised vector of the d-dimensional Hilbert space $\mathbb{C}^d$. An orthonormal basis (o.n.b.) of this Hilbert space is described as $\{|\tau\rangle, \tau \in \Sigma_Q\}$, where Σ_Q is a finite alphabet of symbols such that $|\Sigma_Q| = d$. So the general state $|\phi\rangle \in \mathscr{H}_{\Sigma_Q}$ of a single qudit can be written as

$$\sum_{\tau \in \Sigma_Q} \alpha_\tau |\tau\rangle,$$

with $\sum_{\tau \in \Sigma_Q} |\alpha_\tau|^2 = 1$.

Vectors and inner and outer products are expressed in the notation introduced by Dirac. Vectors are denoted $|\phi\rangle$; the inner product of two vectors $|\phi\rangle$, $|\psi\rangle$ is denoted by $\langle\phi|\psi\rangle$. If $|\phi\rangle = \sum_{\tau \in \Sigma_Q} \alpha_\tau |\tau\rangle$ and $|\psi\rangle = \sum_{\tau \in \Sigma_Q} \beta_\tau |\tau\rangle$, then $\langle\phi|\psi\rangle = \sum_{\tau \in \Sigma_Q} \alpha_\tau^* \beta_\tau$ (where α^* stands for the complex conjugate).

The left hand side $\langle\phi|$ of the inner product is a *bra-vector*, and the right hand side $|\psi\rangle$ is a *ket-vector*. A bra-vector is defined as the adjoint of the corresponding ket-vector: if $|\phi\rangle = \sum_{\tau \in \Sigma_Q} \alpha_\tau |\tau\rangle$, then $\langle\phi| = |\phi\rangle^\dagger = \sum_{\tau \in \Sigma_Q} \alpha_\tau^* \langle\tau|$.

The bra-ket notation can also be used to describe outer products: $|\phi\rangle\langle\psi|$ is a linear operator, $(|\phi\rangle\langle\psi|)\,|\epsilon\rangle = \langle\psi|\epsilon\rangle\,|\phi\rangle$.

The state of a system of n qudits is a normalised vector in $\otimes_{i=1}^n \mathbb{C}^d \cong \mathbb{C}^{d^n}$, where $\otimes$ is the tensor product of vector spaces. $|\tau\gamma\rangle$ denotes $|\tau\rangle \otimes |\gamma\rangle$, such that a basis vector of $\mathbb{C}^{d^n}$ can be denoted $|\omega\rangle$, where $\omega \in \Sigma_Q^n$. As a special case, if $n = 0$, the basis vector in $\mathbb{C}^1$ is denoted $|\rangle$ (Selinger and Valiron 2005). Note that for any $n > 0$, $\mathbb{C}^n \otimes \mathbb{C}^1 \cong \mathbb{C}^n$.

2.2. *Quantum evolutions*

The three basic operations in quantum computing are *unitary transformation*, *initialisation* and *measurement*.

— A unitary operation maps an n-qudit state to an n-qudit state, and is given by a unitary $d^n \times d^n$-matrix U. This unitary operation transforms $|\phi\rangle$ into $U|\phi\rangle$.
— Initialising a qudit according to a special state, say $|\tau_0\rangle$, maps a 0-qudit state to a 1-qudit state, and is given by the matrix $|\tau_0\rangle\langle|$. This initialisation transforms $|\rangle$ into $|\tau_0\rangle\langle|\rangle = |\tau_0\rangle$.
— Two kinds of measurements are considered:
  – A destructive measurement according to an o.n.b. $\{|\tau\rangle, \tau \in \Sigma_Q\}$ maps a 1-qudit state to a 0-qudit state. If the state is $|\phi\rangle$ immediatly before the measurement, the probability that the classical result $\tau \in \Sigma_Q$ occurs is $p(\tau) = |\langle\tau|\phi\rangle|^2$, and the state of the system after the measurement is $|\rangle$.
  – A projective measurement maps an n-qudit state to an n-qudit state, and is given by a collection of $d^n \times d^n$-matrices $\{P_k\}_{k \in I}$, such that $P_k P_l = \delta_{k,l} P_k$ and $\sum_{k \in I} P_k = Id$. Any projective measurement $\{P_k\}_{k \in I}$ can be characterised by an *observable* $O = \sum_{k \in I} \alpha_k P_k$, for some distinct $\alpha_k \in \mathbb{R}$. If the state is $|\phi\rangle$ immediatly

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

before the measurement, the probability that the classical result $k \in I$ occurs is $p(k) = \langle\phi| P_k |\phi\rangle$, and the state of the system after the measurement is

$$\frac{P_k |\phi\rangle}{\sqrt{p(k)}}.$$

Unitary operations can be spatially composed by means of tensor product: if U is an n-qudit unitary operation and V is an m-qudit operation, then $U \otimes V$ is an $n+m$-qudit unitary transformation. Unitary operations can also be sequentially composed by means of matrix products: if U and V are two n-qudit operations, then $V.U$ is an n-qudit unitary transformation consisting of applying U and then V.

The traditional scheme of quantum computation consists of initialising some qudits, then applying unitary operations, and, finally, performing a destructive measurement of each qudit of the system. In this traditional scheme, computations can be described by means of the quantum circuit model (Yao 1993).

Recent alternative models of quantum computation (Danos *et al.* 2004; Perdrix 2005; Raussendorf and Briegel 2003), do not follow this traditional scheme, and allow, for instance, sequential composition of projective measurements. Since projective measurements are not closed under sequential composition, a more general formalism, called *admissible transformations* or *general measurements* is used to describe all the basic quantum operations (unitary operation, initialisation and measurements). Moreover, this formalism is closed under spatial and sequential compositions.

Definition 1 (Admissible transformation). An admissible transformation of an n-qudit state into an m-qudit state is described by a collection $\{M_\tau, \tau \in \Sigma_C\}$ of linear operators mapping $\mathbb{C}^{d^n}$ to $\mathbb{C}^{d^m}$ and satisfying the completness equation

$$\sum_{\tau \in \Sigma_C} M_\tau^\dagger M_\tau = I d_{\mathbb{C}^{d^n}}$$

where Σ_C is a finite set of classical outcomes.

If the state of the quantum system is $|\psi\rangle$ immediately before the transformation, the probability that the classical outcome $\tau \in \Sigma_C$ occurs is given by

$$p(\tau) = \langle\psi| M_\tau^\dagger M_\tau |\psi\rangle,$$

and the state of the system after the transformation is

$$\frac{M_\tau |\psi\rangle}{\sqrt{p(\tau)}}.$$

Property 1 (Sequential composition). Let T be an admissible transformation of an n-qudit state into a m-qudit state, described by $\{M_\tau, \tau \in \Sigma_C\}$, and T' be an admissible transformation transforming an m-qudit state into a k-qudit state, described by $\{N_\gamma, \gamma \in \Sigma'_C\}$. The sequential composition of T and T' is an admissible transformation $\tilde{T}$ transforming an n-qudit state into a k-qudit state, and is described by $\{N_\gamma M_\tau, (\tau, \gamma) \in \Sigma_C \times \Sigma'_C\}$.

Property 2 (Spatial composition). Let T be an admissible transformation of an n-qudit state into an m-qudit state, described by $\{M_\tau, \tau \in \Sigma_C\}$, and T' be an admissible

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

transformation of an n'-qudit state into an m'-qudit state, described by $\{N_\gamma, \gamma \in \Sigma'_C\}$. The spatial composition of T and T' is an admissible transformation $\tilde{T}$ of an $(n+n')$-qudit state into an $(m+m')$-qudit state, described by $\{M_\tau \otimes N_\gamma, (\tau,\gamma) \in \Sigma_C \times \Sigma'_C\}$.

All basic quantum operations can be described by means of admissible transformations:

— A unitary operation U is nothing but an admissible transformation $\{M_\lambda\}$ where $M_\lambda = U$. The completeness equation is satisfied and the classical outcome λ occurs with probability 1, where λ is the void classical outcome.

— A qudit initialisation according to $|\tau_0\rangle$ is described by $\{M_\lambda\}$ where $M_\lambda = |\tau_0\rangle\langle|$. Since $(|\tau_0\rangle\langle|)^\dagger |\tau_0\rangle\langle| = |\rangle\langle\tau_0|\tau_0\rangle\langle| = |\rangle\langle| = Id_{\mathbb{C}^1}$, the completeness equation is satisfied. Moreover, the classical outcome λ occurs with probability 1.

— A destructive measurement according to an o.n.b. $\{|\tau\rangle, \tau \in \Sigma_Q\}$ is described by $\{M_\tau, \tau \in \Sigma_Q\}$ where $M_\tau = |\rangle\langle\tau|$. Since $\{|\tau\rangle, \tau \in \Sigma_Q\}$ is an o.n.b,

$$\sum_{\tau\in\Sigma_Q} |\tau\rangle\langle|\rangle\langle\tau| = \sum_{\tau\in\Sigma_Q} |\tau\rangle\langle\tau| = Id_{\mathbb{C}^{|\Sigma_Q|}},$$

and the completeness equation is satisfied. Moreover, if the state of the qudit is $|\psi\rangle$ just before the measurement, the probability that the classical outcome $\tau \in \Sigma_Q$ occurs is

$$p(\tau) = |\langle\tau|\psi\rangle|^2 = \langle\psi|\tau\rangle\langle\tau|\psi\rangle = \langle\psi||\tau\rangle\langle|\rangle\langle\tau||\psi\rangle = \langle\psi|M_\lambda^\dagger M_\lambda|\psi\rangle.$$

The state of the system after the measurement is

$$\frac{M_\tau|\psi\rangle}{\sqrt{p(\tau)}} = |\rangle.$$

— A projective measurement described by $\{P_k, k \in I\}$ is an admissible transformation described by the same collection of linear operators.

Admissible transformations allow the representation of the basic quantum operations and are closed under sequential and spatial compositions. But one may wonder whether all the admissible transformations have a physical meaning. It turns out that any admissible transformation can be simulated in the traditional scheme of quantum computation consisting of (Nielsen and Chuang 2000):

(i) initialisation,
(ii) sequence of unitary transformations, and
(iii) destructive measurement.

One can imagine a generalised quantum circuit model in which unitary transformations are replaced by admissible transformations. But, unlike unitary transformations, admissible transformations produce a classical result, which allows a classical control consisting, for instance, of conditional compositions and loops. The classically controlled quantum Turing machine is a new model of quantum computation that takes classical control into account.

In the rest of this paper, some basic admissible transformations will be used frequently. For a given Hilbert space $\mathscr{H}_{\Sigma_Q}$, the following are some admissible transformations with classical results belonging to a finite set $\Sigma_C = \Sigma_Q \cup \overline{\Sigma_Q} \cup \{\lambda, \top, \bot\}$, where $\overline{\Sigma_Q} = \{\overline{\tau} : \tau \in \Sigma_Q\}$

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

and $\lambda, \top, \bot \notin \Sigma_Q$:

— $Std = \{M_\tau\}_{\tau \in \Sigma_Q}$ is a projective measurement in the standard basis: $\forall \tau \in \Sigma_Q, M_\tau = |\tau\rangle\langle\tau|$.
— $\mathscr{T}_\tau = \{M_\tau, M_{\overline{\tau}}\}$ is a test for the symbol τ: $M_\tau = |\tau\rangle\langle\tau|$ and $M_{\overline{\tau}} = I - |\tau\rangle\langle\tau|$.
— $\mathscr{P}_{[\tau_a,\tau_b]} = \{M_\lambda\}$ is a unitary transformation with outcome λ, and $M_\lambda = |\tau_a\rangle\langle\tau_b| + |\tau_b\rangle\langle\tau_a| + (\sum_{\tau \in \Sigma_Q - \{\tau_a,\tau_b\}} |\tau\rangle\langle\tau|)$ is a permutation of the symbols τ_a and τ_b.
— $Swap = \{M_\lambda\}$ is a 2-qudit unitary operation with outcome λ, swapping the state of the qudits: $M_\lambda = \sum_{\tau,\gamma \in \Sigma_Q} |\gamma\tau\rangle\langle\tau\gamma|$.
— $\mathscr{U}_V = \{M_\lambda\}$ is the unitary transformation $M_\lambda = V$, with classical outcome λ.
— $\mathscr{O}_O = \{P_k\}_k$, is a projective measurement according to the observable $O = \Sigma_k \alpha_k P_k$.
— $\mathscr{O}_{[\tau_a,\tau_b]} = \{P_\top, P_\bot\} \cup \{P_\tau\}_{\tau \in \Sigma_C - \{\tau_a,\tau_b\}}$ is a projective measurement in a basis *diagonal* according to τ_a, τ_b : $\forall \tau \in \Sigma_C - \{\tau_a, \tau_b\}$, $P_\tau = |\tau\rangle\langle\tau|$, $P_\top = (|\tau_a\rangle + |\tau_b\rangle)(\langle\tau_a| + \langle\tau_b|)/2$, and $P_\bot = (|\tau_a\rangle - |\tau_b\rangle)(\langle\tau_a| - \langle\tau_b|)/2$.

3. Classically controlled quantum Turing machines

For completeness, we give the definition of a deterministic TM (Papadimitriou 1994) in Definition 2. A classically controlled quantum Turing machine (Definition 3) is composed of a quantum tape of quantum cells, a set of classical internal states and a head for applying admissible transformations to cells on the tape. The role of the head is crucial because it implements the interaction across the boundary between the quantum and the classical parts of the machine.

Definition 2. A deterministic (classical) Turing machine is defined by a triplet $M = (K, \Sigma, \delta)$, where K is a finite set of states with an identified initial state s, Σ is a finite alphabet with an identified 'blank' symbol #, and δ is a deterministic transition:

$$\delta : K \times \Sigma \to (K \cup \{\text{'}yes\text{'}, \text{'}no\text{'}, h\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}.$$

We assume that h (the halting state), '*yes*' (the accepting state) and '*no*' (the rejecting state) are not in K.

Definition 3. A classically controlled quantum Turing machine is a quintuple $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$. Here K is a finite set of classical states with an identified initial state s, Σ_Q is a finite alphabet, which denotes basis states of quantum cells, Σ_C is a finite alphabet of classical outcomes, $\mathscr{A}$ is a finite set of one-quantum cell admissible transformations, and δ is a classical transition function:

$$\delta : K \times \Sigma_C \to (K \cup \{\text{'}yes\text{'}, \text{'}no\text{'}, h\}) \times \{\leftarrow, \rightarrow, -\} \times \mathscr{A}.$$

We assume that h (the halting state), '*yes*' (the accepting state) and '*no*' (the rejecting state) are not in K, and that all possible classical outcomes of each transformation of $\mathscr{A}$ are in Σ_C. Moreover, we assume that Σ_Q always contains a 'blank' symbol #, Σ_C always contains a 'blank' symbol # and a 'non-blank' symbol $\overline{\#}$, and $\mathscr{A}$ always contains the admissible 'blank test' transformation $\mathscr{T}_\#$.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

The function δ is a formalisation of the classical control of quantum computations and can also be viewed as the 'program' of the machine. It specifies, for each combination of current classical state $q \in K$ and last obtained classical outcome $\tau \in \Sigma_C$, a triplet $\delta(q,\tau) = (p, D, A)$, where p is the next classical state, $D \in \{\leftarrow, \rightarrow, -\}$ is the direction in which the head will move, and $A \in \mathscr{A}$ is the admissible transformation to be performed next. The *blank test* admissible transformation $\{M_{\#}, M_{\overline{\#}}\}$ establishes a correspondence between the quantum blank symbol (#) and the classical blank (#) and non-blank ($\overline{\#}$) symbols: if the state $|\phi\rangle$ of the measured quantum cell is $|\#\rangle$, the outcome of the measurement is #, but if $|\phi\rangle$ is orthogonal to $|\#\rangle$ ($\langle\phi|\#\rangle = 0$), the outcome is $\overline{\#}$.

How does the program start? The generally unknown quantum input of the computation $|\phi\rangle = \sum_{\tau \in (\Sigma_Q - \{\#\})^n} \alpha_\tau |\tau\rangle$ is placed on n adjacent cells of the tape, while the state of all other quantum cells of the tape is $|\#\rangle$. The head is pointing at the blank cell located immediately to the left of the input. Initially, the classical state of the machine is s and # is considered as the last classical outcome, thus the first transition is always $\delta(s, \#)$.

How does the program halt? The transition function δ is total on $K \times \Sigma_C$ (irrelevant transitions will be omitted from its description). There is only one reason why the machine cannot continue: one of the three halting states h, '*yes*', and '*no*' has been reached. If a machine M halts on input $|\phi_{in}\rangle$, the output $M(|\phi_{in}\rangle)$ of the machine M on $|\phi_{in}\rangle$ is defined. If states '*yes*' or '*no*' are reached, then $M(|\phi_{in}\rangle) =$ '*yes*' or '*no*', respectively. Otherwise, if halting state h is reached, the output is the state $|\phi_{out}\rangle$ of the tape of M at the time of halting. Since the computation has gone on for finitely many steps, only a finite number of cells are not in the state $|\#\rangle$. The output state $|\phi_{out}\rangle$ is the state of the finite register composed of the quantum cells from the leftmost cell in a state that is not $|\#\rangle$ to the rightmost cell in a state that is not $|\#\rangle$. Naturally, it is possible that M never halts on input $|\phi_{in}\rangle$. If this is the case, we write $M(|\phi_{in}\rangle) = \nearrow$.

Since quantum measurement is probabilistic, for a given input state $|\phi_{in}\rangle$, a CQTM does not, in general, always produce the same output, so there exists a probability distribution over possible outputs. Moreover, the halting time of a CQTM M on an input $|\phi_{in}\rangle$ is also a probability distribution. Thus, two special classes of CQTMs can be distinguished: *Monte Carlo* and *Las Vegas*. For a given CQTM M, if for a given input $|\phi_{in}\rangle$ there exists a finite and non-probabilistic bound for the execution time of M, then M is *Monte Carlo*. If the output $M(|\phi_{in}\rangle)$ is not probabilistic, then M is *Las Vegas*. An example of a *Monte Carlo* CQTM is given in Example 1: this CQTM recognises a language composed of 'quantum palindromes', that is, quantum states that are superpositions of palindromes. An example of a *Las Vegas* CQTM, which simulates the application of a given 1-qubit unitary transformation U (H in the example) on a quantum state using projective measurements only, is given in Example 3. In Section 4, we use a CQTM that is both *Las Vegas* and *Monte Carlo* for simulating a classical TM.

A *configuration* of a CQTM M is a complete description of the current state of the computation. Formally, a configuration is a triplet $(q, \tau, |\psi\rangle)$, where $q \in K \cup \{h, \text{'}yes\text{'}, \text{'}no\text{'}\}$ is the internal state of M, $\tau \in \Sigma_C$ is the last obtained outcome, and $|\psi\rangle \in \mathscr{H}_{\Sigma'_Q}$ represents the state of the tape and the position of the head. Here $\Sigma'_Q = \Sigma_Q \cup \underline{\Sigma}_Q$, where $\underline{\Sigma}_Q = \{\underline{\tau} : \tau \in \Sigma_Q\}$ is a set of *pointed* versions of the symbols in Σ_Q: if $|\phi\rangle \in \mathscr{H}_{\Sigma_Q}$ is the state of the tape, and if the head is pointing at cell number k, then $|\psi\rangle \in \mathscr{H}_{\Sigma'_Q}$ is obtained by replacing all

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

$p \in K$,	$\tau \in \Sigma_C$	$\delta(p,\tau)$
s	$\#$	$(q, \rightarrow, Std)$
q	$\#$	$(\text{`}yes\text{'}, -, -)$
q	0	$(q_0, -, \mathscr{P}_{[0,\#]})$
q	1	$(q_1, -, \mathscr{P}_{[1,\#]})$
q_0	λ	$(q_0, \rightarrow, \mathscr{T}_{\#})$
q_0	$\overline{\#}$	$(q_0, \rightarrow, \mathscr{T}_{\#})$
q_0	$\#$	$(q'_0, \leftarrow, Std)$
q_1	λ	$(q_1, \rightarrow, \mathscr{T}_{\#})$
q_1	$\overline{\#}$	$(q_1, \rightarrow, \mathscr{T}_{\#})$
q_1	$\#$	$(q'_1, \leftarrow, Std)$

$p \in K$	$\tau \in \Sigma_C$	$\delta(p,\tau)$
q'_0	$\#$	$(\text{`}yes\text{'}, -, -)$
q'_0	0	$(\tilde{q}, -, \mathscr{P}_{[0,\#]})$
q'_0	1	$(\text{`}no\text{'}, -, -)$
q'_1	$\#$	$(\text{`}yes\text{'}, -, -)$
q'_1	0	$(\text{`}no\text{'}, -, -)$
q'_1	1	$(\tilde{q}, -, \mathscr{P}_{[1,\#]})$
$\tilde{q}$	λ	$(\tilde{q}, \leftarrow, \mathscr{T}_{\#})$
$\tilde{q}$	$\overline{\#}$	$(\tilde{q}, \leftarrow, \mathscr{T}_{\#})$
$\tilde{q}$	$\#$	$(q, \rightarrow, Std)$

Fig. 1. CQTM for quantum palindromes. The symbol '$-$' used as an admissible transformation, means $\mathscr{U}_I$, that is, the identity transformation with λ as classical outcome.

symbols $x \in \Sigma_Q$ at the kth position in $|\phi\rangle$ by the corresponding $\underline{x} \in \Sigma'_Q$. For instance, if $K = \{q_1, q_2\}$, $\Sigma_C = \{\#, \overline{\#}, t, u, v\}$ and $\Sigma_Q = \{\#, a, b\}$, the configuration

$$(q_1, u, \frac{1}{\sqrt{2}}(|a\#\underline{b}b\rangle + |b\#\underline{a}b\rangle))$$

means that the internal state of the machine is q_1, the last outcome is u, the state of the tape is $\frac{1}{\sqrt{2}}(|a\#bb\rangle + |b\#ab\rangle)$, and the head is pointing at the third cell from the left.

Example 1 (Quantum palindromes). Consider the CQTM $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$, with $K = \{s, q, q_0, q_1, q'_0, q'_1, \tilde{q}\}$, $\Sigma_C = \{\#, \overline{\#}, 0, 1, \lambda\}$, $\Sigma_Q = \{\#, 0, 1\}$ and $\mathscr{A} = \{\mathscr{T}_{\#}, Std, \mathscr{P}_{[0,\#]}, \mathscr{P}_{[1,\#]}\}$ (these admissible transformations are defined in Section 2), and δ as described in Figure 1.

The purpose of this machine is to tell whether its input is a quantum palindrome, that is, a state that is a superposition of basis states such that each basis state in the superposition is a palindrome. For instance, the states $|00\rangle$, $\frac{1}{\sqrt{2}}(|010\rangle + i|111\rangle)$, $\frac{1}{\sqrt{2}}(|00000\rangle + |11111\rangle)$ are quantum palindromes. The machine works as follows: the first cell of the input is measured in the standard basis and replaced with $|\#\rangle$, the result is memorised by means of the internal states q_0 and q_1, then M moves right, up to the end of the input. The last cell is then measured in the standard basis: if the outcome agrees with the one remembered, it is replaced with $|\#\rangle$. M then moves back left to the beginning of the remaining input and the process is repeated. The transition function is described in Figure 1. For instance, if the internal state is q_0 and the last obtained classical outcome is $\#$, then the internal state becomes q'_0, the head moves to the left and then the pointed at cell is measured in the standard basis.

This machine is a Monte Carlo CQTM operating in time $O(n^2)$, where n is the size of the input. Considering the language $L \subset \mathscr{H}_{\Sigma_Q}$ composed of quantum palindrome states, if $|\phi_{in}\rangle \in L$, then the probability that M accepts $|\phi_{in}\rangle$ is $Pr[M(|\phi_{in}\rangle) = \text{`}yes\text{'}] = 1$: if the input is a quantum palindrome, then, in any case, the machine recognises $|\phi_{in}\rangle$, but M may accept states that are not palindromes with high probability, for instance, $\forall \epsilon > 0, Pr[M(\sqrt{1-\epsilon}\,|00\rangle + \sqrt{\epsilon}\,|10\rangle) = \text{`}yes\text{'}] = 1 - \epsilon$.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

4. CQTM and TM

The following theorem shows that any TM is simulated by a CQTM without loss of efficiency.

Theorem 1. Given any TM M_C operating in time $f(n)$, where n is the input size, there exists a CQTM M_Q operating in time $O(f(n))$ and such that for any input x, $M_C(x) = M_Q(|x\rangle)$[†].

Proof. For a given TM $M_C = (K, \Sigma, \delta_C)$, we describe a CQTM M_Q that simulates M_C. One way to do this is to simulate the classical tape of M_C using only basis states of the quantum tape of M_Q.

Formally, we consider the CQTM $M_Q = (K \cup K_\Sigma \cup \{s'\}, \Sigma \cup \{\overline{\#}, \lambda\}, \Sigma, \mathscr{A}, \delta_Q)$. Here $K_\Sigma = \{q_\tau : q \in K, \tau \in \Sigma\}$ and $\mathscr{A} = \{Std\} \cup \{\mathscr{P}_{[\tau_1,\tau_2]}\}_{\tau_1,\tau_2 \in \Sigma}$. The initial state of M_Q is s' and its first transition is $\delta_Q(s', \#) = (s, -, Std)$, where s is the initial state of M_C. For any $(q, \tau) \in K \times \Sigma$, the transition $\delta_C(q, \tau) = (q', \tau', D)$ is decomposed into two transitions: $\delta_Q(q, \tau) = (q_\tau, -, \mathscr{P}_{[\tau,\tau']})$ and $\delta_Q(q_\tau, \lambda) = (q', D, Std)$.

Since each transition of M_C is simulated with probability 1 by two transitions of M_Q, if M_C operates in time $f(n)$, M_Q operates in time $2f(n)$, where n is the size of the input. □

Any TM is simulated by a CQTM without loss of efficiency. However, as will be shown in Lemma 1, a CQTM with one tape cannot simulate some other models of quantum computation, like quantum circuits, because only one-cell admissible transformations are allowed. In order to allow transformations on more than one cell, we introduce multi-tape CQTMs. With k heads, k-cell admissible transformations can be performed.

5. CQTM with multiple tapes

We show that any k-tape CQTM is simulated by a 2-tape CQTM with an inconsequential loss of efficiency. Moreover, by showing that 1- and 2-tape CQTM are not equivalent, we point out a *gap* between classical and quantum computations.

Definition 4. A k-tape classically controlled quantum Turing machine where $k > 0$, is a quintuple $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$, where K is a finite set of classical states with an identified initial state s and Σ_Q is a finite alphabet that denotes basis states of each quantum cell, $\mathscr{A}$ is a finite set of k-cell admissible transformations, Σ_C is a finite alphabet of classical outcomes of k-cell admissible transformations and δ is a classical transition function

$$\delta : K \times \Sigma_C \to (K \cup \{\text{`}yes\text{'}, \text{`}no\text{'}, h\}) \times (\{\leftarrow, \rightarrow, -\})^k \times \mathscr{A}.$$

We assume that all possible classical outcomes of each measurement of $\mathscr{A}$ are in Σ_C and that $\mathscr{A}$ always contains the k admissible 'blank test' transformations, one for each tape of the machine.

[†] If the halting state h is reached, $M_Q(|x\rangle)$ denotes the final state of the tape. So, if h is reached, $M_C(x) = M_Q(|x\rangle)$ has to be replaced by $M_Q(|x\rangle) = |M_C(x)\rangle$.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

$p \in K,$	$\tau \in \Sigma_C$	$\delta(p,\tau)$
s	$\#$	$(q_0, (\leftarrow, -), Swap)$
q_0	λ	$(q_1, (\rightarrow, -), Swap)$
q_1	λ	$(h, (\rightarrow, -), -)$

Fig. 2. 2-tape CQTM for inserting a blank symbol.

Intuitively, $\delta(q,\tau) = (q', (D_1, \ldots D_k), A)$ means that if M is in state q and the last classical outcome is τ, then the next state will be q', the k heads of the machine will move according to $D_1, \ldots, D_k$, and the next k-quantum cell admissible transformation will be A. This admissible transformation will be performed on the k quantum cells pointed at by the heads of the machine after they have moved. A k-cell admissible transformation A can be defined directly, for instance by use of a k-cell unitary transformation V ($A = \mathscr{U}_V$). A can also be defined as a composition of two admissible transformations A_1 and A_2, respectively, on j and l cells such that $j + l = k$, then $A = [A_1, A_2]$ means that the first j heads apply A_1 and, simultaneously, the last l heads apply A_2. The classical outcome is the concatenation of the outcomes of A_1 and A_2, where λ is the unit element of the concatenation (that is, $\tau.\lambda = \tau$).

A k-tape CQTM starts with an input state $|\phi\rangle$ on a specified tape T_1, with all cells of other tapes in state $|\#\rangle$, and if the halting state h is reached, the machine halts and the output is the state of the specified tape T_1.

Example 2 (Inserting a blank symbol). Consider the problem of inserting a blank symbol between the first and second cells of a quantum state $|\psi_{in}\rangle$ that resides on one of the tapes. For instance, $|abba\rangle$ is transformed into $|a\#bba\rangle$, and $\frac{1}{\sqrt{2}}(|aa\rangle + |bb\rangle)$ into $\frac{1}{\sqrt{2}}(|a\#a\rangle + |b\#b\rangle)$. Consider the 2-tape CQTM $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$, with $K = \{s, q_0, q_1\}$, $\Sigma_C = \{\#, \overline{\#}, \lambda\}$ and $\mathscr{A} = \{\mathscr{T}_{\#}, Swap\}$. δ is described in Figure 2.

The input state is on the first tape. Let a be a name for the first cell on the left of the input. In order to insert a blank symbol in the second position of the input state, the state of a is swapped with a cell of the second tape. Then the state of this cell on the second tape is swapped with the state of the cell immediatly located to the left of a.

Theorem 2. Given any k-tape CQTM M operating in time $f(n)$, where n is the input size, there exists a 2-tape CQTM M' operating in time $O(f(n)^2)$ and such that for any input $|\psi\rangle$, $M(|\psi\rangle) = M'(|\psi\rangle)$.

Proof. Suppose that $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$ has k tapes, we describe $M' = (K', \Sigma'_C, \Sigma'_Q, \mathscr{A}', \delta')$ having only two tapes. M' must 'simulate' the k tapes of M. One way to do this is to maintain on one tape T_1 of M' the *concatenation* of the contents of the tapes of M. The position of each head must also be remembered.

To accomplish this, $\Sigma'_Q = \Sigma_Q \cup \underline{\Sigma}_Q \cup \{\triangleright, \triangleleft\}$, where $\underline{\Sigma}_Q = \{\underline{\tau} : \tau \in \Sigma_Q\}$ is a set of pointed versions of the symbols in Σ_Q, and $\triangleright$ ($\triangleleft$) signals the left-hand (right-hand) end of each

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

simulated tape. Intuitively, at each step of the computation, if $|\phi_j\rangle$ is the state of each tape j of M, the state of the tape T_1 of M' is $|\triangleright\triangleright\rangle\,|\phi_1\rangle\,|\triangleleft\triangleright\rangle\,|\phi_2\rangle\,|\triangleleft\triangleright\rangle\ldots|\triangleleft\triangleright\rangle\,|\phi_k\rangle\,|\triangleleft\triangleleft\rangle$. In order to remember the positions of the k heads, a unitary transformation is applied to the cells of M' corresponding to the cells of M pointed at by the heads of M. This unitary transformation replaces the symbols of Σ_Q by their corresponding versions in $\underline{\Sigma}_Q$

Since each k-cell admissible transformation from $\mathscr{A}$ can be decomposed into $l_{\mathscr{A}}$ 2-cell admissible transformations (see Muthukrishnan and Stroud (2000)), $\mathscr{A}'$, which is composed of 1- and 2-cell admissible transformations, is defined such that any transformation from $\mathscr{A}$ can be simulated with a finite number, $l_{\mathscr{A}}$, of transformations of $\mathscr{A}'$.

For the simulation to begin, M' inserts a $\triangleright\triangleright$ to the left and $\triangleleft(\triangleright\triangleleft)^k\triangleleft$ to the right of the input, since the input of M is located on its first tape. To simulate a transition $\delta(q,\tau)=(q',D,A)$ of M, the pointed at cells change first according to D. Note that if a head meets the symbol $\triangleright$, then a blank symbol is inserted to the right of this cell (see Example 2) to simulate the infinity of the tapes, and, similarly, for the symbol $\triangleleft$. A is simulated via a sequence of 2-cell transformations. Since 2-cell transformations can only be performed on cells located on different tapes, the state of one of the two cells is transferred (by means of *Swap*, see Example 2) from tape T_1 to tape T_2. Then the 2-cell transformation is performed, and the state located on T_2 is transferred back to T_1, and so on. In order to reconstruct the classical outcome of the simulated transformation A, M' must go through new internal states that keep track of the classical outcomes of the different 1- and 2-cell transformations.

The simulation proceeds until M halts. How long does the computation from an input $|\phi\rangle$ of size n take? Since M halts in time $f(n)$, no more than $k.f(n)$ cells of M are non-blank cells. Thus the total length of the non-blank cells of M' is $k.(f(n)+2)+3$ (to account for the $\triangleleft,\triangleright$ and the cell of T_2 used for the application of 2-quantum cell transformations). Simulating a move of the heads takes at most two traversals of the non-blank cells of T_1. Each simulation of an admissible transformation of $\mathscr{A}$ requires a constant number $l_{\mathscr{A}}$ of transformations of $\mathscr{A}'$ ($l_{\mathscr{A}}$ is independent of the input size), moreover, the simulation of each transformation in $\mathscr{A}'$ requires two traversals. As a consequence, the simulation of each transition of M requires $O(f(n))$ transitions of M', thus the total execution time of M' is $O(f(n)^2)$. □

The following lemma shows that some 2-tape CQTMs cannot be simulated by 1-tape CQTMs.

Lemma 1. There exists a 2-tape CQTM M such that no 1-tape CQTM simulates M.

Proof. Let $M=(\{s\},\{\lambda,\#,\overline{\#}\},\{\#,0\},\{\mathscr{U}_V\},\delta)$ be a 2-tape CQTM where $V=\frac{1}{2\sqrt{2}}\left((|\#\#\rangle+|00\rangle)\langle\#\#|+(|\#\#\rangle-|00\rangle)\langle\#0|+(|\#0\rangle+|0\#\rangle)\langle 0\#|+(|\#0\rangle-|0\#\rangle)\langle 00|\right)$, and $\delta(s,\#)=(h,-,\mathscr{U}_V)$. If the input is $|0\rangle$, then, when the machine halts, the state of the cells pointed at by the heads is entangled: $\frac{1}{\sqrt{2}}(|\#0\rangle+|0\#\rangle)$. Thus, there is no 1-tape CQTM that simulates M, since entanglement cannot be created by means of one-cell admissible transformations. □

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

6. CQTM can simulate other quantum computational models

6.1. *Unitary-based quantum computation: quantum circuits*

Theorem 2 is strong evidence of the power and stability of CQTMs: adding a bounded number of tapes to a 2-tape CQTM does not increase its computational capabilities, and impacts on its efficiency only polynomially. This stability makes 2-tape CQTMs a good candidate for quantum universality, that is, the ability to simulate any quantum computation. This ability is proved with the following theorem by the simulation of any semi-uniform family of quantum circuits (Nishimura and Ozawa 2002). In this section, some basic notions and properties of quantum circuits are given, refer to Yao (1993) and Kitaev *et al.* (2002) for the fundamentals of quantum circuits.

In the quantum circuit model, the carrier of information is restricted to qubits. Basis states are denoted $|0\rangle$ and $|1\rangle$.

Definition 5 (Quantum circuits). Let $\mathscr{G}$ be a fixed set of unitary operators (also called unitary gates.) An n-qubit quantum circuit C_n based on $\mathscr{G}$ is a sequence $(G_1, R_1), \cdots, (G_s, R_s)$, where $G_i \in \mathscr{G}$, and R_i is an ordered set of qubits, and s is the size of C_n for $\mathscr{G}$.

A uniform family of quantum circuits is a set of circuits $F = \{C_n\}$ such that a classical Turing machine M_F can produce a description of C_n on input n in time $poly(s(n))$, where $s(n)$ is the size of C_n. A semi-uniform family of quantum circuits $\{C_n\}$ is a uniform family defined on a finite set $\mathscr{G}$ of operators.

Theorem 3. For any semi-uniform family of quantum circuits $F = \{C_n\}$ of size s, there exists a 2-tape CQTM M operating in time $poly(n, s(n))$ such that for any n-qubit state $|\psi\rangle$, $C_n|\psi\rangle = M(|\psi\rangle)$.

Proof. Applying $F = \{C_n\}$ to an n-qubit state $|\psi\rangle$ consists of applying the quantum circuit C_n with $|\psi\rangle$ as input. Let $\mathscr{G}$ be a basis of F, that is, the set of all the unitary gates used in F. Since $\mathscr{G}$ is finite, $\mathscr{G}$ has a finite arity w, that is, for all $G \in \mathscr{G}$, the arity of G is less than w, where the arity of a gate is the number of qubits on which it operates.

The description of C_n produced by M_F is of the form $(G_1, R_1) \cdots (G_{s(n)}, R_{s(n)})$, meaning that $G_i \in \mathscr{G}$ is applied to the set $\{R_i^{(j)}\}_j$ of qubits in R_i, then G_{i+1} is applied, and so on.

Let M be a $(w+1)$-tape CQTM. The admissible transformations of M include the unitary transformation $\mathscr{U}_G$, for all $G \in \mathscr{G}$.

A general description of the evolution of M is as folows:

— The size n of the input $|\psi\rangle$ located on tape 1 is computed, using the blank test $\mathscr{T}_{\#}$ admissible transformation, and stored on tape T_{w+1}. This initial stage can be realised within a linear number of steps in n.
— M_F is simulated (see Theorem 1) and produces a classical description of C_n on tape T_{w+1}. The complexity of this stage is $p(s(n))$, for some polynomial p.
— For each (G_i, R_i), if $R_i = \{R_i^{(j)}\}_j$, for each j, the qubit $R_i^{(j)}$ of tape 1 is transferred to tape j, then $\mathscr{U}_{G_i}$ is applied, and the qubits of R_i are transferred back to tape 1. This stage can be realised within $O(ns(n))$ steps.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

One can show that the resulting state on tape 1 is the state produced by $C_n|\psi\rangle$. Moreover, this simulation can be done within $ns(n) + p(s(n))$ steps.

Finally, the $(w+1)$-tape CQTM M is simulated by a 2-tape CQTM M': the first two stages of M, consisting of computing the size of the input and simulating the classical Turing machine M_F can be simulated without slowdown on M' since only two tapes are required. On the other hand, the third stage of M is simulated with a quadratic slowdown (see Theorem 2).

Thus the 2-tape CQTM M' operates in time $O(n^2 s(n)^2 + p(s(n)))$. □

Corollary 1. For any *poly-size* semi-uniform family of quantum circuits $F = \{C_n\}$ there exists a *poly-time* 2-tape CQTM M such that for any n and any input $|\psi\rangle$ on n qubits, $C_n|\psi\rangle = M(|\psi\rangle)$.

Semi-uniform families of quantum circuits can be simulated in polynomial time by means of a 2-tape CQTM. Unlike semi-uniform families, uniform families of quantum circuits have countable but not necessarily finite basis of gates. Since any CQTM is based on a finite set of admissible transformations, we conjecture that some uniform families of quantum circuits cannot be simulated by means of a CQTM. However, any uniform families of quantum circuits can be approximated.

Theorem 4. For any $\epsilon > 0$, and for any uniform family of quantum circuits $F = \{C_n\}$ of size s, there exists a 2-tape CQTM M operating in time $poly(n, s(n), 1/\epsilon, 2^w)$ (where w is the arity of the gates of C_n) and such that for any n and any input $|\psi\rangle$ on n qubits,

$$||C_n|\psi\rangle - M(|\psi\rangle)|| < \epsilon,$$

where $||.||$ is the Euclidian norm.

Proof. The proof consists of combining Theorem 3 with the approximation of any unitary transformation using a finite set of unitary transformations.

There exists a finite set of unitary transformations $\mathscr{U}$ on at most 2-qubits such that the approximation within ϵ of an operation on w qubits takes $poly(1/\epsilon, 2^w)$ gates from $\mathscr{U}$ (Aharonov 1998). Moreover, there exists an algorithm operating in time $poly(1/\epsilon, 2^w)$ that constructs a description of the circuit realising the approximation of U (Bernstein and Vazirani 1997).

For a given $F = \{C_n\}$ and $\epsilon > 0$, let M be a 2-tape CQTM. A general description of the evolution of M is as follows:

— The size n of the input $|\psi\rangle$ located on tape 1 is computed, using the blank test $\mathscr{T}_{\#}$ admissible transformation, and stored on tape 2. This initial stage can be realised within a linear number of steps in n.
— M_F is simulated (see Theorem 1) and produces a classical description of C_n on tape 2. This stage is realised in time $poly(s(n))$.
— In the description of C_n, each gate is replaced by an approximation within $\epsilon/s(n)$ in time L, where L is a polynomial in $s(n)/\epsilon$ and 2^w. This stage is realised in time $Ls(n)$ and the descrition of the circuit is now composed of $poly(s(n), 1/\epsilon, 2^w)$ gates.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

— For each (G_i, R_i), if $R_i = \{p_i^{(j)}\}_j$, for each j, the qubit $R_i^{(j)}$ of tape 1 is transferred to tape j, then $\mathscr{U}_{G_i}$ is applied, and the qubits of R_i are transferred back to tape 1.

Thus, M operates in time $poly(n, s(n), 1/\epsilon, 2^w)$. □

Corollary 2. For any $\epsilon > 0$, and for any *poly-size* uniform family of quantum circuits $F = \{C_n\}$ there exists a 2-tape CQTM M operating in time $poly(n, 1/\epsilon, 2^w)$ (where w is the arity of the gates of C_n) and such that for any n and any input $|\psi\rangle$ on n qubits, $||C_n|\psi\rangle - M(|\psi\rangle)|| < \epsilon$.

6.2. *Measurement-based quantum computation: measurement calculus*

Alternative models of quantum computation, such as the measurement calculus (Danos *et al.* 2004), can also be simulated with 2-tape CQTMs in polynomial time.

In the measurement calculus, computations are described by means of *patterns*. A pattern $\mathscr{P}$ consists of three finite sets of qubits V, I, O, such that $I \subset V$ and $O \subset V$, and a finite sequence of commands $A_s \cdots A_1$ applying to qubits in V. Qubits in $V \setminus I$ are initialised in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, then each command is successively applied. Each command is:

— a two-qubit unitary transformation called controlled-Z, or
— a one-qubit measurement, or
— a corrective Pauli operator, which is applied or not according to the previous classical outcomes of measurements.

When all commands in the pattern have been applied, the output state is the state of the qubits in O.

Lemma 2. For any measurement-calculus pattern $\mathscr{P}$ with commands $A_s \cdots A_1$, acting on n qubits, there exists a 2-tape CQTM $M_{\mathscr{P}}$ acting in time s^2, such that for any $|\psi\rangle$ on n qubits, $\mathscr{P}(|\psi\rangle) = M_{\mathscr{P}}(|\psi\rangle)$.

Proof. A general description of the evolution of the 2-tape CQTM $M_{\mathscr{P}}$ simulating $\mathscr{P}$ is as follows:

— Qubits corresponding to qubits in $V \setminus I$ are initialised in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ within a linear number of steps in s.
— Since the sequence of commands is finite, classical internal states of $M_{\mathscr{P}}$ are used to describe the sequence of commands. For each command, the head moves first to the addressed qubit on the first tape, in at most s steps, then the command is applied. In the case of controlled-Z, this operation is simulated by transferring one of the two qubits to the second tape, applying controlled-Z, and transferring the qubit back to the first tape. In this case the number of steps is still linear in s. If the command is a one-qubit measurement, the measurement is performed and the classical outcome is stored in the internal state (a bounded memory is enough to store all the classical outcomes since the pattern is finite and each measurement is applied once.) If the command is a corrective Pauli operator, this operator is applied or not, depending on the internal states used to store the previous classical outcomes.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

— Finally, the output qubits corresponding to the set O are reorganised on the first tape in a number of steps quadratic in s.

One can show that the resulting state on tape 1 is $\mathscr{P}(|\psi\rangle)$. This simulation can be done within s^2 steps. □

In Lemma 2, unlike Theorem 3, where a semi-uniform family of circuits are simulated, only one pattern, with a fixed input size, is simulated. One can extend the model of the measurement calculus by introducing notions of uniform family of patterns and semi-uniform family of patterns, in a similar way to the quantum circuit model case. Then, similar results to those obtained with quantum circuits can be proved.

Since a 2-tape CQTM naturally simulates unitary-based as well as measurement-based models of quantum computation, the classically controlled quantum Turing machine is a unifying model of quantum computation including most of the ingredients that can be used to realise quantum computations: admissible transformations (unitary transformations or projective measurements are just special instances of admissible transformations); and classical control as in the measurement-calculus, which is a model for one-way quantum computation (Raussendorf and Briegel 2003).

7. 1-tape CQTM vs 2-tape CQTM

To sum up, two tapes are enough for quantum computation (theorem 3): one tape is enough for classical computation (Theorem 1), but not for quantum computation (Lemma 1). Thus a *gap* between classical and quantum computations appears. Note that this result does not contradict the equivalence, in terms of decidability, between classical and quantum computations: the gap appears if and only if quantum data are considered.

One may wonder why 1-tape CQTMs are not quantum universal even though Briegel and Raussendorf have proved, with their One-way quantum computer, that one-qubit measurements are universal (Raussendorf and Briegel 2003). The proof by Briegel and Raussendorf is given with the strong assumption that there exists a grid of auxiliary qubits that have been initially prepared, by some unspecified external device, in a globally entangled state (the *cluster state*), while the *creation* of entanglement is a crucial point in the proof of Lemma 1. Moreover, another strong assumption of one-way quantum computation is that the input state $|\psi\rangle$ has to be classically known (that is, a mathematical description of $|\psi\rangle$ is needed), while the manipulation of unknown states (that is, the manipulation of qubits in an unknown state) is usual in quantum computation (for example, in teleportation (Bennett *et al.* 1993).) Since none of these assumptions are verified by the 1-tape CQTM, the previous results do not contradict the results of Briegel and Raussendorf.

8. Measurement-based quantum Turing machine

In the CQTM model the set of admissible transformations $\mathscr{A}$ represents the quantum resources that are allowed during the computation. By restricting the set $\mathscr{A}$, one can get specific models of quantum computation such as measurement-based quantum

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

computation, or one-way quantum computation. A CQTM M in which $\mathscr{A}$ is restricted to projective measurements only corresponds to measurement-only quantum computation. It turns out that there is a lack of formal models of quantum computation for alternative models based on measurements. Moreover, classical control plays a crucial role in these promising models of quantum computation. The restriction of the CQTM to projective measurements produces such a formal model.

Definition 6 (Measurement-based quantum Turing machine). A Measurement-based quantum Turing machine (MQTM) is a CQTM $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$, where $\mathscr{A}$ is composed of projective measurements only.

The following theorem proves that any TM is simulated by a Las Vegas MQTM, that is, an MQTM with a non-probabilistic outcome, even if the execution time is unbounded.

Theorem 5. Given any TM M operating in time $f(n)$, where n is the input size, there exists an MQTM M' operating in expected time $O(f(n))$, and such that, for any input x, $M(x) = M'(|x\rangle)$.

Proof. The proof is similar to the simulation of any TM by means of a CQTM (see Theorem 1), but the permutation $\mathscr{P}_{[\tau,\sigma]}$ of two symbols, which is a unitary transformation, has to be simulated by means of projective measurements. The simulation consists of applying projective measurement $\mathscr{O}_{[\tau,\sigma]}$ in the diagonal basis, transforming the state $|\tau\rangle$ into a uniform superposition $(|\tau\rangle \pm |\sigma\rangle)/\sqrt{2}$. Then the cell is measured according to the standard basis producing $|\sigma\rangle$ with probability $1/2$ and $|\tau\rangle$ with probability $1/2$. If the simulation of the permutation fails, the sequence of the two measurements $\mathscr{O}_{[\tau,\sigma]}$ and Std is applied again, until successful.

Formally, if $M = (K, \Sigma, \delta)$, let $M' = (K', (\Sigma \cup \{\#, \lambda, \top, \bot\}), \Sigma, \{Std\} \cup \{\mathscr{O}_{[\tau,\tau']}\}_{\tau,\tau' \in \Sigma}, \delta')$. For any $(q, \tau) \in K \times \Sigma$, if $\delta(q, \tau) = (p, \sigma, D)$, we have

$$\begin{aligned}
\delta'(q, \tau) &= (q_{\tau,\sigma}, -, \mathscr{O}_{[\tau,\sigma]}) \\
\delta'(q_{\tau,\sigma}, -) &= (q'_{\tau}, -, Std) \\
\delta'(q'_{\tau,\sigma}, \sigma) &= (p, D, Std) \\
\delta'(q'_{\tau,\sigma}, \tau) &= (q_{\tau,\sigma}, -, \mathscr{O}_{[\tau,\sigma]}).
\end{aligned}$$

Each transition of M is simulated with constant expected execution time. □

According to Theorem 5, unitary transformations are not required to simulate classical computation. One may wonder how powerful MQTMs are compared with CQTMs. In Example 3, an MQTM simulating the application of a Hadamard transformation with probability 1 is described. The simulation is based on state transfer (Perdrix 2005). More generally, Theorem 6 proves that any CQTM can be efficiently simulated by an MQTM.

Example 3 (Measurement-based simulation of the Hadamard transformation). Consider the problem of simulating a given unitary transformation by means of measurements. As an example, we choose to simulate the Hadamard transformation H on a one-qubit input $|\phi\rangle$ using projective measurements only. Consider the 2-tape MQTM $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$, with $K = \{s, s', q', q''\} \cup \{q_{j(a,b)} : a, b \in \{-1, 1\}, j = 1 \ldots 5\}$, $\Sigma_Q = \{\#, 0, 1\}$,

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

	$p \in K,$	$\tau \in \Sigma_C$	$\delta(p,\tau)$
	s	$\#$	$(s', (\rightarrow,-), [-, \mathscr{O}_{[\#,0]}])$
$\forall t \in \{-1,0\}$	s'		$(q', (-,-), [-, Std])$
	q'	$\#$	$(s', (-,-), [-, \mathscr{O}_{O[\#,0]}])$
	q'	0	$(q'', (-,-), [-, \mathscr{O}_X])$
$\forall i \in \{-1,1\}$	q''	i	$(q_{1(i,1)}, (-,-), \mathscr{O}_{X\otimes Z})$
$\forall i,j \in \{-1,1\}$	$q_{1(i,1)}$	j	$(q_{2(i,j)}, (-,-), [\mathscr{O}_Z, -])$
$\forall i,j,k \in \{-1,1\}$	$q_{2(i,j)}$	k	$(q_{3(i,k,j)}, (-,-), [\mathscr{O}_X, -])$
$\forall k',j,l \in \{-1,1\}$	$q_{3(k',j)}$	k	$(q_{4(k',l,j)}, (-,-), \mathscr{O}_{Z\otimes Z})$
$\forall l',j,m \in \{-1,1\}$	$q_{4(l',j)}$	k	$(q_{5(l',j,m)}, (-,-), [-, \mathscr{O}_X])$
$\forall l',n \in \{-1,1\} : l' = n$	$q_{5(l',1}$	n	$(h, (-,-), -)$
$\forall l',m',n \in \{-1,1\} : m' \neq 1 \vee l' \neq n$	$q_{5(l',m')}$	n	$(q_{2(l',n,m')}, (-,-), \mathscr{O}_{Z\otimes Z})$

Fig. 3. 2-tape MQTM with projective measurements only for the simulation of H.

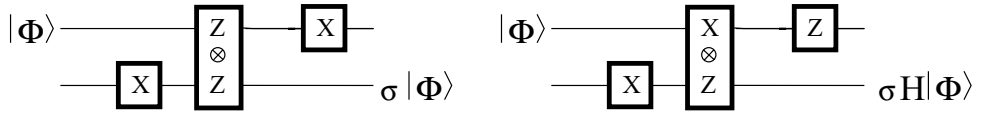


Fig. 4. *Left-hand diagram* – state transfer; *Right-hand diagram* – generalised state transfer for the simulation of H (see Perdrix (2005)).

$\Sigma_C = \{\#, \overline{\#}, -1, 0, 1, \top, \bot\}$, $\mathscr{A} = \{\mathscr{T}_\#, Std, \mathscr{O}_{[\#,0]}, \mathscr{O}_Z, \mathscr{O}_X, \mathscr{O}_{X\otimes Z}, \mathscr{O}_{Z\otimes Z}\}$ (where Z and X are Pauli matrices) and with δ as described in Figure 3.

The input state is placed on tape T_1. The first three transitions are used for transforming the state pointed at by the second head, from $|\#\rangle$ to $|0\rangle$. Then three projective measurements are performed according to generalised state transfer (Perdrix 2005) (see Figure 4): $H|\phi\rangle$, up to a Pauli operator, is placed on T_2. Since the result of the computation has to be located on T_1, the next three transitions transfer the result of the simulation from T_2 to T_1. Since state transfer is obtained up to a known Pauli operation (as in teleportation), internal states $q_{j(a,b)}$ are used to memorise the corrective operation: $q_{j(a,b)}$ means that the state transfer is obtained up to the Pauli operator

$$\sigma_z^{\frac{1-a}{2}} \sigma_X^{\frac{1-b}{2}}.$$

In order to correct this Pauli operator, the state is transferred twice: from T_1 to T_2, then back from T_2 to T_1. If M halts, $M(|\phi\rangle) = H|\phi\rangle$, but M may never halt, thus M is Las Vegas. Notice that $Pr([M(|\phi\rangle) = \nearrow] = 0$.

Theorem 6. Given any k-tape CQTM M operating in time $f(n)$, there exists a k-tape MQTM M' operating in time $O(f(n))$, such that for any input $|\psi\rangle$ of size n, $M(|\psi\rangle) = M'(|\psi\rangle)$.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

Proof. The proof consists of two stages: first, each admissible transformation used in the transition function of M is transformed into a unitary transformation immediately followed by a projective measurement; then, in a second stage, the unitary operation previously obtained is simulated by means of projective measurements.

If $M = (K, \Sigma_C, \Sigma_Q, \mathscr{A}, \delta)$, let $M' = (K', (\Sigma_C^2 \times \{\top, \bot\}), (\Sigma_Q \times \Sigma_C \times \{\top, \bot\}), \mathscr{A}', \delta')$.

The alphabet of the quantum cells of M' is composed of triplets (ϕ, c, r): $\phi \in \Sigma_Q$ is used to simulate the corresponding quantum cell of M; $c \in \Sigma_C$ is used in the first stage for transforming admissible transformations into unitary transformations; and $r \in \{\top, \bot\}$ is an element of the additional workspace needed to simulate any unitary transformations by means of projective measurements.

— For any $(q, c) \in K \times \Sigma_C$, if $\delta(q, c) = (p, D, A)$, let $\tilde{\delta}(q, c) = (q_c, -, \mathscr{U}_V)$ and $\tilde{\delta}(q_c, \lambda) = (p, D, \mathscr{O}_O)$. If $A = \{M_c, c \in \Sigma_C\}$, then $V : \mathscr{H}_{\Sigma_Q \times \Sigma_C} \to \mathscr{H}_{\Sigma_Q \times \Sigma_C}$ is a unitary transformation such that $V|\psi\rangle|\#\rangle = \sum_{c \in \Sigma_C} (M_c|\psi\rangle)|c\rangle$ (V can be extended to the case where the second register is not in state $|\#\rangle$), and O is a projective measurement of the second register in the standard basis $\{|c\rangle, c \in \Sigma_C\}$. If δ is replaced by $\tilde{\delta}$, an exact simulation of M is obtained.

— Let $R = V \otimes |\bot\rangle\langle\top| + V^\dagger \otimes |\top\rangle\langle\bot|$, $P_\top = (I + R)/2$, and $P_\bot = (I - R)/2$. Then $\mathscr{V} = \{P_\top, P_\bot\}$ is a projective measurement. One can show that for any $|\phi\rangle \in \mathscr{H}_{\Sigma_Q \times \Sigma_C}$, a measurement of $|\phi\rangle \otimes |\top\rangle$ according to $\mathscr{V}$, followed by a projective measurement according to $L = \{I \otimes |\top\rangle\langle\top|, I \otimes |\bot\rangle\langle\bot|\}$, produces $(V|\phi\rangle) \otimes |\bot\rangle$ with probability $1/2$, and $|\phi\rangle \otimes |\top\rangle$ with probability $1/2$. Then, for any $(q, c) \in K \times \Sigma_C$, if $\tilde{\delta}(q, c) = (q_c, -, \mathscr{U}_V)$ and $\tilde{\delta}(q_c, \lambda) = (p, D, \mathscr{O}_O)$, let δ' be such that

$$\begin{aligned} \delta'(q, c) &= (q'_c, -, \mathscr{O}_\mathscr{V}) \\ \delta'(q'_c, -) &= (q''_c, -, \mathscr{O}_L) \\ \delta'(q''_c, \top) &= (q'_c, -, \mathscr{O}_\mathscr{V}) \\ \delta'(q''_c, \bot) &= (p, D, \mathscr{O}_O). \end{aligned}$$

Thus, any transition of the CQTM M can be simulated on the MQTM M' within an expected constant number of transitions. As a consequence, M' simulates M with an expected execution time of $O(f(n))$. □

Since any k-CQTM is efficiently simulated by a k-tape MQTM, the results proved for CQTMs in previous sections also hold for MQTMs.

9. Conclusion

This paper has introduced classically controlled quantum Turing machines (CQTM) as a new abstract model for quantum computations. This model allows a rigorous formalisation of the inherent interactions between the quantum and classical worlds during a quantum computation. Any classical Turing machine is simulated by a CQTM without loss of efficiency. Moreover, any k-tape CQTM can be simulated by a 2-tape CQTM with only a polynomial effect on the execution time.

Moreover, we have pointed out a gap between classical and quantum computations in classically controlled quantum computations.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

We have also introduced MQTMs, which are a restriction of CQTMs to projective measurements. Any CQTM can be efficiently simulated by an MQTM. This formally proves the power of alternative models based on measurements only (Nielsen 2003; Leung 2004; Perdrix 2005).

The classically controlled quantum Turing machine is a good candidate for establishing a bridge between, on one side, theoretical models, such as QTMs and MQTMs, and on the other side, practical models of quantum computation, such as quantum random access machines.

References

Aharonov, D. (1998) *Quantum Computation*, Annual Reviews of Computational Physics, vol VI.

Bennett, C., Brassard, G., Crepeau, C., Jozsa, R., Peres, A. and Wootters, W. (1993) Teleporting an unknown quantum state via dual classical and EPR channels. *Phys Rev Lett* 1895–1899.

Bettelli, S., Serafini, L. and Calarco, T. (2003) Toward an architecture for quantum programming. *European Physical Journal* D **25** (2) 181–200.

Bernstein, E. and Vazirani, U. (1997) Quantum complexity theory. *SIAM J. Comput*. **26** 1411–1473.

Deutsch, D. (1985) Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A* **400** 97–117.

Danos, V., Kashefi, E. and Panangaden, P. (2004) The Measurement Calculus. arXiv:quant-ph/0412135.

Iriyama, S., Ohya, M. and Volovich, I. (2004) Generalized Quantum Turing Machine and its Application to the SAT Chaos Algorithm. arXiv:quant-ph/0405191.

Jorrand, Ph. and Lalire, M. (2004) Toward a quantum process algebra. In: *Proceedings of the 1st ACM Conference on Computing Frontiers*, ACM Press. (Also available as arXiv:quant-ph/0312067.)

Kitaev, A. Y., Shen, A. H. and Vyalyi, M. N. (2002) *Classical and Quantum Computation*, American Mathematical Society.

Knill, E. H. (1996) Conventions for Quantum Pseudocode. LANL report LAUR-96-2724.

Leung, D. W. (2004) Quantum computation by measurements. *International Journal of Quantum Information* **2** (1) 33–43

Miyadera, T. and Ohya, M. (2005) On Halting Process of Quantum Turing Machine. *Open Systems and Information Dynamics* **12** (3) 261–264.

Muthukrishnan, A. and Stroud, C. R. (2000) Multi-valued logic gates for quantum computation. *Phys. Rev. A* **62** 052309.

Nielsen, M. A. and Chuang, I. L. (2000) *Quantum Computation and Quantum Information*, Cambridge University Press.

Nielsen, M. A. (2003) Universal quantum computation using only projective measurement, quantum memory, and preparation of the 0 state. *Phys. Lett. A*. **308**.

Nishimura, H. and Ozawa, M. (2002) Computational Complexity of Uniform Quantum Circuit Families and Quantum Turing Machines. *Theoretical Computer Science* **276** 147–181.

Papadimitriou, C. M. (1994) *Computational Complexity*, Addison-Wesley.

Perdrix, S. (2005) State Transfer instead of Teleportation in Measurement-based Quantum Computation. *International Journal of Quantum Information* **3** (1).

Raussendorf, R. and Briegel, H. J. (2003) Measurement-based quantum computation with cluster states. *Phys. Rev. A* **68** 022312.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press

Selinger, P. (2004) Towards a quantum programming language. *Mathematical Structures in Computer Science* **14** (4) 527–586.

Selinger, P. and Valiron, B. (2005) A lambda calculus for quantum computation with classical control. In: Proceedings of the 7th International Conference on Typed Lambda Calculi and Applications (TLCA). *Springer-Verlag Lecture Notes in Computer Science* **3461**. (Also available as arXiv:cs.LO/0404056.)

Watrous, J. (2000) Succinct quantum proofs for properties of finite groups. In: *Proc. 41st Annual Symposium on Foundation of Computer Science* 537–546.

Yao, A. C. (1993) Quantum circuit complexity. In: *Proc. 34th IEEE Symposium on Foundation of Computer Science* 352–361.

https://doi.org/10.1017/S096012950600538X Published online by Cambridge University Press