# Chapter 4

# AUTOMATA

## INTRODUCTION

In addition to the study of problems of the design and analysis of algorithms and circuits, as well as of the computational complexity of algorithmic problems, another main method of theoretical computing to get an insight into the power of computational resources is to study models of quantum computing devices and the corresponding complexity classes. This will be done in this chapter for three most basic models of quantum automata: quantum versions of finite automata, Turing machines and cellular automata.

Quantum finite automata are perhaps the most elementary model of quantum automata. They are in addition the only model so far for which it has been fully proved that they have larger power than their classical counterparts.

Quantum Turing machines are the main model to study the most fundamental questions concerning the power of quantum computing itself and the power of quantum versus classical computing. Quantum cellular automata are of a special interest. They seem to be a model much closer to the physical reality than quantum Turing machines. In addition, it is still a major open problem whether quantum cellular automata are more powerful than quantum Turing machines.

## LEARNING OBJECTIVES

The aim of the chapter is to learn:

1. the basic models of one-way and two-way quantum finite automata as well as their computational power and space efficiency in comparison with classical finite automata;

2. the basic models of quantum Turing machines and methods of implementing the main classical programming primitives in a reversible way;

3. the types and precision of amplitudes needed to ensure sufficient accuracy of computation on quantum Turing machines;

4. the basic models of quantum cellular automata and partitioned quantum cellular automata, as well as simulations between quantum cellular automata and quantum Turing machines.

> By convention there is color, by convention sweetness, by convention bitterness, but in reality there are atoms and space.
>
> Democritus (460–400 BC)

Three basic models of quantum automata are described in detail in this chapter. Each of them plays a special role in the study of the power and limitations of quantum computing.

One-way quantum finite automata can be seen as the simplest model of quantum automata that can also quite well reflect the simplest quantum computers we can expect first. In spite of their simplicity their descriptional efficiency can be remarkable. Two-way quantum finite automata are of interest because they are the first model of finite automata more powerful in polynomial time than classical deterministic finite automata.

With quantum Turing machines the most fundamental questions concerning the ultimate power and limitations of quantum computing can be studied. For example, the concept of universality and the question whether quantum Turing machine should be seen as analogue or discrete devices.

A suitable definition of more dimensional quantum cellular automata is still an important open problem. For one-dimensional cellular automata the problem has been resolved fully satisfactorily and it is with this type of quantum cellular automata we deal in this chapter. We could define in a similar way more dimensional quantum cellular automata but in such a case there are problems to decide whether a given specification of to-be quantum automaton really defines a quantum automaton.

There are two main reasons why to come with an easy to deal definition of two and more dimensional quantum cellular automata is an important and interesting task. Intuitively, two- and three-dimensional quantum cellular automata could much better reflect the nature of the quantum world than quantum Turing machines. In addition, there is still a chance that they could be computationally more powerful than quantum Turing machines.

The way quantum analogues of classical automata models are defined is already pretty standard, at least its basic steps. A quantum automaton is described formally in a similar way to a probabilistic one, only instead of the probabilities of transitions we use probability amplitudes. Evolution takes place then in the inner-product space over the set of finite configurations, which then represent the states of the (standard) basis. The main problem now is that evolution of each quantum automaton has to be unitary. Given a description of a to-be-quantum automaton, it can be far from trivial to decide whether its evolution satisfies the unitarity condition. One way out is to discover some more easy-to-check conditions, usually called "well-formedness conditions", satisfiability of which implies unitarity of the evolution.

## 4.1 Quantum Finite Automata

Just as classical finite automata serve as the very basic model of classical finite-size machines, so can quantum finite automata be seen as basic quantum models of finite state quantum machines.

### 4.1.1 Models of classical finite automata

A variety of models of finite state machines and automata have been developed and investigated in the classical theory of computation. Each of these models captures some mode of actions, or its natural abstraction, of the intuitive concept of computing with finite-size machines.

The very basic model of finite automata is that of the input-driven **deterministic finite automaton** (DFA) $\mathcal{A} = \langle Q, \Sigma, q_0, Q_a, \delta \rangle$, where $Q$ is the finite set of states, $\Sigma$ is the input alphabet, $q_0$ is the initial state, $Q_a$ is the set of final (accepting) states and $\delta : Q \times \Sigma \rightarrow Q$ is the **transition function** such that for a given input symbol $\sigma$ and the state $q$, $\delta(q, \sigma)$ is the new state of the control unit—see Figure 4.1a. Another view of DFA is that of a **one-way finite automaton** (1FA, see Figure 4.1b), with the finite control unit connected to the input tape by a read-only head moving only in a left-to-right way. A natural generalization is that of a **two-way finite automaton** (see Figure 4.1c), with the finite-control unit attached to the input tape by a read-only head which can move in both directions. The transition function of such a 2FA has the form $\delta : Q \times \Sigma \rightarrow Q \times \{\leftarrow, \downarrow, \rightarrow\}$.

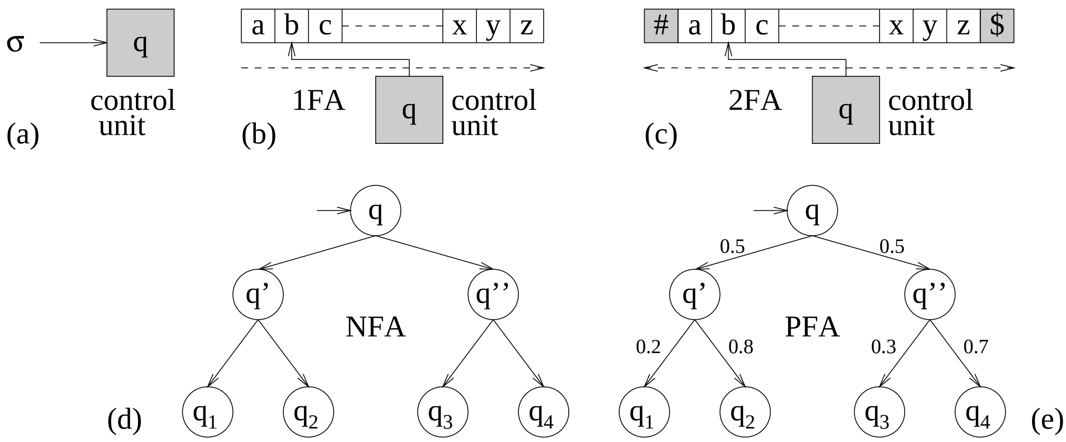


Figure 4.1: Models of finite automata-70%

Two other generalizations of FA are of importance. In a **nondeterministic finite automaton**, (NFA, see Figure 4.1d for a transition tree), there is in general a nondeterministic choice of the next state, and the transition function has the form $\delta : Q \times \Sigma \times Q \rightarrow \{0, 1\}$ (or $\delta : Q \times \Sigma \rightarrow \{0, 1\}^Q$). In **probabilistic finite automata** (PFA or 1PFA, see Figure 4.1e for a transition tree), to each transition a probability is assigned; i.e. the transition function has the form $\delta : Q \times \Sigma \times Q \rightarrow [0, 1]$ (or $\delta \in Q \times \Sigma \rightarrow [0, 1]^Q$), such that for each $q \in Q$ and $\sigma \in \Sigma$ the following **local probability condition** is satisfied: $\sum_{q' \in Q} \delta(q, \sigma, q') = 1$.

One way to compare computational power of various types of finite automata is to consider them as **acceptors** of the set of those input words that make the automaton reach a final state after reading the whole input. One of the main results of the automata theory is that one-way and two-way deterministic and nondeterministic finite automata accept the same family of languages, namely regular languages. Probabilistic finite automata are a special case. With respect to the bounded-error acceptance 1PFA can accept only regular languages and so do 2PFA, a probabilistic version of 2FA, if required to work in linear time with respect to the size of input. (On the other hand, with respect to the unbounded-error acceptance and arbitrary real probabilities 1PFA can accept also non-regular languages.)

### 4.1.2 One-way quantum finite automata

Two main models of quantum finite automata (QFA) will be presented in this section. One-way quantum automata (1QFA) as a quantum analogue of 1FA and two-way quantum automata (2QFA) as an analogue of 2FA.

One-way quantum finite automata seem to model very well the way very simple quantum processors work, (see Ambainis and Freivalds, 1998), and also the way simple classical/quantum processors are expected to work: the classical part reads an input, picks up the corresponding quantum operator (a transition mapping) and performs it on a quantum memory of fixed size, independent of the size of input. 1QFA are very simple but provably less powerful than classical 1FA.

Two-way quantum finite automata, on the other hand, can accept more than regular languages. The price to pay is the "size of quantum memory" needed. A 2QFA can be in a large superposition of its basis states each of which corresponds to a position of the head on a different symbol of the input. The size of quantum memory can therefore be proportional to the size of input.

#### Basic concepts

One-way quantum finite automata are the first type of quantum automata we consider and therefore we shall deal with them in more detail.

**Definition 4.1.1** *A* **one-way quantum finite automaton** $\mathcal{A}$ *is specified by the finite (input) alphabet* $\Sigma$*, the finite set of states* $Q$*, the initial state* $q_0$*, the sets* $Q_a \subseteq Q$ *and* $Q_r \subseteq Q$ *of accepting and rejecting states, respectively, with* $Q_a \cap Q_r = \emptyset$*, and the transition function*

$$\delta : Q \times \Gamma \times Q \to \mathbf{C}_{[0,1]},$$

*where* $\Gamma = \Sigma \cup \{\#, \$\}$ *is the tape alphabet of* $\mathcal{A}$*, and* $\#$ *and* $\$$ *are endmarkers not in* $\Sigma$*.*

*The evolution (computation) of* $\mathcal{A}$ *is performed in the inner-product space* $l_2(Q)$*, i.e., with the basis* $\{|q\rangle \mid q \in Q\}$*, using the linear operators* $V_\sigma$*,* $\sigma \in \Gamma$*, defined by*

$$V_\sigma(|q\rangle) = \sum_{q' \in Q} \delta(q, \sigma, q')|q'\rangle,$$

*which are required to be unitary.*

In computation of $\mathcal{A}$ the so-called computational observable $\mathcal{O}$ is used that corresponds to the orthogonal decomposition

$$l_2(Q) = E_a \oplus E_r \oplus E_n,$$

where $E_a = span\{|q\rangle \mid q \in Q_a\}$, $E_r = span\{|q\rangle \mid q \in Q_r\}$ and $E_n$ is the orthogonal complement of $E_a \oplus E_r$. Denote by $P_p$, $p \in \{a, r, n\}$ the projection operator into the subspace $E_p$.

A **computation of** $\mathcal{A}$ **on an input** $\#\sigma_1 \ldots \sigma_n\$$ proceeds as follows. The operator $V_\#$ is first applied to the starting configuration $|q_0\rangle$ and then the observable $\mathcal{O}$ is applied to the resulting configuration $V_\#|q_0\rangle$. This observable projects $V_\#|q_0\rangle$ into a vector $|\psi'\rangle$ of one of the subspaces $E_a, E_r, E_n$, with the probability equal to the square of the norm of $|\psi'\rangle$. If $|\psi'\rangle \in E_a$ the input is accepted; if $|\psi'\rangle \in E_r$ the input is rejected. If $|\psi'\rangle \in E_n$, then, after the normalization of $|\psi'\rangle$, the operator $V_{\sigma_1}$ is applied to $|\psi'\rangle$ and after that the observable

$\mathcal{O}$ to the resulting vector. This process goes on. Operators $V_{\sigma_1}, V_{\sigma_2}, \ldots, V_{\sigma_n}$ are applied one after another, and after each such application the measurement by the observable $\mathcal{O}$ is performed. In all cases the computation continues only if a projection into $E_n$ occurs.

When no termination occurs the computation can be seen as an application of the composed operator

$$V'_{\sigma_n} V'_{\sigma_{n-1}} \ldots V'_{\sigma_1} |q_0\rangle,$$

where $V'_{\sigma_i} = P_n V_{\sigma_i}$.

**Language recognition by 1QFA**

In order to define formally the overall probability with which an input is accepted (rejected) by a 1QFA $\mathcal{A}$, we define the set $V_{\mathcal{A}} = l_2(Q) \times \mathbf{C} \times \mathbf{C}$ of so-called "total states" of $\mathcal{A}$, that will be used only with the following interpretation. $\mathcal{A}$ is at any time during the computation in the state $(\psi, p_a, p_r)$ if so far in its computation $\mathcal{A}$ accepted the input with probability $p_a$, rejected with probability $p_r$ and neither with probability $1 - p_a - p_r = ||\psi||^2$, and $|\psi\rangle$ is its current, unnormalized state. For each $\sigma \in \Gamma$ the evolution of $\mathcal{A}$, with respect to the total state, on an input $\sigma$ is given by the operator $T_\sigma$ defined by

$$T_\sigma(\psi, p_a, p_r) \to (P_n V_\sigma \psi, p_a + ||P_a V_\sigma \psi||^2, p_r + ||P_r V_\sigma \psi||^2).$$

For $x = \sigma_1 \sigma_2 \ldots \sigma_n \in \Gamma^*$ let $T_{\#x\$} = T_\$ T_{\sigma_n} T_{\sigma_{n-1}} \ldots T_{\sigma_1} T_\#$. If $T_{\#x\$}(|q_0\rangle, 0, 0) = (\psi, p_a, p_r)$, then we say that $\mathcal{A}$ accepts $x$ with probability $p_a$ and rejects with probability $p_r$.

A 1QFA $\mathcal{A}$ is said to accept a language $L$ with probability $\frac{1}{2} + \varepsilon$, $\varepsilon > 0$, if it accepts any $x \in L$ with probability at least $\frac{1}{2} + \varepsilon$ and rejects any $x \notin L$ with probability at least $\frac{1}{2} + \varepsilon$. If there is an $\varepsilon > 0$ such that $\mathcal{A}$ accepts $L$ with probability at least $\frac{1}{2} + \varepsilon$, then $L$ is said to be accepted by $\mathcal{A}$ with bounded-error probability. $L$ is said to be accepted with unbounded error probability if $x \in L$ is accepted with probability at least $\frac{1}{2}$ and $x \notin L$ rejected with probability at least $\frac{1}{2}$.

**Example 4.1.2** *(Ambainis and Freivalds, 1998) We show that the language $L_0 = \{0^i 1^j \,|\, i, j \geq 0\}$ can be recognized by a 1QFA $\mathcal{A}_0$ with probability $p$ such that $p = 1 - p^3$ ($p \approx 0.68$).*

*$\mathcal{A}_0$ has the set of states $Q = \{q_0, q_1, q_2, q_a, q_r\}$, $Q_a = \{q_a\}$, $Q_r = \{q_r\}$. $q_0$ is the initial state. The transition function $\sigma$ is specified by transitions:*

$$V_\# |q_0\rangle = \sqrt{1-p}|q_1\rangle + \sqrt{p}|q_2\rangle,$$
$$V_0|q_1\rangle = (1-p)|q_1\rangle + \sqrt{p(1-p)}|q_2\rangle + \sqrt{p}|q_r\rangle,$$
$$V_0|q_2\rangle = \sqrt{p(1-p)}|q_1\rangle + p|q_2\rangle - \sqrt{1-p}|q_r\rangle,$$
$$V_1|q_1\rangle = |q_r\rangle, V_1|q_2\rangle = |q_2\rangle,$$
$$V_\$|q_1\rangle = |q_r\rangle, V_\$|q_2\rangle = |q_a\rangle,$$

*and the remaining transitions are defined arbitrarily so that unitarity requirements are satisfied.*

**Case 1** $x = 0^i$. *Since $V_0(\sqrt{1-p}|q_1\rangle + \sqrt{p}|q_2\rangle) = \sqrt{1-p}|q_1\rangle + \sqrt{p}|q_2\rangle$, $\mathcal{A}_0$ remains in the state $\sqrt{1-p}|q_1\rangle + \sqrt{p}|q_2\rangle$ while reading $0^i$. At the right endmarker the operator $V_\$$ is applied to yield the state $\sqrt{1-p}|q_r\rangle + \sqrt{p}|q_a\rangle$. Therefore $\mathcal{A}_0$ accepts with probability $p$.*

**Case 2.** $x = 0^i 1^j$, $i \geq 0, j > 0$. *As in Case 1, $\mathcal{A}_0$ is in the state $\sqrt{1-p}|q_1\rangle + \sqrt{p}|q_2\rangle$ after reading $0^i$. The first 1 on the input changes the state into $\sqrt{1-p}|q_r\rangle + \sqrt{p}|q_2\rangle$. The* *nonhalting part is* $\sqrt{p}|q_2\rangle$. *It is unchanged by the following* $1$*s until the right endmarker changes it to* $\sqrt{p}|q_a\rangle$. $\mathcal{A}_0$ *therefore again accepts with probability* $p$.

**Case 3.** $x$ *is not in* $\{0^i1^j \,|\, i \geq 0, j \geq 0\}$. *Hence there is an initial substring of* $x$ *of the form* $0^i1^j0^k$, *where* $i \geq 0$, $j > 0$, $k > 0$. *After reading the first* $1$, $\mathcal{A}_0$ *is in the state* $\sqrt{1-p}|q_r\rangle + \sqrt{p}|q_2\rangle$ *and the automaton rejects with probability* $1-p$. *The nonhalting part,* $\sqrt{p}|q_2\rangle$, *is not changed by the following* $1$*s and the first* $0$ *changes it into* $p\sqrt{1-p}|q_1\rangle + p\sqrt{p}|q_2\rangle - \sqrt{p(1-p)}|q_r\rangle$. *At this moment* $\mathcal{A}_0$ *rejects with probability* $p(1-p)$. *The total probability that* $x$ *is rejected is therefore at least*

$$(1-p) + p(1-p) + p^2(1-p) = 1 - p^3 = p.$$

**Exercise 4.1.3** *What is the largest probability with which a 1QFA can accept the language* $\{0^i1^j0^k \,|\, i,j,k \geq 0\}$*?*

On $V_{\mathcal{A}}$ we define a "norm" $||\cdot||_u$ as follows

$$||(\psi, p_a, p_r)||_u = \frac{1}{2}(||\psi|| + |p_a| + |p_r|)$$

and let $\mathcal{D} = \{v \in V_{\mathcal{A}} \,|\, ||v||_u \leq 1\}$. $\mathcal{D}$ contains all global states of $\mathcal{A}$.

**Exercise 4.1.4** *Show: (a)* $||\cdot||_u$ *is a norm on* $V_{\mathcal{A}}$*; (b) there is a constant* $c$ *such that* $||T_x v - T_x v'||_u \leq c||v - v'||_u$ *for any* $v, v' \in V_{\mathcal{A}}$, $x \in \Gamma^*$*; (c) if a set* $A \subset \mathcal{D}$ *is such that there exists an* $\varepsilon > 0$ *such that for all* $v, v' \in \mathcal{A}$ *it holds* $||v - v'||_u > \varepsilon$, *then there can be only finitely many elements in* $A$.

**Remark 4.1.5** Another possibility to define acceptance by one-way QFA was considered by Moore and Crutchfield (1997)—to measure only at the right endmarker. Let us denote one-way QFA with such acceptance by mc-1QFA. Even if this definition seems to be at first sight more natural, technical results do not confirm that. It holds mc-QFA $\subsetneq 1QFA$.

**Exercise 4.1.6** *Show that if a language is accepted by a mc-1QFA, it is accepted by a 1QFA, but that the opposite statement does not have to be true.*

**Exercise 4.1.7** *Explore closure properties of languages accepted by mc-QFA.*

### 4.1.3 1QFA versus 1FA

Two basic problems concerning one-way quantum automata are dealt with in this section: their power and space efficiency.

#### Language recognition

The basic question is whether 1QFA have larger recognition power than 1FA. The answer is negative (Kondacs and Watrous, 1997).

**Theorem 4.1.8** *If a language $L$ is accepted by a 1QFA with bounded-error probability, then $L$ is regular.*

**Proof.** The proof is a small modification of the proof that probabilistic FA accept only regular languages, see Rabin (1976). Here is the basic idea:

Let $\mathcal{A} = \langle Q, \Sigma, q_0, Q_a, Q_r, \delta \rangle$ be a 1QFA recognizing the language $L$ with probability $\frac{1}{2} + \varepsilon$.

For $w, w' \in \Sigma^*$ we define that the prefix relation $w \equiv_L w'$ holds if for all $y \in \Sigma^*$, $wy \in L$ if and only if $w'y \in L$. It is well known (see, for example Gruska, 1997), that a language $L$ is regular if and only if there are only finitely many equivalence classes with respect to its prefix equivalence.

Let $W \subseteq \Sigma^*$ be *any* set of strings that are mutually inequivalent with respect to the equivalence $\equiv_L$. If we prove that $W$ is finite, the theorem will be proved. This can be done as follows.

If $w, w' \in W$, $w \not\equiv_L w'$, then there must exist a $y$ such that only one of the strings $wy$ and $w'y$ is in $L$. Therefore, for $v = T_{\#w}(|q_0\rangle, 0, 0)$ and $v' = T_{\#w'}(|q_0\rangle, 0, 0)$ it has to hold:

$$||T_{y\$}v - T_{y\$}v'||_u > 2\varepsilon,$$

because $\mathcal{A}$ accepts $L$ with bounded-error probability $\varepsilon$. Consequently, by Exercise 4.1.4, $||v - v'||_u \geq \frac{2\varepsilon}{c}$ for some constant $c$ that does not depend on $\varepsilon$. This implies, by Exercise 4.1.4 again, that the set $W$ has to be finite. □

The next basic question is whether 1QFA have the same recognition power as 1FA. The answer is again negative (Kondacs and Watrous, 1997).

**Theorem 4.1.9** *The regular language $L_0 = \{0,1\}^*0$ cannot be recognized by a 1QFA with bounded-error probability.*

**Proof.** The proof is by contradiction. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, Q_a, Q_r \rangle$ be a 1QFA recognizing the language $L_0$. To each $x = \sigma_1 \ldots \sigma_n \in \Gamma^*$ we assign the state

$$|\psi_x\rangle = V'_{\sigma_n} \ldots V'_{\sigma_1}|q_0\rangle$$

and let $\mu = \inf_{w \in \{0,1\}^*}\{||\psi_{\#w}||\}$. For each $w \in \{0,1\}^*$, $w0 \in L_0$ and $w1 \notin L_0$. If $\mu = 0$, then clearly $\mathcal{A}$ cannot recognize $L_0$ with bounded-error probability $\frac{1}{2} + \varepsilon$. Let us therefore assume that $\mu > 0$.

For any $\varepsilon > 0$ there is a $w$ such that $||\psi_{\#w}|| < \mu + \varepsilon$, and also $||\psi_{\#wy}|| \in [\mu, \mu + \varepsilon]$ for any $y \in \{0,1\}^*$. In particular, for any $m > 0$

$$||{V'_1}^m|\psi_{\#w0}\rangle|| \in [\mu, \mu + \varepsilon). \tag{4.1}$$

This implies that the sequence $\{{V'_1}^i\,|\,\psi_{\#w0}\rangle\}_{i=0}^{\infty}$ is bounded in the finite dimensional inner-product space and must have a limit point. Therefore there have to exist $j$ and $k$ such that

$$||{V'_1}^j(|\psi_{\#w0}\rangle - {V'_1}^k|\psi_{\#w0}\rangle)|| < \varepsilon.$$

The last inequality together with (4.1) imply (see Lemma 4.1.10), that there is a constant $c$ which does not depend on $\varepsilon$ and such that

$$\Big|\Big||\psi_{\#w0}\rangle - {V'_1}^k|\psi_{\#w0}\rangle\Big|\Big| < c\varepsilon^{1/4}.$$

This implies that

$$||T_{\#w0\$}(|q_0\rangle, 0, 0) - T_{\#w01^k\$}(|q_0\rangle, 0, 0)||_u < c'\varepsilon^{1/4}$$

for fixed $c'$. However, this has to be valid for an arbitrarily small $\varepsilon$. This is not possible if $\mathcal{A}$ accepts $L_0$ because $\mathcal{A}$ should accept the string $w0$ and reject $w01^k$. Hence $\mathcal{A}$ cannot accept $L_0$ with bounded-error probability. ∎

**Lemma 4.1.10** *(J. Watrous) If $|u\rangle$ and $|v\rangle$ are vectors such that for a linear operator $A$, reals $0 < \varepsilon < 1$ and $\mu > 0$, $||A(u-v)|| < \varepsilon$, and $||v||$, $||u||$, $||Au||$, $||Av||$ are in $[\mu, \mu+\varepsilon]$, then there is a constant c, that does not depend on $\varepsilon$, such that $||u-v|| < c\varepsilon^{1/4}$.*

**Proof.** First observe that if we can show

$$||u-v||^2 < ||A(u-v)||^2 + c'\sqrt{\varepsilon} \tag{4.2}$$

then we have our proof because the square root of the right-hand side in (4.2) is smaller than $\sqrt{\varepsilon^2 + c'\sqrt{\varepsilon}} \leq \sqrt{c'+1}\,\varepsilon^{1/4}$, since $\varepsilon < 1$. To show (4.2) let us compute

$$\begin{aligned}
||u-v||^2 - ||A(u-v)||^2 &= \langle u-v|u-v\rangle - \langle Au-Av|Au-Av\rangle \\
&= ||u||^2 + ||v||^2 - \langle u|v\rangle - \langle v|u\rangle \\
&\quad -||Au||^2 - ||Av||^2 + \langle Au|Av\rangle + \langle Av|Au\rangle \\
&= (I) + (II) + (III) + (IV),
\end{aligned}$$

where

$$\begin{aligned}
(I) &= ||u||^2 - ||Au||^2, & (II) &= ||v||^2 - ||Av||^2, \\
(III) &= \langle u|A^*Av - v\rangle, & (IV) &= \langle v|A^*Au - u\rangle.
\end{aligned}$$

Since $||u||, ||v|| \leq [\mu, \mu+\varepsilon]$, we have

$$||u||^2 - ||Au||^2 \leq 2\mu\varepsilon + \varepsilon^2, \qquad ||v||^2 - ||Av||^2 \leq 2\mu\varepsilon + \varepsilon^2.$$

In order to get an estimate for $(III)$ and $(IV)$ we proceed as follows:

$$\begin{aligned}
||A^*Au - u||^2 &\leq ||A^*Au||^2 + ||u||^2 - 2||Au||^2 \\
&\leq ||u||^2 - ||Au||^2 \leq 2\mu\varepsilon + \varepsilon^2,
\end{aligned}$$

because $||A^*|| = ||A|| \leq 1$ and therefore

$$\langle v|A^*Au - u\rangle \leq ||v||\,||A^*Au - u|| \leq (\mu+\varepsilon)\sqrt{2\mu\varepsilon + \varepsilon^2}.$$

A similar estimation holds for $(III)$. Altogether we have

$$||u-v||^2 - ||A(u-v)||^2 \leq \sqrt{\varepsilon}[2(2\mu\sqrt{\varepsilon} + \varepsilon^{3/2}) + 2(\mu+\varepsilon)\sqrt{2\mu+\varepsilon})].$$

By taking $c' > 2(2\mu\sqrt{\varepsilon} + \varepsilon^{3/2}) + 2(\mu+\varepsilon)\sqrt{2\mu+\varepsilon}$, for example $c' = 10$, we get the inequality (4.2). ∎

Another natural type of automata one should try to compare with 1QFA are one-way reversible finite automata (1RFA). They can be defined as 1QFA having transition amplitudes either 0 or 1. The following result, due to Ambainis and Freivalds (1998), shows that if 1QFA are required to give the correct answer with high probability, then they have the same recognition power as 1RFA.

**Theorem 4.1.11** *A language can be recognized by a 1QFA with probability $\frac{7}{9}+\varepsilon$ if and only if it can be recognized by 1RFA.*

$\{0^i1^j \,|\, i,j \geq 0\}$ is an example of a language which can be recognized by a 1QFA with probability 0.68 (see Example 4.1.2), but not with probability $\frac{7}{9}+\varepsilon$.

#### Size-space efficiency

The number of states is a natural space measure for both 1FA and 1QFA. A natural related problem to investigate is the following one: if a language $L$ is accepted by a 1QFA, then how many states does a minimal 1QFA recognizing $L$ have in comparison with minimal 1FA recognizing $L$? The results obtained by Ambainis and Freivalds (1998) and Ambainis et al. (1998) show that sometimes a 1QFA is exponentially smaller than any 1FA recognizing the same language, but sometimes a 1FA is almost exponentially smaller than any 1QFA recognizing the same language. It holds:

**Theorem 4.1.12** *(1) Let $p$ be prime. Any 1FA recognizing the language $L_p = \{0^i \,|\, i$ is divisible by $p\}$ has to have $p$ states, but for any $\varepsilon > 0$ there is a 1QFA $\mathcal{A}_{p,\varepsilon}$, with $\mathcal{O}(\lg p)$ states accepting $L_p$ with probability $\frac{1}{2}+\varepsilon$.*

*(2) Let $n \geq 1$ be an integer. The language $L_n = \{w0 \,|\, w \in \{0,1\}^*, |w| \leq n\}$ can be recognized by a 1FA with $2n+3$ states but for any $\varepsilon > 0$ any 1QFA recognizing $L_n$ with probability $\frac{1}{2}+\varepsilon$ has to have $2^{\Omega(n/\lg n)}$ states.*

It is easy to design a $(2n+3)$-state 1FA accepting the language $L_n$. Concerning the proof of the lower bound on the number of states of any 1QFA for $L_n$, the very basic idea is simple. Since a 1QFA can read each input symbol only once, any 1QFA for $L_n$ which is necessary reversible is forced to "remember" all symbols read until it is clear whether the input symbol is in the language. Consequently, the state the automaton reaches after reading $n$ input symbols has to be an encoding of these $n$ symbols. Since this encoding has to be such that any input $n$-bit word can be recovered, the number of states has to be at least $2^n$. Unfortunately, the above idea is not fully valid because a 1QFA can terminate before it reads all inputs. The problem is therefore more complex and consequently the proof is more involved. It makes use of the ideas of dense coding discussed in Section 8.2.4.

It follows from the above theorem that in some cases the requirement of unitarity makes 1QFA much larger than the minimal equivalent 1FA.

### 4.1.4 Two-way quantum finite automata

The model of two-way quantum finite automata (2QFA) due to Watrous (1997c), presented in this section is a natural quantum analogue of the classical two-way finite automata. 2QFA are provably more powerful than their classical counterparts.

**Basic concepts**

Definition of two-way QFA is significantly more complex than of a 1QFA, especially due to the need to make sure their evolutions are unitary.

**Definition 4.1.13** *A two-way quantum finite automaton $\mathcal{A}$ is specified by the finite (input) alphabet $\Sigma$, the finite set of states $Q$, the initial state $q_0$, the sets $Q_a \subset Q$ and $Q_r \subset Q$ of accepting and rejecting states, respectively, with $Q_a \cap Q_r = \emptyset$, and the transition function*

$$\delta : Q \times \Gamma \times Q \times \{\leftarrow, \downarrow, \rightarrow\} \longrightarrow \mathbf{C}_{[0,1]},$$

*where $\Gamma = \Sigma \cup \{\#, \$\}$ is the tape alphabet of $\mathcal{A}$ and $\#$ and $\$$ are endmarkers not in $\Sigma$, which satisfies the following conditions (of well-formedness) for any $q_1, q_2 \in Q$, $\sigma, \sigma_1, \sigma_2 \in \Gamma$, $d \in \{\leftarrow, \downarrow, \rightarrow\}$:*

1. **Local probability and orthogonality condition.** $\sum_{q',d} \delta^*(q_1, \sigma, q', d)\delta(q_2, \sigma, q', d) = \begin{cases} 1, & \text{if } q_1 = q_2; \\ 0, & \text{otherwise.} \end{cases}$
2. **Separability condition I.** $\sum_{q'} \delta^*(q_1, \sigma_1, q', \rightarrow)\delta(q_2, \sigma_2, q', \downarrow) = 0.$
3. **Separability condition II.** $\sum_{q'} \delta^*(q_1, \sigma_1, q', \downarrow)\delta(q_2, \sigma_2, q', \leftarrow) = 0.$
4. **Separability condition III.** $\sum_{q'} \delta^*(q_1, \sigma_1, q', \rightarrow)\delta(q_2, \sigma_2, q', \leftarrow) = 0.$

*(These conditions are equivalent, as shown below, to the requirement that evolution of $\mathcal{A}$ is unitary.) Formally, $\mathcal{A} = \langle \Sigma, Q, q_0, Q_a, Q_r, \delta \rangle$.*

States from $Q_a \cup Q_r$ are called halting states and states from $Q_n = Q - (Q_a \cup Q_r)$ are called non-halting states.

In order to process an input word $x \in \Sigma^*$ by $\mathcal{A}$ we assume that the input is written on the tape with the endmarkers in the form $w_x = \#x\$$ and that such a tape, of length $|x| + 2$, is circular, i.e. the symbol to the right of \$ is #.[1]

For an integer $n$ let $C_n$ be the set (of size $(n+2)|Q|$) of all possible configurations of $\mathcal{A}$ for inputs of length $n$. For each specific input $x$ such a configuration is uniquely determined by a pair $(q, k)$, where $q$ is the state of the configuration and $0 \leq k \leq |x| + 1$ denotes the position of the head.

To a computation of $\mathcal{A}$ on an input $x$ of length $n$ corresponds a unitary evolution in the underlying inner-product space $\mathcal{H}_{\mathcal{A},n} = l_2(C_n)$. For each configuration $c \in C_n$, $|c\rangle$ denotes the basis vector in $l_2(C_n)$. Each state in $\mathcal{H}_{\mathcal{A},n}$ will therefore have a form

$$\sum_{c \in C_n} \alpha_c |c\rangle \quad \textit{where} \quad \sum_{c \in C_n} |\alpha_c|^2 = 1.$$

The automaton $\mathcal{A}$ induces for any input $x \in \Sigma^n$ a linear operator $U_x^\delta$ on $\mathcal{H}_{\mathcal{A},n}$ defined for a configuration $(q, k) \in C_n$ by

$$U_x^\delta |q, k\rangle = \sum_{q',d} \delta(q, w_x(k), q', d) \, | \, q', (k + \mu(d)) \bmod (n+2)\rangle,^2$$

where $w_x(k)$ denotes the $k$th symbol of $w_x = \#x\$$ for $0 \leq k \leq |x| + 1$. By linearity $U_x^\delta$ is extended to map any superposition of basis states.

[1] The requirement of circularity of the tape is not essential but slightly simplifies the treatment of 2QFA.
[2] By definition, $\mu(d) = -1(0)[1]$ if $d = \leftarrow (\downarrow)[\rightarrow]$.

**Lemma 4.1.14** *For any nonempty input string $x$ the mapping $U_x^\delta$ is unitary if and only if the conditions (1) to (4) of Definition 4.1.13 are satisfied.*

**Proof.** To prove the lemma, it is sufficient to investigate the orthogonality of the vectors $U_x^\delta|q,k\rangle$ for $q \in Q$, $0 \leq k \leq |x|+1$. The condition (1) is equivalent to the statement that for every $x$, $|x| \geq 0$, $||U_x^\delta|q,k\rangle|| = 1$ for all $q$ and $k$ and that $U_x^\delta|q_1,k\rangle \perp U_x^\delta|q_2,k\rangle$ for $q_1 \neq q_2$. Conditions (2) to (4) are equivalent to the statement that $U_x^\delta|q_1,k_1\rangle \perp U_x^\delta|q_2,k_2\rangle$ if $k_1$ and $k_2$ differ at most by 2. Finally, it is trivially true that $U_x^\delta|q_1,k_1\rangle \perp U_x^\delta|q_2,k_2\rangle$ if the head positions are more than two cells away because the heads can move only one cell per step. $\Box$

Verification of well-formedness condition of a 2QFA can be in general a quite cumbersome task, but it is easy for so-called simple 2QFA.

Informally, a 2QFA is simple if its transitions (amplitudes) are uniquely determined by transitions (amplitudes) between current states and the states the automaton comes into and, in addition, the move of the head is always uniquely determined by the new state only.

**Definition 4.1.15** *A 2QFA $\mathcal{A} = \langle \Sigma, Q, q_0, Q_a, Q_r, \delta\rangle$ is* **simple**, *or* **unidirectional**, *if for each $\sigma \in \Gamma$ there is a linear unitary operator $V_\sigma$ on the inner-product space $l_2(Q)$ and a function $D : Q \rightarrow \{\leftarrow, \downarrow, \rightarrow\}$ such that for each $q \in Q, \sigma \in \Gamma$,*

$$\delta(q,\sigma,q',d) = \begin{cases} \langle q'|V_\sigma|q\rangle, & \text{if } D(q') = d; \\ 0, & \text{otherwise,} \end{cases} \tag{4.3}$$

*(where $\langle q'|V_\sigma|q\rangle$ denotes the coefficient at $|q'\rangle$ in $V_\sigma|q\rangle$).*

It is straightforward to verify that if we rewrite the well-formedness conditions using the relation 4.3, then we get:

**Lemma 4.1.16** *A simple 2QFA $\mathcal{A}$ satisfies the well-formedness condition if and only if*

$$\sum_{q'} \langle q'|V_\sigma|q_1\rangle^* \langle q'|V_\sigma|q_2\rangle = \begin{cases} 1, & \text{if } q_1 = q_2; \\ 0, & \text{otherwise,} \end{cases}$$

*for each $\sigma \in \Gamma$, which holds if and only if every operator $V_\sigma$ is unitary.*

**Exercise 4.1.17** *Prove in detail Lemma 4.1.16.*

**Example 4.1.18** *Let $\mathcal{A} = \langle \Sigma, Q, q_0, Q_a, Q_r, \delta\rangle$ be a simple 2QFA with $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{0,1\}$, $Q_a = \{q_3\}$, $Q_r = \{q_4\}$ and*

$$\begin{array}{lllll}
V_\#|q_0\rangle = |q_0\rangle, & V_0|q_0\rangle = |q_0\rangle, & V_1|q_0\rangle = |q_1\rangle, & V_\$|q_0\rangle = |q_1\rangle, & D(q_0) = +1, \\
V_\#|q_1\rangle = |q_2\rangle, & V_0|q_1\rangle = |q_2\rangle, & V_1|q_1\rangle = |q_0\rangle, & V_\$|q_1\rangle = |q_0\rangle, & D(q_1) = -1, \\
V_\#|q_2\rangle = |q_4\rangle, & V_0|q_2\rangle = |q_4\rangle, & V_1|q_2\rangle = |q_2\rangle, & V_\$|q_2\rangle = |q_3\rangle. & D(q_2) = +1, \\
V_\#|q_3\rangle = |q_3\rangle, & V_0|q_3\rangle = |q_3\rangle, & V_1|q_3\rangle = |q_3\rangle, & V_\$|q_3\rangle = |q_2\rangle, & D(q_3) = \phantom{+}0, \\
V_\#|q_4\rangle = |q_1\rangle, & V_0|q_4\rangle = |q_1\rangle, & V_1|q_4\rangle = |q_4\rangle, & V_\$|q_4\rangle = |q_4\rangle, & D(q_4) = \phantom{+}0.
\end{array}$$

*By inspection one sees that all $V_\sigma$ operators are unitary and therefore if $\delta$ is defined as in 4.3, then $\mathcal{A}$ is well-formed. Below, Example 4.1.19, we will see the way $\mathcal{A}$ works.*

### 4.1.5 2QFA versus 1FA

Both acceptance and rejection can be defined for QFA in a natural and straightforward way. We only need to form a natural quantum version of the observation used in the classical FA.

For each input $x$ with $n = |x|$ and a QFA $\mathcal{A} = \langle \Sigma, Q, q_0, Q_a, Q_r, \delta \rangle$ let $C_n^a = \{(q,k) \,|\, (q,k) \in C_n, q \in Q_a\}$, $C_n^r = \{(q,k) \,|\, (q,k) \in C_n, q \in Q_r\}$ and $C_n^- = C_n - C_n^a - C_n^r$. Let $E_a, E_r$ and $E_n$ be the subspaces of $l_2(C_n)$ spanned by $C_n^a$, $C_n^r$ and $C_n^-$, respectively.

In order to define language recognition by $\mathcal{A}$ we use the "computational observable" $\mathcal{O}$ that corresponds to the orthogonal decomposition $l_2(C_n) = E_a \oplus E_r \oplus E_n$. The outcome of any observation will be either "accept" or "reject" or "non-terminating".

For example, if the amplitudes of all accepting and rejecting configurations in a superposition are 0, then the result of an observation, according to the observable $\mathcal{O}$ will be "non-terminating" with probability 1 and the state "collapses" into itself (and it will therefore not be disturbed by the observation).

The language recognition by $\mathcal{A}$ is now defined as follows; For an $x \in \Sigma^*$ we consider as the input the string $w_x = \#x\$$, and we assume that the computation starts with $\mathcal{A}$ being in the state $|q_0, 0\rangle$, i.e. with the head on the leftmarker #. Each step of the computation consists of two parts. At first the operator $U^\delta_{w_x}$ is applied to the current state and then the resulting state is observed using the computational observable $\mathcal{O}$ defined above. The computation continues this way until the result of an observation is "accept" or "reject", and in such a case the computation halts.

The language $L(\mathcal{A})$ accepted by $\mathcal{A}$ is defined in a similar way as for 1QFA. This way we can talk about bounded-error acceptance and unbounded-error acceptance.

**Example 4.1.19** *We show that $L(\mathcal{A}) = 0^*1^*$ for the automaton from the Example 4.1.18. More exactly, each $x \in L$ is accepted with probability 1 and each $x \in \{0,1\}^* - L$ is rejected with probability 1.*

*Let us start with an illustration of computations on $\mathcal{A}$ for two different input strings. For the input $0^3 1^2$ we get the following sequence of states:*

$$|q_0,0\rangle \xrightarrow{\#} |q_0,1\rangle \xrightarrow{0} |q_0,2\rangle \xrightarrow{0} |q_0,3\rangle \xrightarrow{0} |q_0,4\rangle \xrightarrow{1} |q_1,3\rangle \xrightarrow{0} |q_2,4\rangle \xrightarrow{1} |q_2,5\rangle \xrightarrow{1} |q_2,6\rangle \xrightarrow{\$} |q_3,6\rangle,$$

*and for the input $0^2 10$:*

$$|q_0,0\rangle \xrightarrow{\#} |q_0,1\rangle \xrightarrow{0} |q_0,2\rangle \xrightarrow{0} |q_0,3\rangle \xrightarrow{1} |q_1,2\rangle \xrightarrow{0} |q_2,3\rangle \xrightarrow{1} |q_2,4\rangle \xrightarrow{0} |q_4,4\rangle.$$

*It is easy to verify that for an input $0^i 1^j$, $i,j \geq 0$, the automaton $\mathcal{A}$ enters the state $|q_3, i+j+1\rangle$ after $i+j+4$ steps and at that time it will be for the first time that $\mathcal{A}$ gets into a halting state. An input $1^i 0x$ with $i > 0$ is rejected after $i+3$ steps and the input of the form $0^i 1^j 0x$ with $i > 0$ and $j > 0$ after $i+j+4$ steps.*

**Exercise 4.1.20** *Design a simple 2QFA accepting the languages: (a) $\{0^*1^*0^*\}$; (b) $\{(00)^*(11)^*\}$.*

As the next step we show that the family of languages recognized by one-sided bounded-error simple 2QFA in linear time contains also non-regular languages, namely the language $L = \{0^i 1^i \,|\, i > 0\}$.

For each integer $n$ let $\mathcal{A}^{(n)} = \langle\{0,1\}, Q^{(n)}, q_0, Q_a^{(n)} Q_r^{(n)}, \delta^{(n)}\rangle$ be a simple 2QFA with

$$\begin{aligned} Q^{(n)} &= \{q_0, q_1, q_2, q_3\} \cup \{r_{j,k} \,|\, 1 \leq j \leq n, 0 \leq k \leq \max\{j, n-j+1\}\} \\ &\quad \cup \{s_j \,|\, 1 \leq j \leq n\}, \end{aligned}$$

$Q_a^{(n)} = \{s_n\}$, $Q_r^{(n)} = \{q_3\} \cup \{s_j, 1 \leq j < n\}$. The transition function $\delta^{(n)}$ is defined as in (4.3) where $V_\sigma^{(n)}, \sigma \in \{\#, 0, 1, \$\}$ and $D$ are given in Figure 4.2. An extension of $V_\sigma^{(n)}$ to other basis states in such a way that all $V_\sigma^{(n)}$ are unitary is straightforward.

| | |
|---|---|
| $V_\#\|q_0\rangle = \|q_0\rangle$, | $V_\$\|q_0\rangle = \|q_3\rangle$, |
| $V_\#\|q_1\rangle = \|q_3\rangle$, | $V_\$\|q_2\rangle = \frac{1}{\sqrt{n}}\sum_{j=1}^n \|r_{j,0}\rangle$, |
| $V_\#\|r_{j,0}\rangle = \frac{1}{\sqrt{n}}\sum_{l=1}^n e^{\frac{2\pi i}{n}jl}\|s_l\rangle, 1 \leq j \leq n$, | |
| $V_0\|q_0\rangle = \|q_0\rangle$, | $D(q_0) = \rightarrow$, |
| $V_0\|q_1\rangle = \|q_2\rangle$, | $D(q_1) = \leftarrow$, |
| $V_0\|q_2\rangle = \|q_3\rangle$, | $D(q_2) = \rightarrow$, |
| $V_0\|r_{j,0}\rangle = \|r_{j,j}\rangle, 1 \leq j \leq n$, | $D(q_3) = \downarrow$, |
| $V_0\|r_{j,k}\rangle = \|r_{j,k-1}\rangle, 1 \leq k \leq j, 1 \leq j \leq n$, | |
| $V_1\|q_0\rangle = \|q_1\rangle$, | $D(r_{j,0}) = \leftarrow, 1 \leq j \leq n$, |
| $V_1\|q_2\rangle = \|q_2\rangle$, | $D(r_{j,k}) = \downarrow, 1 \leq j \leq n, k \neq 0$, |
| $V_1\|r_{j,0}\rangle = \|r_{j,n-j+1}\rangle, 1 \leq j \leq n$, | $D(s_j) = \downarrow, 1 \leq j \leq n$, |
| $V_1\|r_{j,k}\rangle = \|r_{j,k-1}\rangle, 1 \leq k \leq j \leq n$. | |

Figure 4.2: Transition function of $\mathcal{A}^{(n)}$.

**Lemma 4.1.21** *Let $x \in \{0,1\}^*, n \in \mathbf{N}$. If $x \in \{0^i1^i \,|\, i \geq 1\}$, then the 2QFA $\mathcal{A}^{(n)}$ accepts $x$ with the probability 1; otherwise $\mathcal{A}^{(n)}$ rejects $x$ with the probability at least $1 - \frac{1}{n}$. In either case $\mathcal{A}^{(n)}$ halts after $\mathcal{O}(n|x|)$ steps.*

**Proof.** Figure 4.3 illustrates the basic trick of a 2QFA described in Figure 4.2, that accepts strings from the language $\{0^i1^i \,|\, i > 0\}$. Each computation of $\mathcal{A}^{(n)}$ consists of three phases. In the first phase, in which only states $q_0, q_1, q_2$ and $q_3$ are involved, any input word not of the form $0^i1^j$ is rejected, in a similar way as in Example 4.1.19. For words of the type $0^i1^j$ the phase ends in the state $|q_2\rangle$ with the head on the rightmost endmarker \$. As the first step of the second phase the operator $V_\$$ is applied and a superposition of $n$ states is formed. This way computation branches into $n$ parallel paths starting in the states $|r_{1,0}\rangle, \ldots, |r_{n,0}\rangle$, each with the amplitude $\frac{1}{\sqrt{n}}$.

In the $j$th of the paths, starting in the state $|r_{j,0}\rangle$ the head moves, deterministically, to the left endmarker according to the following rules. Each time the head is on a new cell and reads 0 (1) it remains stationary for $j$ $(n-j+1)$ steps and then moves one cell left. Therefore, for an input of the form $0^u1^v$ the $j$th head requires exactly $(j+1)u + (n-j+2)v + 1$ steps to reach the left endmarker. If $j \neq j'$, then

$$(j+1)u + (n-j+2)v + 1 = (j'+1)u + (n-j'+2)v + 1 \text{ if and only if } u = v.$$

This implies that any two heads of all $n$ different computational paths reach the left endmarker at the same time if and only if $u = v$.[3]

[3]This means that in the second phase the automaton is always in a superposition of $n$ basis states (configurations) each corresponding to a different branch, and therefore one can see the computation as being performed by $n$ different heads which can be on different cells.

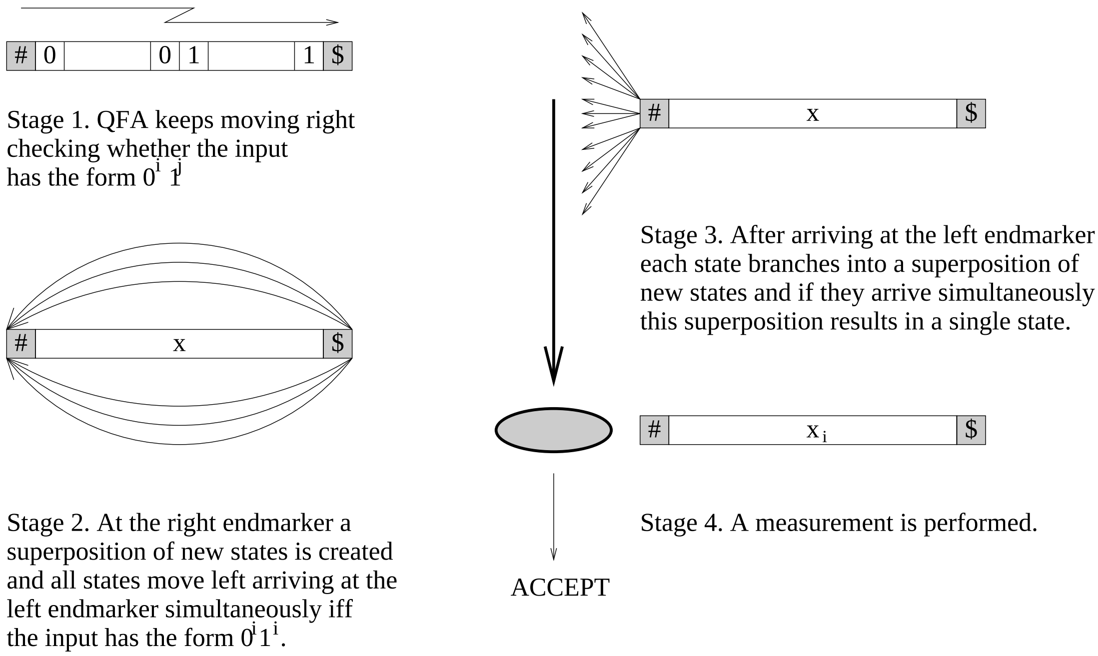


Figure 4.3: QFA recognizing the language $\{0^i 1^i \mid i \geq 1\}$ - 60%

In the third phase, consisting of only one step, each computation path splits again, this time the resulting superposition is obtained by an application of the QFT.

In the case $u = v$ all these splittings occur simultaneously and the resulting superposition has the form

$$\frac{1}{n}\sum_{j=1}^{n}\sum_{l=1}^{n} e^{\frac{2\pi i}{n}jl}|s_j\rangle, \tag{4.4}$$

and equals exactly to $|s_n\rangle$ on the basis of the same reasoning as in Sections 3.1 and 3.2. In the last phase, in addition, an observation is performed using the observable $\mathcal{O}$. In the case $u = v$ the result of such an observation is "accept" with probability 1.

Finally, in the case $u \neq v$, no two different computational paths come to the left endmarker at the same time and therefore after Phase 3, when the first observation is made, we get "accept" with probability only $\frac{1}{n}$ and "reject" with probability $1 - \frac{1}{n}$. ▯

The above method can be used to show that some non-context-free languages can also be accepted in linear time by bounded error 2QFA.

**Exercise 4.1.22** *Which of the following languages can be accepted by bounded-error linear time QFA: (a)* $\{a^n b^n a^n \mid n \geq 1\}$*; (b)* $\{a^n b^{2n} \mid n \geq 1\}$*; (c)* $\{a^i b^j c^j \mid i, j \geq 1\} \cup \{a^i b^i c^j \mid i, j \geq 1\}$*?*

Simple 2QFA can be shown to accept all regular languages. Actually a stronger statement has been shown. Namely, that any deterministic (one-way) FA can be simulated by a two-way reversible finite automaton (which can be defined as a 2QFA, amplitudes of which are only 0 or 1). As a corollary we have

**Theorem 4.1.23** *Every regular language is accepted by a 2QFA.*

This theorem has been shown by Kondacs and Watrous (1997) directly and it also follows from a more general result due to Lange at al. (1997).

**Exercise 4.1.24** *Show that any 1FA can be simulated by a reversible two-way finite automaton.*

**Exercise 4.1.25** *Show that the language of words over the alphabet* $\{0,1\}$ *that contain the same number of* $0$*'s and* $1$*'s can be accepted by a 2QFA*

**Exercise 4.1.26** *Let us call a 2QFA* $\mathcal{A}$ **one-directional** *if it never moves left. Show that one-directional 2QFA can accept non-regular languages.*

**Open problem 4.1.27**

1. *Is it true that one-directional 2QFA accept all regular languages?*
2. *Is it true that unidirectional 2QFA are as powerful as 2QFA?*
3. *Can we increase the power of 2QFA by considering some other observables?*
4. *Can exponential time, bounded-error, 2QFA recognize languages not recognizable by linear time, bounded-error 2QFA?*
5. *How powerful are 2QFA compared to multihead FA?*
6. *Find an elegant characterization of the family of languages acceptable by 2QFA?*

**Remark 4.1.28** One-directional 2QFA seem to be very powerful. Amano and Iwama (1999), who denote them as 1.5QFA, showed that one of the basic decision problem, the emptiness problem, which is decidable even for pushdown automata, is undecidable for 1.5QFA. Perhaps it would have been more proper to call them 1QFA and those denoted that way so far as "real-time QFA",

There are several variants of the models of quantum finite automata discussed here and they need to be explored in order to get a better insight into the power of quantum computation on the "finite state" level. Some of these models are presented in the following exercises (Ambainis and Freivalds (1998)).

**Exercise 4.1.29** *Let us consider special two-way QFA in which the head keeps making left-to-right and right-to-left passes between endmarkers. (a) Show that such a quantum automaton can recognize the language* $\{0^i1^i \mid i \geq 0\}$ *in the following sense. If* $x \notin L$ *the automaton stops with probability* $1$ *after* $\mathcal{O}(|x|)$ *scans of the tape; if* $x \in L$*, then it never stops; (b) explore more potentials of such quantum automata.*

**Exercise 4.1.30** *Show that for any $\varepsilon > 0$ there is a 2-way probabilistic finite automaton $\mathcal{A}_1$ and an 1QFA $\mathcal{A}_2$ such that with probability at least $1-\varepsilon$: (1) $\mathcal{A}_1$ stops in time quadratic in the length of the input; (2) $\mathcal{A}_2$ accepts the output of $\mathcal{A}_1$ if and only if the input $x$ of $\mathcal{A}_1$ is in $\{0^i1^i \mid i > 0\}$.*

The result of Exercise 4.1.30 is of interest for the following reason: neither 1QFA nor 2PFA can recognize non-regular languages (see Freivalds, 1981, and Kaneps and Freivalds, 1991), but their composition can.

# 4.2 Quantum Turing Machines

Quantum versions of Turing machines play a similar fundamental role in the theory of quantum computing as classical Turing machines play in the classical complexity theory.

## 4.2.1 One-tape quantum Turing machines

One-tape quantum Turing machines are the main type of quantum Turing machines to be investigated in the following.

### Basic model and its evolution

**Definition 4.2.1** *A (one-tape)* **quantum Turing machine** *$\mathcal{M} = \langle \Sigma, Q, q_0, q_f, \delta\rangle$, QTM in short, is defined analogously to a PTM, with the initial state $q_0$ and the final state $q_f$, but the transition amplitude mapping*

$$\delta : Q \times \Sigma \times \Sigma \times Q \times \{\leftarrow, \downarrow, \rightarrow\} \longrightarrow \mathbf{C}_{[0,1]}$$

*is required to be such that quantum evolution of $\mathcal{M}$, to be defined later, is unitary. If the unitarity condition is not required to be valid we talk about a quasi-QTM (in short qQTM). Moreover, when neither the initial nor the final state is of importance a QTM is specified in the form $\langle \Sigma, Q, \delta\rangle$.*[4]

**Exercise 4.2.2** *(A random walk QTM—Watrous, 1998) Let us consider a QTM $\mathcal{M}$ with two states alphabet $\{0,1\}$, $q_0 = 0$, and any tape alphabet $\Sigma$, such that at each step $\mathcal{M}$ applies Hadamard transform to its state and moves its head left or right, depending on whether new state is 0 or 1. ($\mathcal{M}$ therefore never changes the content of the tape.) Describe the superposition of $\mathcal{M}$ after $i$ steps, $i = 1, 2, \ldots, 10$. What can you say about the behaviour of $\mathcal{M}$ in general?*

[4] Quite a different approach to QTM is pursued by Benioff, see page 259 and Benioff (1998) for the recent paper. It is the so-called "physical QTM", and the main difference is in problems on which investigation concentrates. Benioff (1998) is concerned with the process defined by a step operator that is used to construct Hamiltonian according to Feynman's ideas—see page 259. It is interesting to compare this "physical" approach to QTM with the "automata theoretic" approach as presented in this section. In spite of the fact that Benioff's approach may be seen, from an implementation point of view, as better reflecting quantum physics laws; so far the automata-theoretic approach seems to be far more insightful, stimulating, and productive.

The concept of configuration is the basic one for the description of the quantum evolution of QTM. A configuration of $\mathcal{M}$ is determined by the content $\tau$ of the tape, $\tau \in \Sigma^{\mathbf{Z}}$, by an $i \in \mathbf{Z}$ which specifies the position of the head, and by a $q \in Q$, the current state of the tape. Let $C_{\mathcal{M}}$ denote the set of all configurations of $\mathcal{M}$. Computation (evolution) of $\mathcal{M}$ is performed in the inner-product space $H_{\mathcal{M}} = l_2(C_{\mathcal{M}})$ with the basis $\{|c\rangle \mid c \in C_{\mathcal{M}}\}$.

The transition function $\delta$ uniquely determines a mapping $a : C_{\mathcal{M}} \times C_{\mathcal{M}} \rightarrow \mathbf{C}$ such that for $c_1, c_2 \in C_{\mathcal{M}}$, $a(c_1, c_2)$ is the amplitude of the transition of $\mathcal{M}$ from the basis state $|c_1\rangle$ to $|c_2\rangle$. The time evolution mapping $U_{\mathcal{M}} : H_{\mathcal{M}} \rightarrow H_{\mathcal{M}}$ is then defined as follows.

1. If $|c\rangle$ is a basis state, then
$$U_{\mathcal{M}}|c\rangle = \sum_{c' \in C_{\mathcal{M}}} a(c, c')|c'\rangle.$$

2. If $|\phi\rangle = \sum_{c \in C_{\mathcal{M}}} \alpha_c |c\rangle$ is a superposition, then
$$U_{\mathcal{M}}|\phi\rangle = \sum_{c \in C_{\mathcal{M}}} \alpha_c U_{\mathcal{M}}|c\rangle.$$

**Remark 4.2.3** 1. Sometimes it is convenient to see the basis states of a QTM as being tensor products of the form $|q\rangle|\tau\rangle|i\rangle$, where $q \in Q$, $\tau \in \Sigma^{\mathbf{Z}}$, $i \in \mathbf{Z}$, with $i$ representing the position of the head.

2. Observe that the subspace of $H_{\mathcal{M}} = l_2(C_{\mathcal{M}})$ consisting of finite sums of configurations is not a Hilbert space but a dense subspace of $l_2(C_{\mathcal{M}})$.

### Measurements

In addition to a standard measurement with respect to the basis $\{|c\rangle \mid c \in C_{\mathcal{M}}\}$, which, when applied to a state $|\phi\rangle = \sum_{c \in C_{\mathcal{M}}} \alpha_c |c\rangle$, provides a configuration $c$ with probability $|\alpha_c|^2$, measurements of certain cells of the tape are also of importance. The concept of an equivalence of configurations is needed to introduce the corresponding observables.

If $c$ is a configuration, then let $c(i)$ be the $i$th symbol of its tape. The equivalence relation on configurations, denoted $\sim_I$ and associated to a set $I = \{-N, \ldots, N\}$, $N \in \mathbf{N}$, is defined by
$$c_i \sim_I c_j \Longleftrightarrow c_i(k) = c_j(k) \quad \textit{whenever} \quad k \in I.$$

(In other words, two configurations are equivalent over $I$ if they have the same symbols in the positions from $I$.) Let us now denote by $[c]$ the equivalence class of configurations containing the configuration $c$. In the case of the state $|\psi\rangle = \sum_{c \in C_{\mathcal{M}}} \alpha_c |c\rangle$, the probability of such equivalence class is defined by
$$p([c]) = \sum_{c_i \in [c]} |\alpha_{c_i}|^2.$$

**Definition 4.2.4** *The standard observable of the cells $I$ of the tape yields the equivalence class $[c]$ with probability $p([c])$ and the post-observation superposition is*
$$\frac{1}{\sqrt{\sum_{c_j \in [c]} |\alpha_{c_j}|^2}} \sum_{c_j \in [c]} \alpha_{c_j} c_j,$$
*provided the class $[c]$ was observed.*

### Well-formedness conditions

As shown in Theorem 4.2.6, unitarity of a qQTM $\mathcal{M}$ is ensured if $\mathcal{M}$ satisfies the following so-called strong well-formedness conditions.

**Definition 4.2.5** *A qQTM $\mathcal{M} = \langle \Sigma, Q, q_0, q_f, \delta \rangle$ with the transition mapping*

$$\delta : Q \times \Sigma \times \Sigma \times Q \times \{\leftarrow, \downarrow, \rightarrow\} \longrightarrow \mathbf{C}$$

*is said to be strongly well-formed if the following conditions are satisfied.*

1. **Local probability condition.** *For any $(q_1, \sigma_1) \in Q \times \Sigma$;*

$$\sum_{(\sigma,q,d) \in \Sigma \times Q \times \{\leftarrow,\downarrow,\rightarrow\}} |\delta(q_1, \sigma_1, \sigma, q, d)|^2 = 1.$$

2. **Separability conditon I.** *For any two different pairs $(q_1, \sigma_1), (q_2, \sigma_2)$ from the set $Q \times \Sigma$:*

$$\sum_{(q,\sigma,d) \in Q \times \Sigma \times \{\leftarrow,\downarrow,\rightarrow\}} \delta^*(q_1, \sigma_1, \sigma, q, d)\delta(q_2, \sigma_2, \sigma, q, d) = 0.$$

3. **Separability condition II.** *For any $(q, \sigma, d), (q', \sigma', d')$ from the set $Q \times \Sigma \times \{\leftarrow, \downarrow, \rightarrow\}$ such that $(q, \sigma, d) \neq (q', \sigma', d')$:*

$$\sum_{(q_1,\sigma_1) \in Q \times \Sigma} \delta^*(q_1, \sigma_1, \sigma, q, d)\delta(q_1, \sigma_1, \sigma', q', d') = 0.$$

4. **Separability condition III.** *For any $(q_1, \sigma_1, \sigma_1'), (q_2, \sigma_2, \sigma_2') \in Q \times \Sigma \times \Sigma$ and $d_1 \neq d_2 \in \{\leftarrow, \downarrow, \rightarrow\}$:*

$$\sum_{q \in Q} \delta^*(q_1, \sigma_1, \sigma_1', q, d_1)\delta(q_2, \sigma_2, \sigma_2', q, d_2) = 0.$$

A **computation** of a QTM $\mathcal{M}$ is a sequence of superpositions $c_0, c_1, c_2, \ldots$, where $c_0$ is the initial configuration and for $i \geq 1$ $c_i = U_{\mathcal{M}}(c_{i-1})$ or $c_i$ is obtained from $c_{i-1}$ via an observation procedure.

**Theorem 4.2.6** *(Hirvensalo[5], 1997) If a qQTM $\mathcal{M}$ satisfies the above strongly well-formedness conditions, then its evolution is unitary.*

**Proof.** Let us assume a fixed enumeration of all configurations. For any configuration $c_i$ then

$$U_{\mathcal{M}}(c_i) = \sum_{l=1}^{\infty} \alpha_{li} c_l, \tag{4.5}$$

where $\alpha_{li}$ is the amplitude of reaching $c_l$ from $c_i$. Observe that for each $i$ the sum (4.5) is actually finite, because there are only finitely many configurations reachable from any $c_i$ in one step. $U_{\mathcal{M}}$ can now be seen as a finite or infinite matrix with $U_{\mathcal{M}}(l, i) = \alpha_{li}$. Each column of the matrix has therefore only finitely many nonzero elements.

[5]The proof makes use of the ideas of Bernstein and Vazirani (1993).

If now $U^*_{\mathcal{M}}$ is the adjoint matrix to $U_{\mathcal{M}}$, then

$$U^*_{\mathcal{M}}(c_i) = \sum_{l=1}^{\infty} \alpha^*_{il} c_l.$$

We first show that $U^*_{\mathcal{M}} U_{\mathcal{M}} = I$. This will imply that the mapping $U_{\mathcal{M}}$ is one-to-one. Indeed,

$$U^*_{\mathcal{M}}(U_{\mathcal{M}}(c_i)) = \sum_{l=1}^{\infty} \left( \sum_{k=1}^{\infty} \alpha_{ki} \alpha^*_{kl} \right) c_l.$$

The right-hand side is exactly $c_i$ if

$$\sum_{k=1}^{\infty} \alpha_{ki} \alpha^*_{kl} = 0, \tag{4.6}$$

whenever $l \neq i$ and

$$\sum_{k=1}^{\infty} \alpha_{ki} \alpha^*_{ki} = \sum_{k=1}^{\infty} |\alpha_{ki}|^2 = 1.$$

The last equalities follow from the condition (1) in the definition of well-formedness.

Let us now consider condition (4.6), which actually requires orthogonality of the sequence of amplitudes $\{\alpha_k \,|\, c_k$ is the successor configuration of $c_i, k \in \mathbf{N}\}$ and $\{\beta_k \,|\, c_k$ is the successor configuration of $c_l, k \in \mathbf{N}\}$.

Observe that from different configurations $c_i$ and $c_l$ it is possible to obtain $c_k$ in one step only if the head positions in $c_i$ and $c_l$ differ at most by two positions.

Now if the head is in the same position in $c_i$ and $c_l$, the orthogonality follows from condition 3; if the heads are in different positions, the orthogonality follows from condition 4 of the definition of well-formedness.

To finish the proof of the theorem it is now sufficient to show that the mapping $U_{\mathcal{M}}$ is surjective. This will imply that $U^*_{\mathcal{M}} = U^{-1}_{\mathcal{M}}$ and the theorem is proved.

A configuration $c_i$ will be called reachable (by $\mathcal{M}$) if $c_i$ occurs with a non-zero coefficient in some superpositions $U_{\mathcal{M}}(c_k) = \sum_{j=1}^{\infty} \alpha_j c_j$—which happens if the $i$th row of $U_{\mathcal{M}}$ is not empty.

We first show that each configuration is reachable. This will imply that $U_{\mathcal{M}}$ has no empty row.

Let us assume, on the contrary, that there is a configuration $c'$ not reachable from any other configuration. Let $\sigma_1 q \sigma_2 \sigma_3$ be a "local subconfiguration" of $c'$ containing the state $q$, symbols in the cell with the head on $\sigma_2$, and in two neighbouring cells $\sigma_1, \sigma_3$. If $c'$ is not reachable, then the same has to be true for all configurations having $\sigma_1 q \sigma_2 \sigma_3$ as the local subconfiguration. All these configurations will be said to be locally like $c'$. Let us now take an $n \geq 4$ and consider all configurations $c_{i_1}, \ldots, c_{i_K}$ such that in these configurations all cells outside cells $0, 1, \ldots, n$ have blanks and the state symbol is not in the $n$th cell. For the subspace $V \subset C_{\mathcal{M}}$ generated by configurations $c_{i_1}, \ldots, c_{i_K}$, we have $dim(U_{\mathcal{M}}(V)) = K = n|Q|\,|\Sigma|^n$ because $U_{\mathcal{M}}$ is injective.

In one step from any $c_{i_j}$ one can either reach another $c_{i_k}$, or exit the chosen $n+1$ cells, and there are at most $2|Q||\Sigma|^n$ configurations to exit into. In total this gives $K + 2|Q|\,|\Sigma|^n$ possible successor configurations. However, at least $(n-2)|\Sigma|^{n-3}$ of them look locally like $c'$, and therefore they cannot be reached from any configuration. Consequently we get

$$U_{\mathcal{M}}(c_{i_k}) = \sum_{j_l \in J} \alpha_l c_{j_l}$$

with the index set $J$ of configurations of the cardinality at most $K+2|Q|\,|\Sigma|^n-(n-2)|\Sigma|^{n-3}$. Therefore,

$$K = dim(U_{\mathcal{M}}(V)) \leq K + 2|Q|\,|\Sigma|^n - (n-2)|\Sigma|^{n-3},$$

which yields $n \leq 2|Q|\,|\Sigma|^3 + 2$ and contradicts the fact that $n$ could be chosen arbitrarily large.

Without loss of generality let us now assume that the configuration $c_1$ is not in the range of $U_{\mathcal{M}}$. As shown above, the first row of $U_{\mathcal{M}}$ is not empty. Moreover, it follows from the condition 2 of the definition of well-formedness of QTM that any two rows of $U_{\mathcal{M}}$ are orthogonal.

Let us now choose $N$ such that $\alpha_{ij} = 0$ for $j > N$. From the orthogonality of rows of $U_{\mathcal{M}}$ it then follows that

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2N} \\ \vdots & \vdots & \ldots & \vdots \end{pmatrix} \begin{pmatrix} \alpha^*_{11} \\ \alpha^*_{12} \\ \vdots \\ \alpha^*_{1N} \end{pmatrix} = \begin{pmatrix} A \\ 0 \\ 0 \\ \vdots \end{pmatrix},$$

where $A = |\alpha_{11}|^2 + \ldots + |\alpha_{1N}|^2 > 0$. Hence

$$U_{\mathcal{M}}(\alpha^*_{11}c_1 + \ldots + \alpha^*_{1N}c_N) = Ac_1$$

and by dividing both sides by $A$ we get that $c_1$ is in the range of $U_{\mathcal{M}}$. ▯

**Exercise 4.2.7** *Show that any reversible TM is QTM.*

**Exercise 4.2.8** *Let $\delta : Q \times \Sigma \rightarrow \Sigma \times Q \times \{\leftarrow, \downarrow, \rightarrow\}$ be a partial mapping satisfying the following conditions.*

1. *If $\delta(q_1, \sigma_1) = (\sigma'_1, q, d_1)$, $\delta(q_2, \sigma_2) = (\sigma'_2, q, d_2)$, then $d_1 = d_2$;*
2. *$\delta$ is one-to-one if the direction components ($\{\leftarrow, \downarrow, \rightarrow\}$) are ignored.*

*Show that $\delta$ can be completed to be a total mapping in such a way that $\mathcal{M} = \langle Q, \Sigma, \delta\rangle$ is an RTM.*

Hirvensalo was the first to come with some conditions for a general qQTM to have unitary evolution. However, conditions presented in Definition 4.2.5 are only sufficient conditions. As shown by Ozawa, they are not necessary. The first set of sufficient and necessary well-formedness conditions is due to Ozawa (1998). Instead of separability conditions II and III, he considers the following two conditions:

3. **Separability condition II':** For any $(q_1, \sigma_1, \sigma'_1)$, $(q_2, \sigma_2, \sigma'_2) \in Q \times \Sigma \times \Sigma$

$$\sum_{p \in Q} \delta^*(q_1, \sigma_1, \sigma'_1, p, \rightarrow)\delta(q_2, \sigma_2, \sigma'_2, p, \leftarrow) = 0.$$

4. **Separability condition III':** For any $(q_1, \sigma_1, \sigma'_1)$, $(q_2, \sigma_2, \sigma'_2) \in Q \times \Sigma \times \Sigma$

$$\sum_{p \in Q, d \in \{\downarrow, \rightarrow\}} \delta^*(q_1, \sigma_1, \sigma'_1, p, d-1)\delta(q_2, \sigma_2, \sigma'_2, p, d) = 0,$$

where $d-1 = \leftarrow$ if $d = \downarrow$ and $d-1 = \downarrow$ if $d = \rightarrow$.

The proof that these conditions are really sufficient and necessary has been given in Ozawa and Nishimura (1998) and it is too technical and lengthy to be presented here. The following QTM, due to Ozawa, 1998, is an example of a QTM that does not satisfies conditions of Definition 4.2.5:

**Exercise 4.2.9** *Verify that QTM with the set of states* $Q = \{0, 1\}$, $\Sigma = \{0\}$ *and the transition function defined for simplicity as the mapping* $\delta : Q \times Q \times \{\leftarrow, \downarrow, \rightarrow\} \rightarrow \mathbf{C}$ *defined for* $a = \delta(p, q, c)$ *as follows*

| $q$ | $p$ | $d$ | $a$ | $q$ | $p$ | $d$ | $a$ | $q$ | $p$ | $d$ | $a$ | $q$ | $p$ | $d$ | $a$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | $\leftarrow$ | 0 | 0 | 1 | $\leftarrow$ | $\frac{1}{2}$ | 1 | 0 | $\leftarrow$ | 0 | 1 | 1 | $\leftarrow$ | $\frac{1}{2}$ |
| 0 | 0 | $\downarrow$ | $\frac{1}{2}$ | 0 | 1 | $\downarrow$ | $\frac{1}{2}$ | 1 | 0 | $\downarrow$ | $\frac{1}{2}$ | 1 | 1 | $\downarrow$ | $-\frac{1}{2}$ |
| 0 | 0 | $\rightarrow$ | $-\frac{1}{2}$ | 0 | 1 | $\rightarrow$ | 0 | 1 | 0 | $\rightarrow$ | $\frac{1}{2}$ | 1 | 1 | $\rightarrow$ | 0 |

*satisfies Ozawa's but not Hirvensalo's conditions,*

### Input–output conventions

The initial configuration of a QTM $\mathcal{M} = \langle \Sigma, Q, q_0, q_f, \delta \rangle$ has the form $q_0 x$, $x \in (\Sigma - \{\lambda\})^*$, where $\lambda$ stands for the blank symbol, and $x$ is written on cells numbered $0, 1, 2, \ldots, |x| - 1$, with all other cells filled with the blank and the head is on the cell number 0. $\mathcal{M}$ halts on the input $x$ when it finally enters the final state $q_f$. The number of steps needed to reach $q_f$ is the computation time of $\mathcal{M}$ on $x$. After $\mathcal{M}$ halts the standard measurement is performed and its output is the string on the tape of the resulting configuration consisting of the tape contents between the leftmost and rightmost non-blank symbol. Each output is therefore produced with certain probability. For each input $\mathcal{M}$ produces a sample from a probability distribution on its outputs.

**Remark 4.2.10** It has been shown by Bernstein (1997) that it is sufficient to consider a single final state and a single measurement, after halting, if time efficiency of quantum Turing machines is considered. On the other hand, when space efficiency of QTM is considered, then, as in the case of QFA, a measurement after each step seems to be more appropriate.

Two QTM will be considered as equivalent if for each input their output probability distributions are close to each other; or more exactly, if their total variation distance is small.[6]

**Exercise 4.2.11** *Let* $|\psi\rangle$, $|\phi\rangle$ *be states of an inner product space* $H$ *such that* $||\phi|| = ||\psi|| = 1$, $||\phi - \psi|| \leq \varepsilon$. *Show that the total variation distance between probability distributions corresponding to measurements of* $|\phi\rangle$ *and* $|\psi\rangle$, *with respect to the standard basis, is at most* $4\varepsilon$.

## 4.2.2 Variations on the basic model

Quantum Turing machines are powerful, but far from easy to deal with, models of computation. The requirement of unitarity imposed on their evolutions is very severe and makes

[6]Total variation distance of two probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ on the domain $I$ is $\frac{1}{2}\sum_{i \in I} |\mathcal{D}_1(i) - \mathcal{D}_2(i)|$.

it far from trivial to perform even small modifications of the transition function of a QTM. For example, even a simple classical design step, to add a transition from the final state to the initial state, may not work—unitarity can disappear—and the new qQTM is not really quantum.

There are several other models of QTM that are easier to deal with because of the restrictive nature of the movement of the head, and their computational power, or even efficiency, is not worse than that of the most general model presented in Section 4.2.1.

### DBV–model of quantum Turing machine

The basic restriction, due to Deutsch (1985), and worked out by Bernstein and Vazirani (1993, 1997), is to consider only such QTM the head of which in each step must move either right or left. If the transition function $\delta$ of a qQTM $\mathcal{M}$ satisfies this condition, then the evolution of $\mathcal{M}$ is unitary if the first two conditions of Definition 4.2.5 and the first condition of Remark 4.2.1 are satisfied, as shown in detail by Bernstein and Vazirani (1997).

The DBV-model of quantum Turing machine will be the basic one to consider in the rest of this section and in Chapter 5.

### Unidirectional quantum Turing machines

In various situations, for example when designing a universal QTM, it is very handy to work with so-called unidirectional QTM, in which each state can be entered only by a head moving in the direction uniquely specified by the state itself.

**Definition 4.2.12** *A QTM $\mathcal{M} = \langle \Sigma, Q, q_0, q_f, \delta \rangle$ is called* **unidirectional** *(uQTM) if $d_1 = d_2$ whenever $\delta(q_1, \sigma_1, \sigma_1', q, d_1) \neq 0 \neq \delta(q_2, \sigma_2, \sigma_2', q, d_2)$.*

It is not obvious whether a restriction to uQTM does not represent an essential reduction of the power or efficiency of QTM. In order to deal with this problem the concept of simulation of QTM is needed.

**Definition 4.2.13** *A QTM $\mathcal{M}'$ simulates a QTM $\mathcal{M}$, with slowdown $f : \mathbf{N} \to \mathbf{N}$ and accuracy $\varepsilon$, if the following condition holds. If $\mathcal{D}$ is a distribution which is produced by $\mathcal{M}$ on input $x$ after $t$ steps and $\mathcal{D}'$ is the distribution produced by $\mathcal{M}'$ on the same input after $f(t)$ steps, then $|\mathcal{D} - \mathcal{D}'| \leq \varepsilon$.*

Concerning the power and efficiency of uQTM the following basic result has been shown by Bernstein and Vazirani (1997).

**Theorem 4.2.14** *Any QTM can be simulated by a uQTM with slowdown factor at most* 5.

**Exercise 4.2.15** *Show that if $\mathcal{M}$ is an RTM, then there is a DBV-RTM $\mathcal{M}'$ that simulates $\mathcal{M}$ with slowdown at most* 2.

**Exercise 4.2.16** *Show that if $\mathcal{M}$ is a QTM, then there is a DBV-QTM $\mathcal{M}'$ that simulates $\mathcal{M}$ with constant slowdown.*

**Multitape Turing machines**

A natural generalization of one-tape QTM are, as in the classical case, multitape QTM. They are defined in a similar way to one-tape QTM. Concerning time efficiency, it is far from easy to see what kind of increase in the computational power such a model brings. However, as Bernstein (1997) shows, such an increase is not essential; namely, each multitape QTM can be simulated by a one-tape QTM with only polynomial time slowdown and the degree of the polynomial does not depend on the number of tapes. Multitape quantum Turing machines are a basic model to study the space efficiency of quantum Turing machines. They are discussed in more detail in Section 5.2.

The first attempt to formulate necessary and sufficient local well-formedness conditions for multitape qQTM to be QTM, similar to those for one-tape QTM, page 166, are due to Ozawa and Nishimura (1998).

### 4.2.3 Are quantum Turing machines analogue or discrete?

It is intuitively clear that in order to be able to investigate the computational power of QTM and to consider the idea of a universal QTM some restrictions have to be made on the amplitudes of the transitions of QTM. Otherwise we could encode sometimes into amplitudes what is to be computed. Moreover, the existence of a universal QTM would be out of the question because for that we need a clear and finite way of specifying other Turing machines.

The restriction we will make is that to each amplitude $\alpha$ there is a deterministic algorithm that can compute both the real and imaginary parts of $\alpha$ to within $2^{-n}$ in time polynomial in $n$ (see Adleman et al. 1997).

Three problems concerning amplitudes will be considered in this section. (1) Do we really need such a wide range of amplitudes to be able to fully utilize the power of QTM? (2) How precisely amplitudes need to be specified in order to get outcomes with a desirable precision? (In reality to each amplitude a physical quantity corresponds, for example the angle of a polarizer, and therefore practically there are strong limitations on precision to be achieved.) (3) Are QTM analogue or discrete models of computation?

**Types of amplitudes needed**

The first result presented below (due to Bernstein and Vazirani, 1993), says that as far as the computational power of QTM is concerned, it is sufficient to consider only QTM with real amplitudes.

**Theorem 4.2.17** *Any QTM $\mathcal{M}$ can be simulated, with constant time slowdown, by a QTM $\mathcal{M}'$ all transitions of which have real amplitudes.*

**Proof sketch**. The basic idea is simple: in any superposition of configurations replace each basis configuration $\alpha|q\rangle|\tau\rangle|j\rangle$ with a superposition $\alpha_i|q_i\rangle|\tau\rangle|j\rangle + \alpha_r|q_r\rangle|\tau\rangle|j\rangle$, where $\alpha_i$ and $\alpha_r$ are the imaginary and real parts of the amplitude $\alpha$ and $q_i$ and $q_r$ are new states. In correspondence with this the transition function $\delta'$ of $\mathcal{M}'$ is defined using the transition function $\delta$ of $\mathcal{M}$ as follows:

$$\begin{array}{rcl}
\delta'(\sigma, p_r, \sigma', q_r, d) & = & Re(\delta(\sigma, p, \sigma', q, d)), \\
\delta'(\sigma, p_r, \sigma', q_i, d) & = & Im(\delta(\sigma, p, \sigma', q, d)), \\
\delta'(\sigma, p_i, \sigma', q_r, d) & = & -Im(\delta(\sigma, p, \sigma', q, d)), \\
\delta'(\sigma, p_i, \sigma', q_i, d) & = & Re(\delta(\sigma, p, \sigma', q, d)).
\end{array}$$

$\Box$

**Exercise 4.2.18** *Show that if a transition function of a QTM is modified as shown in Theorem 4.2.17, then the evolution remains unitary.*

**Exercise 4.2.19** *Show that to any quantum circuit over $k$ qubits we can design an equivalent (in a reasonable sense) quantum circuit over $k+1$ qubits all gates of which are represented by unitary matrices over reals.*

Observe that the QTM $\mathcal{M}'$ constructed in the proof of Theorem 4.2.17, which uses only real amplitudes and simulates $\mathcal{M}$, works in the Hilbert space of twice as large dimension as $\mathcal{M}$.

Theorem 4.2.17 was strengthened by Adleman et al. (1997). They showed that with respect to the computational power it is sufficient to consider only rational amplitudes, and that the set of amplitudes $\{0, \pm\frac{3}{5}, \pm\frac{4}{5}, 1\}$ is sufficient to construct a universal QTM.

**Remark 4.2.20** It can be shown (see Watrous, 1995), quite surprisingly, that any QTM $\mathcal{M}$ can be simulated with a constant slowdown by a QTM $\mathcal{M}'$ with the "deterministic head position". In other words, if $\mathcal{M}'$ is observed during a computation, then the probability that the head will be observed at any given tape cell will be either 1 or 0. This implies that the position of the head of $\mathcal{M}'$ can be observed at every time step without affecting its computation.

### Precision required in QTM

Two issues will now be considered (see Bernstein and Vazirani, 1997). How imprecisions, either in the initial state, or during evolution steps, influence the overall imprecision of the computation; and how precisely amplitudes need to be given if the overall precision of computation is not to be compromised. To both questions there are very encouraging answers.

The first result shows that imprecisions during computations only add (and they do not increase exponentially as in the case of classical computations).

**Theorem 4.2.21** *Let $U_{\mathcal{M}}$ be the evolution operator of a QTM. If $|\phi_i\rangle$ and $|\phi_i'\rangle$, $i = 0, 1, \ldots, t$ are superpositions from $H_{\mathcal{M}}$, such that $||\phi_i - \phi_i'|| \leq \varepsilon$, $|\phi_i\rangle = U_{\mathcal{M}}|\phi_{i-1}'\rangle$, then $||\phi_t' - U_{\mathcal{M}}^t\phi_0|| \leq t\varepsilon$.*

**Proof.** Denote $|\psi_i\rangle = |\phi_i'\rangle - |\phi_i\rangle$. It holds.

$$\begin{aligned} |\phi_t'\rangle &= |\phi_t\rangle + |\psi_t\rangle = U_{\mathcal{M}}|\phi_{t-1}'\rangle + |\psi_t\rangle = \ldots = \\ &= U_{\mathcal{M}}^t|\phi_0\rangle + U_{\mathcal{M}}^t|\psi_0\rangle + U_{\mathcal{M}}^{t-1}|\psi_1\rangle + \ldots + U_{\mathcal{M}}|\psi_{t-1}\rangle + |\psi_t\rangle \end{aligned}$$

□

To formulate the second result we need the concept of "closeness" for QTM.

**Definition 4.2.22** *Two QTM $\mathcal{M}$ and $\mathcal{M}'$ are $\varepsilon$-close, $\varepsilon > 0$, if they have the same sets of states and symbols and if the difference between pairs of the corresponding amplitudes has magnitude at most $\varepsilon$.*

Two close QTM produce two close evolutions in the following sense.

**Theorem 4.2.23** *(Bernstein, Vazirani, 1998) If two QTM, $\mathcal{M}$ and $\mathcal{M}'$, with the set of states $Q$ and alphabet $\Sigma$ are $\varepsilon$-close, then the difference of their time evolutions has the norm bounded by $2|Q|\,|\Sigma|\varepsilon$. (The statement holds also in the case $\mathcal{M}'$ is a qQTM.)*

**Proof.** Let $\mathcal{M}$ and $\mathcal{M}'$ be QTM that are $\varepsilon$-close and let $\Sigma$ and $Q$ be their tape symbols and states. Both $H_{\mathcal{M}}$ and $H_{\mathcal{M}'}$ have the same basis $\{|c\rangle \mid c \in C_{\mathcal{M}} = C_{\mathcal{M}'}\}$. The difference in their evolution from a state $|\phi\rangle = \sum_{c\in C_{\mathcal{M}}} \alpha_c|c\rangle$ can be expressed as follows

$$U|\phi\rangle - U'|\phi\rangle = \sum_{c_j\in C_{\mathcal{M}}} \left( \sum_{c_i\in C_{\mathcal{M}'}} (\varepsilon_{i,j} - \varepsilon'_{i,j})\alpha_{c_i} \right) |c_j\rangle,$$

where $\varepsilon_{i,j}$ and $\varepsilon'_{i,j}$ are amplitudes of the transition from $c_i$ to $c_j$ in $\mathcal{M}$ and $\mathcal{M}'$. Since each configuration has at most $2|\Sigma|\,|Q|$ predecessor configurations and $|\sum_{i=1}^n a_i|^2 \leq n\sum_{i=1}^n a_i^2$ for any real $a_i$s and any $n$, we get

$$\begin{aligned}
||(U-U')\phi||^2 &= \sum_{c_j\in C_{\mathcal{M}}} \left| \sum_{c_i\in C_{\mathcal{M}}} (\varepsilon_{i,j} - \varepsilon'_{i,j})\alpha_{c_i} \right|^2 \\
&\leq \sum_{c_j\in C_{\mathcal{M}}} 2|\Sigma|\,|Q| \sum_{c_i\in C_{\mathcal{M}}} |(\varepsilon_{i,j} - \varepsilon_{i,j})\alpha_{c_i}|^2 \\
&\leq 2|\Sigma|\,|Q|\varepsilon^2 \sum_{c_i\in C_{\mathcal{M}}} \sum_{c_j\in C_{\mathcal{M}}} |\alpha_{c_i}|^2 \\
&\leq 4|\Sigma|^2|Q|^2\varepsilon^2.
\end{aligned}$$

□

**Corollary 4.2.24** *Let $\mathcal{M}$ be a QTM, and let $\mathcal{M}'$ be a QTM which is $\frac{\varepsilon}{24|\Sigma|\,|Q|\,t}$-close to $\mathcal{M}$, where $\varepsilon > 0$. Then $\mathcal{M}'$ simulates $\mathcal{M}$ for $t$ steps with accuracy $\varepsilon$. (The statement holds also in the case when $\mathcal{M}'$ is a qQTM.)*

**Proof.** Let $\varepsilon > 0$. If unitary operators $U_{\mathcal{M}}$ and $U_{\mathcal{M}'}$ are applied to the same state, then, by Theorem 4.2.23, the norm of the difference of the resulting states is bounded from above by $\delta = \frac{\varepsilon}{12t}$.

An application of $U_{\mathcal{M}'}$ can also be seen as an application of $U_{\mathcal{M}}$ and then an addition of perturbations of length at most $\delta$ times the length of current superposition. Therefore the length of the superpositions of $U_{\mathcal{M}'}$ after $t$ steps is increased by $(1+\delta)^t \leq e$, because $\delta \leq \frac{1}{t}$. By Theorem 4.2.21, the difference between superpositions of $\mathcal{M}$ and $\mathcal{M}'$ after $t$ steps is a superposition of the norm at most $3\delta t \leq \frac{\varepsilon}{4}$. Now the Corollary follows from Exercise 4.2.11. □

It follows from the last corollary that $\mathcal{O}(\lg t)$ bits of precision in transition amplitudes are sufficient to support $t$ steps of a QTM with the resulting precision $\varepsilon$. QTM can therefore be considered as discrete models of computation.

**Remark 4.2.25** As discussed later, especially in Chapter 5, a variety of fundamental results have been obtained concerning the computational power of QTM. In spite of that many basic problems are still far from being well understood. Two of such problems are termination and measurement issues for QTM.

### 4.2.4 Programming techniques for quantum Turing machines

Programming of QTM is significantly more difficult than of classical TM. On a general level difficulties are caused by the necessity to produce unitary, and consequently reversible, evolutions. This makes it nontrivial to implement even such simple classical programming primitives as composition, branching and looping. In addition, one has to manage the implementation of such inherently quantum primitives as an efficient implementation of arbitrary unitary transformations.

One general technique, often very useful, is first to design a deterministic TM (DTM) to do the task and then to use Bennett's construction to produce an equivalent reversible, and consequently also quantum, TM to perform the given task. The basic techniques presented bellow are due to Bernstein and Vazirani (1997).

#### Standard forms for QTM

One specific source of difficulties in programming of QTM is the need to time different computational paths correctly, because configurations interfere only if they are obtained after the same number of evolution steps. In addition, only identical configurations interfere—in other words only configurations with the same tape content, state and position of the head can interfere.

In order to overcome the timing problem and other problems that need to be solved in order to combine various QTM into more complex ones, it is convenient to work only with QTM that have standard input/output behaviour.

**Definition 4.2.26** *(1) A QTM $\mathcal{M}$ is called* **well-behaved** *if it halts for all inputs in a state all configurations of which are in the final state and have the head on the same cell. If this cell is always the starting cell, then $\mathcal{M}$ is called* **stationary**.

*(2) A QTM $\mathcal{M}$ is said to be in the* **normal form** *if each transition from the final state is into its initial state.*

**Remark 4.2.27** If a QTM is in the normal form, then there can be no other transition into the initial state than those from the final state. This allows one to redirect transitions from the final state and then add transitions to the initial state.

**Example 4.2.28** *We design a stationary, normal-form QTM which maps any state $|\psi\rangle$ of the $n$-qubit register into the state $H_n|\psi\rangle$, where $H_n$ is the Hadamard transformation, and which halts in time $2n+4$ with its head back on the starting cell. The machine has tape alphabet $\{0,1,\lambda\}$, states $\{q_0,q_a,q_b,q_c,q_f\}$ and the transition function shown in Table 4.1:*

| | $\lambda$ | 0 | 1 |
|---|---|---|---|
| $q_0$ | | $\|0,q_a,\leftarrow\rangle$ | $\|1,q_a,\leftarrow\rangle$ |
| $q_a$ | $\|\lambda,q_b,\rightarrow\rangle$ | | |
| $q_b$ | $\|\lambda,q_c,\leftarrow\rangle$ | $\frac{1}{\sqrt{2}}(\|0,q_b,\rightarrow\rangle+\|1,q_b,\rightarrow,\rangle)$ | $\frac{1}{\sqrt{2}}(\|0,q_b,\rightarrow\rangle-\|1,q_b,\rightarrow\rangle)$ |
| $q_c$ | $\|\lambda,q_f,\rightarrow\rangle$ | $\|0,q_c,\leftarrow\rangle$ | $\|1,q_c,\leftarrow\rangle$ |
| $q_f$ | $\|\lambda,q_0,\rightarrow\rangle$ | $\|0,q_0,\rightarrow\rangle$ | $\|1,q_0,\rightarrow\rangle$ |

Table 4.1: A Turing machine realizing the Hadamard transformation

*The machine processes a string $x$ from $\{0,1\}^*$ as follows. Starting in the state $q_0$ on the leftmost square of $x$ the machine moves left and right and enters the state $q_b$. In this state* *it keeps moving right, replacing each* 0 *by* $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ *and each* 1 *by* $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, *until the first blank is reached. Afterwards the machine keeps moving left until the first blank left from* $x$ *is reached. As the last step it moves right into the final state.*

Concepts of well-behavedness, stationarity, and normal form have their meaning also for deterministic TM. They also play the key role in the following result, due to Bernstein and Vazirani (1997), that modifies Bennett's result on reversibility.

**Theorem 4.2.29 (Synchronization Theorem)** *If* $f$ *is a string-to-string function which can be computed by a DTM in polynomial time and such that the length of* $f(x)$ *depends only on the length of* $x$, *then there is a polynomial time, stationary, normal-form and reversible TM which on any input* $x$ *produces* $x, f(x)$ *and whose running time depends only on the length of* $x$.

**Exercise 4.2.30** *Prove synchronization theorem.*

### Programming primitives

Just as in the classical case, one of the techniques that makes the design and behaviour of QTM more transparent is to consider cells (tapes) as having several tracks. An equivalent way is to use tape alphabets that are cartesian products of subalphabets. Some of the very basic techniques of manipulation with several tracks that can be used in the case of quantum Turing machines are presented in the following exercises.

**Exercise 4.2.31** *Show that given a QTM (RTM)* $\mathcal{M} = \langle\Sigma, Q, \delta\rangle$ *and a set* $\Sigma'$, *there is a QTM (RTM)* $\mathcal{M}' = \langle\Sigma\times\Sigma', Q, \delta'\rangle$ *such that* $\mathcal{M}'$ *behaves as* $\mathcal{M}$ *while leaving its second track unchanged.*

**Exercise 4.2.32** *Show that given any QTM (RTM)* $\mathcal{M} = \langle\Sigma_1 \times \ldots \times \Sigma_k, Q, \delta\rangle$ *and a permutation* $\pi$ *on* $\{1, \ldots, k\}$, *there exists a QTM (RTM)* $\mathcal{M}' = \langle\Sigma_{\pi(1)} \times \ldots \times \Sigma_{\pi(k)}, Q, \delta'\rangle$ *such that* $\mathcal{M}'$ *behaves exactly as* $\mathcal{M}$ *except that its tracks are permuted according to the permutation* $\pi$.

The results of the following two easy exercises will be used below in showing how to compose programs in several ways. The first exercise deals with swapping of transitions. We talk about swapping of outgoing transitions of states $q_1$ and $q_2$ in a QTM with transition function $\delta$, if $\delta$ is replaced by $\delta'$ defined as follows; $\delta'(q_1, \sigma) = \delta(q_2, \sigma)$, $\delta'(q_2, \sigma) = \delta(q_1, \sigma)$ for all tape symbols $\sigma$ and $\delta'(q, \sigma) = \delta(q, \sigma)$ for all $q \notin \{q_1, q_2\}$. In a similar way swapping of incoming transition of two states is defined.

**Exercise 4.2.33** *Show that if* $\mathcal{M}$ *is a well-formed QTM (RTM), then swapping of the incoming or outgoing transitions between a pair of states in* $\mathcal{M}$ *yields another well-formed QTM (RTM).*

**Exercise 4.2.34** *Show that if* $\mathcal{M}_1 = \langle\Sigma, Q_1, \delta_1\rangle$ *and* $\mathcal{M}_2 = \langle\Sigma, Q_2, \delta_2\rangle$ *are two well-formed QTM (RTM) with disjoint sets of states, then the "union" of these two QTM (RTM)* $\mathcal{M} = \langle\Sigma, Q_1 \cup Q_2, \delta_1 \cup \delta_2\rangle$ *is also a well-formed QTM (RTM).*

**Lemma 4.2.35** *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are normal-form QTM (RTM) with the same alphabet and $q$ is a state of $\mathcal{M}_1$, then there is a normal-form QTM $\mathcal{M}$ which acts as $\mathcal{M}_1$ except that each time $\mathcal{M}_1$ would enter the state $q$ it runs $\mathcal{M}_2$ instead of that.*

**Proof.** Let $q_0^{(1)}$, $q_0^{(2)}$, $q_f^{(1)}$ and $q_f^{(2)}$ be the initial and final states of $\mathcal{M}_1$ and $\mathcal{M}_2$. $\mathcal{M}$ will be designed as union of $\mathcal{M}_1$ and $\mathcal{M}_2$ (see Exercise 4.2.34), with the initial state $q_0^{(1)}$ if $q \neq q_0^{(1)}$ and $q_0^{(2)}$ otherwise. In addition, the incoming transitions of $q$ and $q_0^{(2)}$ and also the outgoing transitions of $q$ and $q_f^{(2)}$ will be swapped (see Exercise 4.2.33). According to the above two exercises, the resulting QTM $\mathcal{M}$ is well-formed. Since $\mathcal{M}_1$ is in the normal form the final state of $\mathcal{M}$ leads back to its initial state (no matter whether $q$ is the initial or the final state of $\mathcal{M}_1$ or neither of them). □

Now we can formulate the first two basic QTM design techniques.

**Lemma 4.2.36 (Composition Lemma)** *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are well-formed, normal-form QTM (RTM) with the same alphabet, then there is a normal-form QTM (RTM) $\mathcal{M}$ which performs at first computation of $\mathcal{M}_1$, and after $\mathcal{M}_1$ halts, it performs computation of $\mathcal{M}_2$. (That is $\mathcal{M}$ acts as a composition of $\mathcal{M}_1$ and $\mathcal{M}_2$.)*

**Proof.** Let again $q_0^{(1)}$, $q_0^{(2)}$, $q_f^{(1)}$ and $q_f^{(2)}$ be the initial and final states of $\mathcal{M}_1$ and $\mathcal{M}_2$. To design $\mathcal{M}$ we simply insert $\mathcal{M}_2$ into $\mathcal{M}_1$ for the final state as shown in the proof of Lemma 4.2.35. To show that the resulting machine satisfies the claim of the lemma we proceed as follows.

Since both $\mathcal{M}_1$ and $\mathcal{M}_2$ are in the normal form, the only transition to $q_0^{(1)}$ is from $q_f^{(1)}$ and to $q_0^{(2)}$ from $q_f^{(2)}$ in $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. Therefore no transition in $\mathcal{M}_1$ is changed when inserting $\mathcal{M}_2$ into $\mathcal{M}_1$ except for those into and out of the state $q_f^{(1)}$. $\mathcal{M}_1$ is well-behaved, and therefore when $\mathcal{M}$ starts in $q_0^{(1)}$, it computes exactly as $\mathcal{M}_1$ until $\mathcal{M}_1$ would have entered the state $q_f^{(1)}$. However, in such a case $\mathcal{M}$ will not reach the state $q_f^{(1)}$ but instead of that the state $q_0^{(2)}$. Moreover, since no other transition of $\mathcal{M}$ has been changed in the union of $\mathcal{M}_1$ and $\mathcal{M}_2$ than those into and out of $q_f^{(1)}$, $\mathcal{M}$ will proceed exactly as if $\mathcal{M}_2$ had been started in the superposition computed by $\mathcal{M}_1$. □

**Lemma 4.2.37 (Branching Lemma)** *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be well-behaved normal-form QTM (RTM) with the same alphabet. Then there is a two-track, well-behaved, normal form QTM (RTM) $\mathcal{M}$ such that if the second track is empty, then $\mathcal{M}$ runs $\mathcal{M}_1$ on the first track and leaves its second track empty, and if the second track has a 1 in the start cell (and all other cells blank), then $\mathcal{M}$ runs $\mathcal{M}_2$ on its first track and leaves 1 where its head ends up. In either case $\mathcal{M}$ makes at most 4 steps more than the corresponding machine ($\mathcal{M}_1$ or $\mathcal{M}_2$).*

**Proof.** $\mathcal{M}$ is constructed as follows. At first a stationary, normal-form RTM $R$ is designed which (i) always makes four steps and leaves its input unchanged, (ii) always is in a single configuration superposition, and (iii) has two distinguished states $q_1$ and $q_2$ with the following properties. If $R$ is run with 1 in the starting cell and blanks elsewhere, then $R$ visits $q_1$ with the blank tape and with the head on the starting cell and does not visit $q_2$

at all. On the other hand, if $R$ is run with the blank tape, then $R$ visits $q_2$ once with the blank tape and does not visit $q_1$ at all. $R$ will now be used to design $\mathcal{M}$.

At first $\mathcal{M}_1$ and $\mathcal{M}_2$ are extended to have also the second track with the alphabet of $R$—see Exercise 4.2.31. Afterwards $R$ is extended to have a first track with the common alphabet of $\mathcal{M}_1$ and $\mathcal{M}_2$. Finally, $\mathcal{M}_1$ and $\mathcal{M}_2$ are inserted for the states $q_1$ and $q_2$, respectively.

$R$ will have the alphabet $\{\lambda, 1\}$, the set of states $\{q_0, q_1, q_1', q_2, q_2', q_3, q_f\}$ and transitions:

$$\begin{array}{lll} (q_0, \lambda, \lambda, q_2', \leftarrow) & (q_0, 1, \lambda, q_1', \leftarrow) & (q_1', \lambda, \lambda, q_1, \rightarrow) \\ (q_1, \lambda, 1, q_3, \leftarrow) & (q_2', \lambda, \lambda, q_2, \rightarrow) & (q_2, \lambda, \lambda, q_3, \leftarrow) \\ (q_3, \lambda, \lambda, q_f, \rightarrow) & (q_f, \lambda, \lambda, q_0, \rightarrow) & (q_f, 1, 1, q_0, \rightarrow) \end{array}$$

□

Loops are the third main classical design tool. Building an RTM that loops indefinitely is easy. Finite loops can be realized in the following way.

**Lemma 4.2.38 (Looping Lemma)** *There is a stationary, normal-form RTM $\mathcal{M}$ and a constant $c$ with the following property. Given as input an integer $k$, written in binary, $\mathcal{M}$ makes $\mathcal{O}(k \lg^c k)$ steps and halts with its tape unchanged. Moreover, $\mathcal{M}$ has a special state $q^*$ which $\mathcal{M}$ visits exactly $k$ times, each time with its head on the starting cell.*

**Exercise 4.2.39** *Prove the looping lemma.*

**Remark 4.2.40** Observe that all constructions of this section that support main classical algorithm design primitives—composition, branching, and looping—can be implemented without any inherently quantum step.

## 4.3 Quantum Cellular Automata

Cellular automata (CA) have been invented to model the biological world. Using this model von Neumann gave in the 1950s a positive answer to one of the intriguing questions of those times, namely whether machines can reproduce themselves.

During the last four decades, cellular automata have been studied from very many points of view and used to model a surprisingly broad variety of phenomena. The attempts to consider cellular automata as a model of the physical world (see Fredkin, 1990) naturally lead to an intensive study of the reversibility issues of cellular automata. Cellular automata are a model of computation so closely related to the physical world that some of the very first attempts to create quantum versions of the classical models of computation have been oriented on cellular automata. Surprisingly, it took quite a while for a proper model to emerge—see Watrous (1995) and Dürr, LêThanh, and Santha (1996)—for quantum one-dimensional cellular automata.

### 4.3.1 Classical cellular automata

Informally, a $d$-dimensional cellular automaton $\mathcal{A}$ with a finite set of **states** $Q$, a finite **neighbourhood** $N \subseteq \mathbf{Z}^d$, and a **local transition function** $\delta : Q^{|N|} \rightarrow Q$, is a $d$-dimensional array network of identical finite automata with $Q$ as their set of states. For each node $n \in \mathbf{Z}^d$, the neighbourhood $N$ determines the set $\{n\} + N$ of $|N|$ "neighbours" of the node $n$. Formally, $\mathcal{A} = \langle d, Q, N, \delta \rangle$ and elements (nodes) of $\mathbf{Z}^d$ are regarded as representing those finite automata which $\mathcal{A}$ consists of.

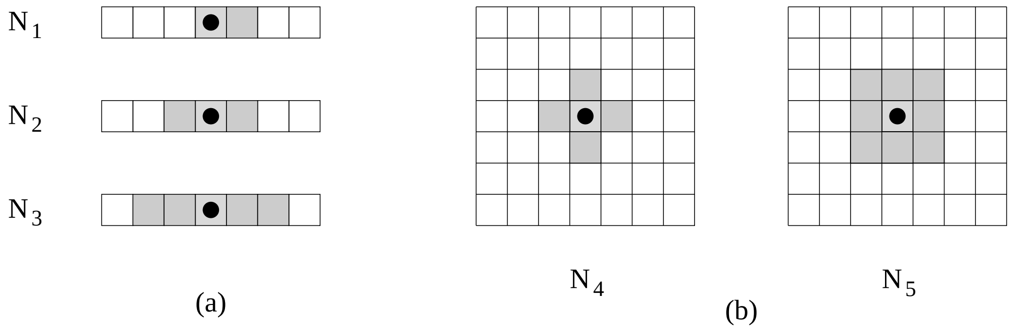


Figure 4.4: Neighbourhoods for one- and two-dimensional cellular automata-65%

Figure 4.4a depicts three neighbourhoods of the dotted nodes for one-dimensional cellular automata: $N_1 = \{0, 1\}$, $N_2 = \{-1, 0, 1\}$, $N_3 = \{-2, -1, 0, 1, 2\}$. Figure 4.4b illustrates two often-used neighbourhoods (again for the dotted nodes) for two-dimensional cellular automata: **von Neumann neighbourhood** $N_4 = \{(-1, 0), (0, -1), (0, 0), (0, 1), (1, 0)\}$, and **Moore neighbourhood** $N_5 = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}$.

All finite automata of a cellular automaton work concurrently, synchronized, and in discrete time steps. At each time moment the new state of each finite automaton is defined to be the value of the local transition function applied to the current states of a cell and all its neighbours.

In order to describe more formally the overall behaviour of a cellular automaton $\mathcal{A} = \langle d, Q, N, \delta \rangle$ the concept of configuration is again the basic one.

A **configuration** of $\mathcal{A}$ is an element of $c \in Q^{\mathbf{Z}^d}$ assigning to each finite automaton of $\mathcal{A}$ (that is to each node of $\mathbf{Z}^d$) a state. The **global transition function**

$$G_\delta : Q^{\mathbf{Z}^d} \rightarrow Q^{\mathbf{Z}^d},$$

$$G_\delta(c)(w) = \delta(c(\{w\} + N)),$$

is used to define a **computation (behaviour)** of $\mathcal{A}$ to be a sequence of configurations $c_0, c_1, c_2, \ldots$ such that $c_i = G_\delta(c_{i-1})$, for $i \geq 1$.

For example, if $d = 1$, $N = \{-1, 0, 1\}$, then $G_\delta(c)(i) = \delta(c(i-1), c(i), c(i+1))$, if $N = \{-k, -k+1, \ldots, -1, 0, 1, \ldots, k-1, k\}$, then

$$G_\delta(c)(i) = \delta(c(i-k), c(i-k+1), \ldots, c(i+k-1), c(i+k));$$

if $d = 2$ and the von Neumann neighbourhood is considered, then

$$G_\delta(c)(i, j) = \delta(c(i-1, j), c(i, j-1), c(i, j), c(i, j+1), c(i+1, j)).$$

Of special interest for us are reversible cellular automata.

**Definition 4.3.1** *A cellular automaton $\mathcal{A} = \langle d, Q, N, \delta\rangle$ is reversible, if there is another cellular automaton $\mathcal{A}' = \langle d, Q, N', \delta'\rangle$ such that for each configuration $c$ of $\mathcal{A}$ it holds*

$$G_\delta(c) = c_1 \qquad \textit{if and only if} \qquad G_{\delta'}(c_1) = c.$$

In other words, a cellular automaton $\mathcal{A}$ is reversible if there is another cellular automaton $\mathcal{A}'$ such that for any sequence of configurations $c_1, c_2, \ldots, c_{n-1}, c_n$ of $\mathcal{A}$, where $G_\delta(c_i) = c_{i+1}$, for $1 \leq i < n$, $\mathcal{A}'$ can reverse this computation to get the following sequence of configurations $c_n, c_{n-1}, \ldots, c_2, c_1, i.e., G_{\delta'}(c_{i+1}) = c_i$. (Observe that the reverse cellular automaton $\mathcal{A}'$ may use a much smaller or larger neighbourhood than $\mathcal{A}$ does.)

**Example 4.3.2** *A simple four-state cellular automaton, due to I. Korec, with the neighbourhood $\{0, 1\}$ is depicted in Figure 4.5a and its reversible counterpart, with the neighbourhood $\{-1, 0\}$ is shown in Figure 4.5b.*

| * | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 2 | 3 | 3 | 2 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 2 | 3 | 3 | 2 |

(a)

| * | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 3 |
| 1 | 2 | 2 | 1 | 1 |
| 2 | 0 | 0 | 3 | 3 |
| 3 | 2 | 2 | 1 | 1 |

(b)

Figure 4.5: A cellular automaton and its reversible counterpart

There do not seem to be many reversible cellular automata. For two-state automata with a neighbourhood $N$ where $|N| = 2$ or $|N| = 3$ there are none. For the neighbourhood $N = \{-1, 0, 1, 2\}$ there are 65 536 cellular two-state automata but only 8 of them are reversible and all of them are insignificant modifications of the same one.[7] The following theorem, due to Toffoli (1977), Dubacq (1985) and Kari (1990), of importance for cellular automata applications, is therefore quite a surprise.

**Theorem 4.3.3** *(1) Any $k$-dimensional CA can be simulated in real time by a $(k+1)$-dimensional reversible CA. (2) There is a universal cellular automaton that is reversible. (3) It is decidable whether a one-dimensional cellular automaton is reversible but it is undecidable whether a two-dimensional cellular automaton is reversible.*

[8]

**Exercise 4.3.4** *Show that the one-dimensional cellular automaton with the neighbourhood $N = \{0, 1\}$, states $\{0, 1, \ldots, 9\}$ and the transition function $\delta(x, y) = (5x + \lceil\frac{5y}{10}\rceil) \bmod 10$ is reversible.*

[7] These 8 reversible CA have been found by a brute force search. For details see (http://alife.santafe.edu/alife/topics/cas/ca-faq/properties/properties.html). Amoroso and Patt (1972) found some 4-neighbour reversible CA and presented a general construction to get an infinite family of reversible CA—(Ch. Dürr).

[8] For more about cellular automata and their relations to othe models of computation see, for example, Gruska (1997).

### 4.3.2 One-dimensional quantum cellular automata

The most basic model of cellular automata is that of the one-dimensional cellular automaton, and we deal in this section with its quantum version, due to Watrous (1995).

A **one-dimensional quantum cellular automaton**, in short 1QCA, $\mathcal{A} = \langle Q, \lambda, N, \delta\rangle$ is determined by a finite set $Q$ of states that includes one so-called *quiescent state* $\lambda$, a neighbourhood $N = \{n_1, \ldots, n_r\} \subseteq \mathbf{Z}$ with $n_1 < n_2 < \ldots < n_r$ and a local transition function

$$\delta : Q^{r+1} \to \mathbf{C}_{[0,1]}$$

satisfying the following three conditions:

1. **Local probability condition**: for any $(q_1, \ldots, q_r) \in Q^r$,

$$\sum_{q\in Q} |\delta(q_1, \ldots, q_r, q)|^2 = 1,$$

2. **Stability of the quiescent state condition:** if $q \in Q$, then

$$\delta(\lambda, \ldots, \lambda, q) = \begin{cases} 1, & \text{if } q = \lambda; \\ 0, & \text{otherwise.} \end{cases}$$

(Sometimes it is more convenient to consider an equivalent definition of $\delta$, as a mapping from $Q^r$ to $\mathbf{C}_{[0,1]}^Q$.) In order to define the third condition several concepts have to be introduced.

A configuration $c : \mathbf{Z} \to Q$ is a mapping such that $c(i) \neq \lambda$ only for finitely many $i$. Let $C(\mathcal{A})$ denote the set of all configurations. Computation of $\mathcal{A}$ is then done in the inner-product space $H_{\mathcal{A}} = l_2(C(\mathcal{A}))$ with the basis $\{|c\rangle \mid c \in C(\mathcal{A})\}$.

In one step $\mathcal{A}$ transfers from one basis state $|c_1\rangle$ to another $|c_2\rangle$. The amplitude of such a transition, $\alpha(c_1, c_2)$, is defined as follows:

$$\alpha(c_1, c_2) = \prod_{i\in\mathbf{Z}} \delta(c_1(i + n_1), c_1(i + n_2) \ldots c_1(i + n_r), c_2(i)).^9$$

A state in $H_{\mathcal{A}}$ has therefore in general the form

$$|\phi\rangle = \sum_{c\in C(\mathcal{A})} \alpha_c|c\rangle, \text{ where } \sum_{c\in C(\mathcal{A})} |\alpha_c|^2 = 1.$$

The evolution operator $E_{\mathcal{A}}$ of $\mathcal{A}$ maps any state $|\phi\rangle \in l_2(C(\mathcal{A}))$ into the state $|\psi\rangle = E_{\mathcal{A}}|\phi\rangle$ such that

$$|\psi\rangle = E_{\mathcal{A}}|\phi\rangle = \sum_{c\in C(\mathcal{A})} \beta_c|c\rangle, \text{ where } \beta_c = \sum_{c'\in C(\mathcal{A})} \alpha_{c'}\alpha(c', c).$$

Now we can formulate the third condition $\delta$ has to satisfy:

3. **Unitarity of evolution condition:** the mapping $E_{\mathcal{A}}$ has to be unitary.

If $\mathcal{A}$ is such that the first two conditions are satisfied, then we say that $\mathcal{A}$ is a quasi-1QCA (in short q1QCA).

[9] It follows from the stability of the quiescent state condition that this infinite product is a well-defined complex number.

**Remark 4.3.5** As mentioned at the beginning of this section, in classical cellular automata a configuration is usually defined as a mapping $c : \mathbf{Z} \rightarrow Q$ without the restriction that $c(i) \neq \lambda$ only for finitely many $i$, and therefore one also has so-called infinite configurations. Such an approach is normally used when the behaviour of cellular automata is studied as that of complex systems. If the computational aspects of classical cellular automata are investigated, only finite configurations are considered.

A quantum cellular automaton $\mathcal{A}$ is called **trivial** if its neighbourhood is $N = \{0\}$, i.e. each cell has only itself as the "neighbour". $\mathcal{A}$ is called **simple** if its neighbourhood is an interval of integers, i.e. $n_i = n_1 + i - 1$ for $1 < i \leq r$. $\mathcal{A}$ is called **symmetric** if it is simple and $n_r = -n_1$. A special case of a symmetric 1QCA is the one with the neighbourhood $N = \{-1, 0, 1\}$. For such a basic 1QCA the notation $\mathcal{A} = \langle Q, \lambda, \delta \rangle$ and QCA will be used.

Given a q1QCA a nontrivial problem is to decide whether the unitarity condition is satisfied. If we take as the size of a 1QCA $size(\mathcal{A}) = |Q|^{|N|+1}$, because the description of $\mathcal{A}$ is dominated by the description of its transition function, then for simple 1QCA the unitarity condition can be verified (by Dürr and Santha, 1996), in time $\mathcal{O}(n^{\frac{3r+1}{r+1}}) = \mathcal{O}(n^3)$, where $n$ is the size of $\mathcal{A}$ and $r$ is the size of the neighbourhood.

**Exercise 4.3.6** *What is a good upper bound for deciding unitarity of arbitrary 1QCA?*

Unitarity is easy to verify for trivial 1QCA; we can make use of this in Section 4.3.3.

**Lemma 4.3.7** *Evolution of a trivial 1QCA is unitary if and only if the following condition holds for any $q_1, q_2 \in Q$.*

$$\sum_{q \in Q} \delta^*(q_1, q)\delta(q_2, q) = \begin{cases} 1, & \text{if } q_1 = q_2 \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

**Proof.** In this case $\delta$ is a mapping from $Q \times Q$ to $\mathbf{C}_{[0,1]}$. Each cell of $\mathcal{A}$ evolves independently and actually as a 1QFA with one-letter inputs. Condition (4.7) is a necessary and sufficient condition for unitarity of the evolution of such a 1QFA (if input symbols are discarded). The evolution matrix of such a 1QFA is therefore given by the matrix $U$ of dimension $|Q|$ such that $U(q_1, q_2) = \delta(q_2, q_1)$. The evolution matrix for any initial configuration of length $n$ is then given by the matrix $\bigotimes_{i=1}^{n} U$. ∎

The transition function of a q1QCA is said to satisfy the **well-formedness condition** if for every state $|\phi\rangle \in H_{\mathcal{A}}$, it holds

$$||\phi|| = 1 \Leftrightarrow ||E_{\mathcal{A}}|\phi\rangle|| = 1. \tag{4.8}$$

Satisfiability of such a condition can be verified in time $\mathcal{O}(n^2)$ by Dürr, LêThanh and Santha (1996). The satisfiability of the above well-formedness condition is not sufficient for the unitarity condition to hold, as the following example shows:

**Example 4.3.8** *Each classical one-dimensional CA $\mathcal{A}$ can be seen as a q1QCA. Its evolution operator is a matrix with the property that in every column indexed by a configuration there is a single non-zero entry with value* 1 *(for the unique next configuration). This matrix has all columns of norm* 1*. Its columns are all pairwise orthogonal if and only if the automaton mapping is injective. Its evolution is unitary if and only if its global mapping is bijective.*

*Any classical 1CA whose global evolution is injective but not surjective is well-formed but not unitary.*

*For example, due to Ch. Dürr, in the one-dimensional classical CA $\mathcal{A}$, called "controlled not", with states $\{0,1\}$, the quiescent state 0, the neighbourhood $N = \{0,1\}$, and the transition function $\delta(x,y) = x \oplus y$, no configuration of the form $0^*10^*$ has a pre-image (among configurations in $C_{\mathcal{A}}$). The global transition function of the automaton is injective but not surjective and the automaton satisfies the well-formedness condition.*

**Remark 4.3.9** The history of attempts to introduce the concept of quantum cellular automata goes back to Grössing and Zeilinger (1988) and a series of subsequent papers of the same authors. However, their model has little in common with the one discussed above. This history well illustrates the merits and limits of methodologies developed in physics and theoretical computing to deal with such basic problems and to come up with proper concepts.

**Remark 4.3.10** In order to study more in depth computability problems for QCA and to consider the problem of universality, some restrictions have to be made on the types of amplitudes allowed. For example, for each amplitude $\alpha$ there has to be an algorithm to compute $\alpha$ to $2^{-n}$ in time polynomial in $n$; or that all amplitudes are rational.

Even for 1QCA several basic problems are still open. One of them concerns universality. Is there a single 1QCA which would be universal in a reasonable sense for the whole class of 1QCA and could efficiently simulate any 1QCA?[10]

**Remark 4.3.11** (CH. Dürr). A suitable definition of two- and more- dimensional quantum cellular automata is an untrivial issue. Formally, it can be done by a similar modification of the definition of the classical cellular automata—to assign complex amplitudes to transitions—as in the case of one-dimensional cellular automata. The difficulty with such a straightforward approach is that in such a case one cannot decide in polynomial time whether a qQCA is really a QCA. This follows from the result of Kari (1990) that reversibility of two-dimensional cellular automata is undecidable.

Another basic problem not yet fully solved is the problem of mutual efficient simulation of quantum cellular automata and quantum Turing machines. There is so far only a partial solution of this problem for a special class of 1QCA, for the so-called **partitioned quantum cellular automata** (PQCA). They are a natural quantum version of the model introduced by Morita and Harao (1989) and have been shown to be important in the classical case because they are much more easy to deal with. For PQCA it is also easy to verify whether their evolution is unitary. Results presented in the following section follow Watrous (1995).

### 4.3.3 Partitioned quantum one-dimensional cellular automata

For the very basic version of the partitioned one-dimensional cellular automaton (P1QCA) with which we deal in the next section, a PQCA is defined similarly to a QCA but each state

[10] van Dam (1996) solved positively the universality problem for a special "circular" model of 1QCA, in which cells may form cycles of various length. Or, in other words, for space-periodic configurations only. This interesting result did not seem to contribute much to the solution of the basic universality problem for 1QCA. However, as shown by Dürr (1997), if a q1QCA is QCA with respect to van Dam model, then it is also 1QCA with respect to our model. This implies that a subset of 1QFA has nice properties from van Dam's model. For example, this subset has a universal instance and can be simulated by a QTM.

(cell) is considered as being partitioned into three substates (subcells): left one, middle one and right one. The next state of any cell depends on the right substate of the left neighbour, the middle substate of the current cell and on the left substate of the right neighbour (see Figure 4.6a).

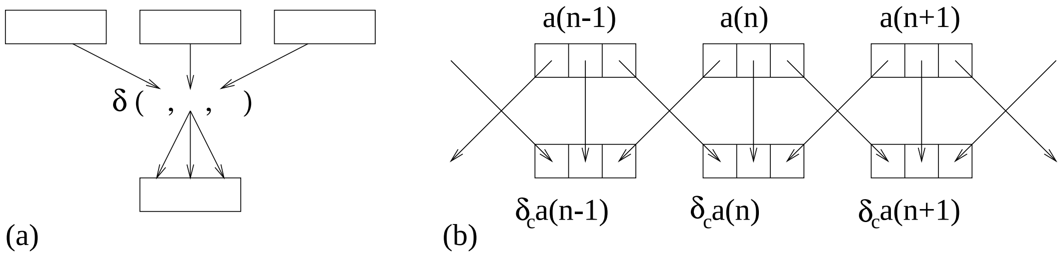


Figure 4.6: Evolution in a partial QCA-70%

In the full generality a partitioned one-dimensional quantum cellular automaton is a 1QCA $\mathcal{A} = \langle Q, \lambda, N, \delta \rangle$ which satisfies the following restrictions.

1. The set of states $Q$ is the Cartesian product $Q = Q_1 \times \ldots \times Q_r$ of $r = |N|$ nonempty sets. (In the case of the neighbourhood $N = \{-1, 0, 1\}$ we use notation $Q = Q_l \times Q_m \times Q_r$.)
2. The local transition function $\delta : Q^r \rightarrow \mathbf{C}^Q$ is the composition of two functions:
   (a) $\delta_c : Q^r \rightarrow Q$; a classical mapping.
   (b) $\delta_q : Q \rightarrow \mathbf{C}^Q$; a quantum mapping,

   where

$$\delta_c((q_{1,1}, \ldots, q_{1,r}), (q_{2,1}, \ldots, , q_{2,r}), \ldots, (q_{r,1}, \ldots, q_{r,r})) = (q_{1,1}, q_{2,2}, \ldots, q_{r,r}).$$

The function $\delta_c$ defines a 1CA (one-way CA) $\mathcal{A}_c = \langle Q, \lambda, N, \delta_c \rangle$ and the global transition function $G_{\delta_c}$ is a permutation on configurations of $\mathcal{A}$ such that for any $a \in C_{\mathcal{A}}$, $i \in N$, $[G_{\delta_c}(a)](i) = \delta_c(a(i+N))$. The case $N = \{-1, 0, 1\}$ is illustrated in Figure 4.6b.

$\mathcal{A}_c$ can also be seen as a 1QCA whose evolution operator $U_{\mathcal{A}_c}$ is defined by

$$U_{\mathcal{A}_c}(q_2, q_1) = \begin{cases} 1, & G_{\delta_c}(q_1) = q_2; \\ 0, & \text{otherwise}, \end{cases}$$

and therefore $U_{\mathcal{A}_c}$ is unitary.

The function $\delta_q$ defines a trivial 1QCA, $\mathcal{A}_q = \langle Q, \lambda, \{0\}, \delta_q \rangle$. Its evolution operator $U_{\mathcal{A}_q}$ is defined by

$$U_{\mathcal{A}_q}(q_2, q_1) = [\delta_q(q_1)](q_2).$$

$U_{\mathcal{A}_q}$ is also called the local transition matrix of $\mathcal{A}$. Unitarity of P1QCA is now easy to verify. The following theorem is due to Watrous (1995) and its proof is due to Dürr et al. (1996).

**Theorem 4.3.12** *Evolution of a P1QCA $\mathcal{A}$ is unitary if and only if the evolution of the corresponding trivial 1CA is unitary, and this holds if and only if the local transition matrix of $\mathcal{A}$ is unitary.*

**Proof.** Since the local transition function $\delta$ of $\mathcal{A}$ is a composition of local transition functions $\delta_c$ and $\delta_q$, the evolution operator $U_{\mathcal{A}}$ is the following product of the evolution operators $U_{\mathcal{A}} = U_{\mathcal{A}_q} U_{\mathcal{A}_c}$. Since $U_{\mathcal{A}_c}$ is trivially unitary we can state that evolution of $\mathcal{A}$ is unitary if and only if the local transition matrix $U_{\mathcal{A}_q}$ is unitary. ∎

As a consequence $\mathcal{A}$ can be specified using its local transition matrix $U$, i.e. in the form $\mathcal{A} = \langle Q, \lambda, N, U\rangle$ or, when the neighbourhood $N$ is known, as $\mathcal{A} = \langle Q, \lambda, U\rangle$.

**Example 4.3.13 (Watrous, 1997)** *We describe a simple PQCA $\mathcal{A} = \langle Q, \lambda, U\rangle$ that simulates, in a sense, the EPR phenomenon. Let*

$$Q = Q_l \times Q_m \times Q_r, \text{ where } Q_l = Q_r = \{0, +, -\}, Q_m = \{0\}, \lambda = (0,0,0).$$

*The matrix $U$ of degree $|Q|$ is defined by*

$$U(q', q) = \begin{cases} 1, & \text{if } q = q' \notin S = \{(-,0,+), (+,0,-)\}; \\ 0, & \text{if } q \neq q' \text{ and } \{q, q'\} \neq S, \end{cases}$$

*and for $q, q' \in S$,*

$$U(q', q) = \begin{cases} -\frac{1}{\sqrt{2}}, & \text{if } q = q' = (-,0,+); \\ \frac{1}{\sqrt{2}}, & \text{otherwise.} \end{cases}$$

*If $\mathcal{A}$ evolves from the initial basis state $|c\rangle$, where $c$ is defined by*

$$c(n) = \begin{cases} (0,0,-), & \text{if } n = -1; \\ (+,0,0), & \text{if } n = 1; \\ \lambda, & \text{otherwise,} \end{cases}$$

*then, after one step, $\mathcal{A}$ is in the superposition*

$$\frac{1}{\sqrt{2}}(|c_0\rangle + |d_0\rangle),$$

*where*

$$c_0(n) = \begin{cases} (+,0,-), & \text{if } n = 0; \\ (0,0,0), & \text{otherwise,} \end{cases} \qquad d_0(n) = \begin{cases} (-,0,+), & \text{if } n = 0; \\ (0,0,0), & \text{otherwise,} \end{cases}$$

*By an induction one can show that after $t > 1$ steps $\mathcal{A}$ is in the superposition*

$$\frac{1}{\sqrt{2}}(|c_{t-1}\rangle + |d_{t-1}\rangle),$$

*where*

$$c_t(n) = \begin{cases} (+,0,0), & \text{if } n = -t; \\ (0,0,-), & \text{if } n = t; \\ (0,0,0), & \text{otherwise,} \end{cases} \qquad d_t(n) = \begin{cases} (-,0,0), & \text{if } n = -t; \\ (0,0,+), & \text{if } n = t; \\ (0,0,0), & \text{otherwise.} \end{cases}$$

*If the states $(0,0,-)$ and $(-,0,0)$ are interpreted as negative particles and $(0,0,+), (+,0,0)$ as positive particles, then the configuration $|c_t\rangle$ models the situation in which the positive particle moves to the left and the negative one to the right; the configuration $|d_t\rangle$ models the situation when particles are reversed.*

### 4.3.4 Quantum cellular automata versus quantum Turing machines

The problem of effective mutual simulations of 1QCA and one-tape quantum Turing machines has been solved so far only partially.

As shown below, each QTM the head of which moves always either to left or to right, can be efficiently simulated by a PQCA, and therefore by a QCA. On the other hand, only PQCA have been shown, so far, to be efficiently simulated by QTM.

One way to discuss mutual simulation of QTM and QCA is to consider both QTM and QCA as acceptors and to define when one machine model simulates the other one.

The problem of simulation of 1QCA by QTM does not seem to be easy. One reason for that may be the fact that QTM and 1QCA differ in the following property. While to each QTM $\mathcal{M}$ there exists a QTM $\mathcal{M}'$ that runs $\mathcal{M}$ backwards, this is not always true for 1QCA as the following example, due to Ch. Dürr, shows.

**Example 4.3.14** *Let us consider 1QCA* $\mathcal{A} = \langle\{0,1\},\{0,1\},\delta\rangle$, *where* $\delta(x,y) = |x\rangle$ *if* $y = 0$ *and* $\delta(x,y) = H'|x\rangle$ *if* $y = 1$ *(with* $H' = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$*). Denote by* $U_{\mathcal{A}}$ *the global transition function of* $\mathcal{A}$ *and let* $|c_k\rangle$ *be the configuration of* $\mathcal{A}$ *with* 1 *in the cells with index* $\{-k,-1\}$ *and* 0 *otherwise.*

*It can be shown, using algorithm due to Dürr and Santha (1996), that* $U_{\mathcal{A}}$ *is unitary and that*

$$|\phi\rangle = U_{\mathcal{A}}^{-1}|c_1\rangle = \sum_{k\geq 1}\frac{1}{\sqrt{2^k}}(-1)^{k-1}|c_k\rangle.$$

$U_{\mathcal{A}}^{-1}$ *is not the global transition function of a 1QCA. This follows from any of the following two observations. (1) In* $|\phi\rangle$ *"all cells with negative index are entangled". At the same time in any 1QCA the image of a "classical configuration" has only "unentangled states". (2) If* $|c\rangle$ *is the all-zero-configuration, then* $U_{\mathcal{A}}^{-1}|c\rangle = |c\rangle$. *This implies that* $U_{\mathcal{A}}^{-1}|c_1\rangle$ *has an infinite range influence. Hence all cells with negative index depend on whether there is* 0 *or* 1 *in the cell with index* $-1$.

**Exercise 4.3.15** *(a) Replace in the definition of the 1QCA from Example 4.3.14* $H'$ *with the square-root-of-not matrix, page 63, and show that the inverse of the global evolution of the resulting 1QCA is not evolution of any 1QCA; (b) find necessary and sufficient condition that if the matrix* $H'$ *from Example 4.3.14 is replaced by a matrix* $U$, *then the resulting 1QCA has such a global evolution function the inverse of which is not an evolution function of a 1QCA.*

In order to consider QTM as acceptors two additional components of QTM will be needed: an integer $k$ denoting the distinguished acceptance tape square (cell) and a set $\Sigma_a$ of accepting tape symbols. An input $x \in \Sigma^*$ is accepted by a QTM $\mathcal{M} = \langle\Sigma, Q, q_0, \delta, k, \Sigma_a\rangle$, if a symbol from $\Sigma_a$ is written into the tape square $k$, when $\mathcal{M}$ process the input $x$.

In order to consider 1QCA as acceptors also two additional components have to be considered: an integer $k$ denoting the distinguished acceptance cell and a set $Q_a$ of accepting states. An input $x \in Q^*$ will be accepted by a 1QCA $\mathcal{A} = \langle Q, \lambda, \delta, k, Q_a\rangle$ if the cell number $k$ gets into a state from $Q_a$ during the computation with $x$ as the initial configuration.

**Definition 4.3.16** *We say that a PQCA* $\mathcal{A} = \langle Q, \lambda, U, k_{\mathcal{A}}, Q_a\rangle$ *simulates a QTM* $\mathcal{M} = \langle\Sigma, Q_{\mathcal{M}}, \delta, k_{\mathcal{M}}, \Sigma_a\rangle$ *if and only if there are linear-time computable functions* $T : C_{\mathcal{M}} \to C_{\mathcal{A}}$

*and* $f : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ *such that for any configuration* $c \in C_{\mathcal{M}}$ *the probability that* $\mathcal{M}$ *accepts* $c$ *after* $t$ *steps, i.e., that* $\mathcal{M}$ *comes to a configuration with a terminating state, is equal to the probability that* $\mathcal{A}$ *accepts* $T(c)$ *after* $f(t, |T(c)|)$ *steps.*

The first problem we will deal with is the simulation of a uQTM (see Definition 4.2.12), by a PQCA. By Theorem 4.2.14, the restriction to unidirectional QTM is not essential. Let a uQTM $\mathcal{M} = \langle \Sigma, K, \delta, k, \Sigma_a \rangle$ be given. In such a case $K$ can be partitioned into the subsets $K_l$ and $K_r$. $K_l$ ($K_r$) contains those states into which $\mathcal{M}$ comes only when the head moves left (right).

A PQCA $\mathcal{A} = \langle Q, \lambda, U, k_{\mathcal{A}}, Q_a \rangle$ which simulates $\mathcal{M}$ is defined as follows: $Q = Q_l \times Q_m \times Q_r$ with

$$Q_l = K_l \cup \{\#\}, \quad Q_m = \mathbf{\Sigma}, \quad Q_r = K_r \cup \{\#\},$$

where $\# \notin K$, $\lambda = (\#, b, \#)$, where $b$ is the blank symbol of $\mathcal{M}$. The transition matrix $U$ is defined as follows:

1. For each $(s_1, \tau_1), (s_2, \tau_2) \in K_l \times \Sigma$ let
$$U((s_2, \tau_2, \#), (s_1, \tau_1, \#)) = \delta(s_1, \tau_1, \tau_2, s_2, \leftarrow).$$
2. For each $(s_1, \tau_1) \in K_l \times \Sigma$, $(s_2, \tau_2) \in K_r \times \Sigma$, let
$$U((\#, \tau_2, s_2), (s_1, \tau_1, \#)) = \delta(s_1, \tau_1, \tau_2, s_2, \rightarrow),$$
$$U((s_1, \tau_1, \#), (\#, \tau_2, s_2)) = \delta(s_2, \tau_2, \tau_1, s_1, \leftarrow).$$
3. For each $(s_1, \tau_1), (s_2, \tau_2) \in K_r \times \Sigma$, let
$$U((\#, \tau_2, s_2), (\#, \tau_1, s_1)) = \delta(s_1, \tau_1, \tau_2, s_2, \rightarrow).$$
4. For any $q_1, q_2 \in Q$ for which $U(q_1, q_2)$ has not been defined in (1) to (3), let
$$U(q_1, q_2) = \begin{cases} 1, & \text{if } q_1 = q_2; \\ 0, & \text{otherwise.} \end{cases}$$

Let $k_{\mathcal{A}} = k$ and $Q_a = \{(q_l, q_m, q_r) \in Q \mid q_m \in \Sigma_a\}$. It can be shown in a straightforward way that if $\mathcal{M}$ is a QTM the transition function of which satisfies conditions on page 166, then $\mathcal{A}$ is a PQCA with the unitary matrix $U$ and therefore the evolution of $\mathcal{A}$ is unitary.

In order to consider a simulation of $\mathcal{M}$ by $\mathcal{A}$ we need to define the mappings $T : \mathcal{C}_{\mathcal{M}} \to \mathcal{C}_{\mathcal{A}}$ and $f : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$.

A configuration $c \in C_{\mathcal{M}}$ has the form $xqy$, where $x, y \in \Sigma^*, q \in K$. The mapping $T$ is then defined as follows

$$T(xqy)(n) = \begin{cases} (\#, xy_n, \#), & \text{if } q \in K_l, n \neq |x| + 2; \\ (q, xy_n, \#), & \text{if } q \in K_l, n = |x| + 2; \\ (\#, xy_n, \#), & \text{if } q \in K_r, n \neq |x|; \\ (\#, xy_n, q), & \text{if } q \in K_r, n = |x|, \end{cases}$$

where $xy_n$ denotes the $n$th symbol of the string $xy$, or $\lambda$ if out of the range.

**Exercise 4.3.17** *Show, for any* $n \in \mathbf{N}$, *that the probability that* $\mathcal{M}$ *accepts the initial configuration* $q_0 w$ *after* $n$ *steps is equal to the probability that* $\mathcal{A}$ *accepts* $T(q_0 w)$ *after* $n$ *steps.*

To summarize:

**Theorem 4.3.18** *(Watrous, 1995) To any QTM $\mathcal{M}$ there is a PQCA simulating $\mathcal{M}$ with a constant slowdown.*

The concept of simulation of PQCA by QTM is defined in an analogous way as in Definition 4.3.16.

We show now how to construct, given a PQCA $\mathcal{A} = \langle Q, \lambda, U, k, Q_a\rangle$, a QTM $\mathcal{M} = \langle \Sigma, K, q_0, \delta, k', K_a\rangle$ which simulates $\mathcal{A}$. In order to do that we assume that tape cells of $\mathcal{A}$ and of $\mathcal{M}$ are indexed. Each tape cell of $\mathcal{M}$ will represent the cell of $\mathcal{A}$ of the same index. Let $Q = Q_l \times Q_m \times Q_r$. We define

$$\Sigma = Q \cup Q' \cup \{b\},$$

where $Q'$ is a "copy" of $Q$ in the sense that $q' = (q'_l, q'_m, q'_r) \in Q'$ if and only if $q = (q_l, q_m, q_r) \in Q$, and $b$ is the blank symbol of $\mathcal{M}$. The states from $Q'$ will be used as endmarkers to limit the non-blank portion of the tape.

Moreover, let

$$K = Q_l \times \{s_0, s_1, s'_1, s_2, s'_2, s_3\} \times Q_r,$$

where $s_0, s_1, s'_1, s_2, s'_2$ and $s_3$ are new states not used in $Q_m$. Hence each state of $\mathcal{M}$ consists of three components: one element of $Q_l$, one from the set $\{s_0, s_1, s'_1, s_2, s'_2, s_3\}$ and one from $Q_r$. The elements from $Q_l$ and $Q_r$ are needed in order to "move" substates from one tape square to another when performing the permutation $\delta_c$ described on page 183.

The simulation of one step of $\mathcal{A}$ is done in three phases. The task of the first phase is to perform reversibly the permutation $\delta_c$ on the given configuration of $\mathcal{A}$ represented on the tape of $\mathcal{M}$. This is done in one left-to-right and one right-to-left pass of the head between the cells with endmarkers. In the left-to-right move the leftmost components of the triple states are shifted one cell (triple) to the right. In the right-to-left move the rightmost components of triple states are shifted one triple to the left. All that is done by the moves defined for each $(l, r) \in Q_l \times Q_r$, $(q_l, q_m, q_r), (q'_l, q'_m, q'_r) \in Q_l \times Q_m \times Q_r$:

$$\begin{aligned}
\delta((l, s_0, r), (q_l, q_m, q_r)', (q_l, q_m, r)', (l, s_1, q_r), \rightarrow) &= 1,\\
\delta((l, s_1, r), (q_l, q_m, q_r), (q_l, q_m, r), (l, s_1, q_r), \rightarrow) &= 1,\\
\delta((l, s_1, r), (q_l, q_m, q_r)', (q_l, q_m, r), (l, s'_1, q_r), \rightarrow) &= 1,\\
\delta((l, s'_1, r), (q_l, q_m, q_r)', (q_l, q_m, r)', (l, s'_1, q_r), \rightarrow) &= 1;
\end{aligned}$$

$$\begin{aligned}
\delta((l, s'_1, r), (q_l, q_m, q_r), (q_l, q_m, r)', (q_l, s_2, q_r), \leftarrow) &= 1,\\
\delta((l, s_2, r), (q_l, q_m, q_r), (l, q_m, q_r), (q_l, s_2, r), \leftarrow) &= 1,\\
\delta((l, s_2, r), (q_l, q_m, q_r)', (l, q_m, q_r), (q_l, s'_2, r), \leftarrow) &= 1,\\
\delta((l, s'_2, r), (q_l, q_m, q_r)', (l, q_m, q_r)', (q_l, s'_2, r), \leftarrow) &= 1.
\end{aligned}$$

Not all of the above transitions are really needed for simulation. Some of them are just to make evolution unitary.

The aim of the first phase was to put all the information needed for the main second phase in which transitions of $\mathcal{A}$ defined by $U$ are simulated by transitions of $\mathcal{M}$ according to the rules:

$$\begin{aligned}
\delta((l, s'_2, r), (q_l, q_m, q_r), (q'_l, q'_m, q'_r)', (q_l, s_3, r), \rightarrow) &= U((q_l, q_m, q_r), (q'_l, q'_m, q'_r)),\\
\delta((l, s_3, r), (q_l, q_m, q_r), (q'_l, q'_m, q'_r), (l, s_3, r), \rightarrow) &= U((q_l, q_m, q_r), (q'_l, q'_m, q'_r)),\\
\delta((l, s_3, r), (q_l, q_m, q_r)', (q'_l, q'_m, q'_r)', (l, s_0, r), \leftarrow) &= U((q_l, q_m, q_r), (q'_l, q'_m, q'_r)).
\end{aligned}$$

In the third phase $\mathcal{M}$ just moves its head to come to the cell representing the current leftmost cell of the configuration of $\mathcal{A}$. This is achieved by the transitions:

$$\delta((l, s_0, r), (q_l, q_m, q_r), (q_l, q_m, q_r), (l, s_0, r), \leftarrow) = 1.$$

Finally, let $\delta$ take the value 0 everywhere not defined above. In addition, let $k' = k$ and $\Sigma_a = \{q, q' \mid q \in Q_a\}$.

On the base of Theorem 4.3.12 it is now straightforward to show:

**Lemma 4.3.19** *If $\mathcal{A}$ and $\mathcal{M}$ are defined as above, then if $\mathcal{A}$ is a PQCA, then $\mathcal{M}$ is a QTM.*

In order to formulate the new main simulation result we need to define two mappings: $T : C_{\mathcal{A}} \to C_{\mathcal{M}}$ and $f : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$.

For $a \in C_{\mathcal{A}}$ let $n_l^a$ and $n_r^a$ be the indices of the leftmost and rightmost non-quiescent cells of $a$. Define

$$T(a) = (\lambda, s_0, \lambda)c,$$

where

$$c(i) = \begin{cases} a(i), & \text{if } i \neq n_l^a, i \neq n_r^a; \\ a^*(i), & \text{otherwise,} \end{cases}$$

and let $f(t, |c|) = 4t^2 + 4|c|t - t$ be the number of steps needed by $\mathcal{M}$ to simulate $t$ steps of $\mathcal{A}$. One can now prove (Watrous, 1995):

**Lemma 4.3.20** *For any $t \in \mathbf{N}$ and $c \in C_{\mathcal{A}}$ the probability that $\mathcal{A}$ accepts $c$ after $t$ steps is equal to the probability that $\mathcal{M}$ accepts $T(c)$ after $f(t, |T(c)|)$ steps.*

To summarize:

**Theorem 4.3.21** *To any PQCA $\mathcal{A}$ there exists a QTM simulating $\mathcal{A}$ with quadratic slow-down.*