Quantum Energy–based P Systems

Alberto Leporati, Dario Pescini and Claudio Zandron

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano – Bicocca
Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy

e-mail: {leporati,pescini,zandron}@disco.unimib.it

Abstract

Energy–based P systems have been recently defined as P systems in which the amount of energy manipulated and/or consumed during computations is taken into account. In this paper we propose two quantum versions of energy–based P systems. Both versions are defined just like classical energy–based P systems, but for objects and rules. Objects are represented as pure states in the Hilbert space $\mathbb{C}^d$, whereas the definition of rules differs between the two models. In the former, rules are defined as bijective functions — implemented as unitary operators — which transform the objects from the alphabet. In the latter, rules are defined as generic functions which map the alphabet into itself. Such functions are implemented using a generalization of the Conditional Quantum Control technique, and may yield to non–unitary operators. Finally, we address some problems and outline some directions for future work.

1 Introduction

P systems (also called *membrane systems*) have been introduced in [14] as a new class of distributed and parallel computing devices, inspired by the structure and functioning of living cells. The basic model consists of a hierarchical structure composed by several membranes, embedded into a main membrane called the *skin*. Membranes divide the Euclidean space into *regions*, that contain some *objects* (represented by symbols of an alphabet) and *evolution rules*. Using these rules, the objects may evolve and/or move from a region to a neighboring one. The rules are applied in a nondeterministic and maximally parallel way: all the objects that may evolve are

forced to evolve. A *computation* starts from an initial configuration of the system and terminates when no evolution rule can be applied. The result of a computation is the multiset of objects contained into an *output membrane* or emitted from the skin of the system.

In what follows we assume the reader is already familiar with the basic notions and the terminology underlying P systems. For a systematic introduction, we refer the reader to [15]. The latest information about P systems can be found in [18].

Energy–based P systems have been defined in [16] as P systems in which the amount of energy manipulated and/or consumed during computations is taken into account. A given amount of energy is associated to each object of the system. Moreover, instances of a special symbol e are used to denote free energy units occurring into the regions of the system. These energy units can be used to transform objects, using appropriate rules. The rules are defined according to conservativeness considerations. An object can always be transformed into another object having the same energy. On the other hand, if the transformed object has a different energy then the required (resp., exceeding) free energy units are taken from (resp., released to) the region where the rule is applied. We assume that the application of rules consumes no energy. This means, in particular, that objects can be moved (without altering them) between the regions of the system without energy consumption. A special case of energy–based P systems are *conservative* P systems, where the amount of energy entering the system with the input values is completely returned with the output values at the end of the computation.

Formally, an energy–based P system (of degree $m \geq 1$) is a construct

$$\Pi = (A, \varepsilon, \mu, e, w_1, \ldots, w_m, R_1, \ldots, R_m, i_{\text{in}}, i_{\text{out}})$$

where:

- A is an alphabet; its elements are called *objects*;
- $\varepsilon : A \to \mathbb{N}$ is a linear mapping that associates to each object $a \in A$ the value $\varepsilon(a)$ (also denoted by ε_a), which can be thought of as the "energy value of a". If $\varepsilon(a) = \ell$, we also say that object a *embeds* ℓ units of energy. Precisely, if $A = \{a_1, a_2, \ldots, a_d\}$ then for all $i \in \{1, 2, \ldots, d\}$ it holds $\varepsilon(a_i) = \varepsilon(a_1) + (i-1)\delta$ for an appropriate integer value $\delta > 0$. Hence, the energy values considered in the system are equispaced by the quantity δ. By adding "dummy" symbols into the alphabet (that is, symbols which never appear in the system during the computations), we can always assume $\delta = 1$ without loss of generality;

- μ is a hierarchical membrane structure consisting of m membranes. For the sake of clarity, we will label membranes with mnemonic identifiers which recall their function;

- $e \notin A$ is a special symbol that denotes one *free energy* unit, that is, one unit of energy which is not embedded into any object;

- w_i, for all $i \in \{1, \ldots, m\}$, specify the multiset (over $A \cup \{e\}$) of objects initially present in region i;

- R_i, for all $i \in \{1, \ldots, m\}$, is a finite set of evolution rules over A associated with region i. Only rules of the following types are allowed:

$$ae^k \to (b, p) \ , \qquad a \to (b, p)e^k \ , \qquad e \to (e, p)$$

  where $a, b \in A$, $p \in \{\text{here}, \text{in}(name), \text{out}\}$ and k is a non negative integer;

- i_{in} is an integer between 1 and m and specifies the input membrane of Π;

- i_{out} is an integer between 0 and m and specifies the output membrane of Π. If $i_{\text{out}} = 0$ then the environment is used for the output, that is, the output value is the multiset of objects (over A) emitted from the skin.

A special attention is due to the definition of rules. The meaning of rule $ae^k \to (b, p)$, with $a, b \in A$, $p \in \{\text{here}, \text{in}(name), \text{out}\}$, and k a positive integer number, is the following: the object a, in presence of k free energy units, is allowed to be transformed into object b. If $p = \text{here}$ then the new object b remains in the same region; if $p = \text{out}$ then b exits from the current membrane. Finally, if $p = \text{in}(name)$ then b enters into the membrane labelled with $name$, which must be a child of the current membrane in the membrane hierarchy.

The meaning of rule $a \to (b, p)e^k$, when k is a positive integer number, is analogous. The object a is allowed to be transformed into object b by releasing k units of free energy. As above, the new object b may optionally move one level up or down into the membrane hierarchy. The k free energy units can now be used by another rule to produce "more energetic" objects from "less energetic" ones.

When $k = 0$ the rule $ae^k \to (b, p)$ is written as $a \to (a, p)$, and simply moves (if $p \neq \text{here}$) the object a upward or downward into the membrane

hierarchy, without acquiring nor releasing any free energy unit. Analogously, rules $e \to (e, p)$ simply move (if $p \neq$ here) one unit of free energy upward or downward into the membrane hierarchy.

A further constraint is that each rule must be "conservative", in the sense that the amount of energy occurring on the left side of the rule must be the same as the amount of energy which occurs on the right side.

With a little abuse of notation, when the pair (x, p), with $x \in A \cup \{e\}$ and $p \in \{\text{here}, \text{in}(name), \text{out}\}$, appears into a rule we will write x_p. Also, if $p = \text{in}(name)$ and no confusion arises we will usually write just the name of the membrane. Moreover, instead of writing e^k we will sometimes explicitly write k instances of e. It is also understood that the position of e^k (that is, on the left or on the right of the symbol from A) either into the left or into the right side of a rule is uninfluent. Finally, when the position p of an object which occurs in the right side of a rule is "here" we will omit to write it.

A *configuration* of Π is the tuple $(M_1, \ldots, M_m)$ of multisets (over $A \cup \{e\}$) of objects contained in each region of the system. $(w_1, \ldots, w_m)$ is called the *initial configuration*. For two configurations $(M_1, \ldots, M_m)$, $(M'_1, \ldots, M'_m)$ of Π we write $(M_1, \ldots, M_m) \Rightarrow (M'_1, \ldots, M'_m)$ to denote a *transition* from $(M_1, \ldots, M_m)$ to $(M'_1, \ldots, M'_m)$. The reflexive and transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$. A *final configuration* is a configuration where no rule can be applied.

A *computation* is a sequence of transitions between configurations of Π, starting from the initial configuration. A computation is *successful* if and only if it reaches a final configuration or, in other words, it *halts*. It is understood that the multiset (over A, that is, not considering free energy units) of objects which occur in $w_{i_{\text{in}}}$ are the *input values* for the computation. Analogously, the multiset (over A) of objects occurring in the output membrane (or emitted from the skin if $i_{\text{out}} = 0$) in the final configuration is the *output* of the computation. A non–halting computation produces no output.

Since energy is an additive quantity, it is natural to define the *energy of a multiset* as the sum of the amounts of energy associated to each instance of the objects which occur into the multiset. Analogously, the energy of a configuration is the sum of the amounts of energy associated to each multiset which occurs into the configuration. A *conservative computation* is a computation where each configuration has the same amount of energy. A *conservative energy–based P system* is an energy–based P system that performs only conservative computations.

Energy–based P systems are by no means the first model of P systems which involve energy. We recall in particular [1, 8, 17, 9]. In [12] it is shown

that energy–based P systems are able to simulate the Fredkin gate. By allowing different objects of the alphabet to embed the same amount of energy, in [13] the simulation is extended to reversible Fredkin circuits. Moreover it is shown that the simulating P systems can be made *self–reversible* (that is, able to perform both "forward" and "backward" computations) and conservative. This result shows that (non–uniform families of) energy–based P systems are able to perform universal computations.

2 Quantum versions of energy–based P systems

In this section we propose two quantum versions of energy–based P systems. From now on, quantum energy–based P systems will simply be called *quantum P systems* for short. Both versions are defined just like classical energy–based P systems, but for objects and rules. The objects are represented as pure states in the Hilbert space $\mathbb{C}^d$, whereas the definition of rules differs between the two models. In the former, rules are defined as bijective functions — implemented as unitary operators — which transform the objects from the alphabet. In the latter, rules are defined as generic functions which map the alphabet into itself. Such functions are implemented using a generalization of the Conditional Quantum Control technique, and may yield to non–unitary operators.

Before delving into the details of quantum P systems, let us recall some basic notions of quantum computing. From an abstract point of view a quantum computer can be considered as made up of interacting parts. The elementary units (memory cells) that compose these parts are two–levels quantum systems called *qubits*. A qubit is typically implemented using the energy levels of a two–levels atom, or the two spin states of a spin–$\frac{1}{2}$ atomic nucleus, or a polarization photon. The mathematical description — independent of the practical realization — of a single qubit is based on the two–dimensional complex Hilbert space $\mathbb{C}^2$. The Boolean truth values 0 and 1 are represented in this framework by the unit vectors of the canonical orthonormal basis, called the *computational basis* of $\mathbb{C}^2$:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad\qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Qubits are thus the quantum extension of the classical notion of bit, but whereas bits can only take two different values, 0 and 1, qubits are not confined to their two basis (also *pure*) states, $|0\rangle$ and $|1\rangle$, but can also exist in states which are coherent superpositions such as $\psi = c_0 |0\rangle + c_1 |1\rangle$, where

c_0 and c_1 are complex numbers satisfying the condition $|c_0|^2 + |c_1|^2 = 1$. A qubit in this state is not simply in state $|0\rangle$ or $|1\rangle$, nor is it in an intermediate state; rather the qubit is in both states *simultaneously* and a mere act of measurement alters this state. Indeed, performing a measurement on a qubit in the above superposition will return 0 with probability $|c_0|^2$ and 1 with probability $|c_1|^2$; the state of the qubit after the measurement (*post–measurement state*) will be $|0\rangle$ or $|1\rangle$, depending on the outcome.

Let us stress that in axiomatic quantum mechanics a pure state is described by a one–dimensional subspace of the involved Hilbert space, whose vectors are *representatives* of this state. Thus, two unit vectors $|\psi\rangle$ and $|\varphi\rangle$ describe (belong to) the same state if and only if they differ of a phase factor, that is, if and only if there exists a real value $\vartheta \in [0, 2\pi)$ such that $|\psi\rangle = e^{i\vartheta} |\varphi\rangle$.

A *quantum register* of size n (also called an *n–register*) is mathematically described by the Hilbert space $\otimes^n \mathbb{C}^2 = \underbrace{\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2}_{n \text{ times}}$, representing a set of n qubits labelled by the index $i \in \{1, \ldots, n\}$. An *n–configuration* (also *pattern*) is a vector $|x_1\rangle \otimes \ldots \otimes |x_n\rangle \in \otimes^n \mathbb{C}^2$, usually written as $|x_1, \ldots, x_n\rangle$, considered as a quantum realization of the Boolean tuple $(x_1, \ldots, x_n)$. Let us recall that the dimension of $\otimes^n \mathbb{C}^2$ is 2^n and that $\{|x_1, \ldots, x_n\rangle : x_i \in \{0, 1\}\}$ is an orthonormal basis of this space called the n–register *computational basis*.

Unlike the situation of the classical wired computer, where voltages of a wire go over voltages of another, in quantum computers something different happens. Each qubit of a given n–register is prepared in some particular pure state ($|0\rangle$ or $|1\rangle$) in order to realize the required n–configuration $|x_1, \ldots, x_n\rangle$, quantum realization of an input Boolean tuple of length n. Then, a linear operator $G : \otimes^n \mathbb{C}^2 \to \otimes^n \mathbb{C}^2$ is applied to the n–register. The application of G has the effect of transforming the n–configuration $|x_1, \ldots, x_n\rangle$ into a new n–configuration $G(|x_1, \ldots, x_n\rangle) = |y_1, \ldots, y_n\rangle$, which is the quantum realization of the output tuple of the computer. In other words, G transforms the vectors of the n–register computational basis into vectors of the same basis. Let us stress that in particular such operator G changes the state $|x_i\rangle$ (with $x_i \in \{0, 1\}$) of each qubit of the register into a new state $|y_i\rangle$ (with $y_i \in \{0, 1\}$) of the same qubit, and we interpret such modifications as a computation step performed by the quantum computer.

The action of the operator G on $\Phi = \sum c^{i_1 \ldots i_n} |x_{i_1}, \ldots, x_{i_n}\rangle$, expressed as a linear combination of the elements of the n–register basis, is obtained by linearity: $G(\Phi) = \sum c^{i_1 \ldots i_n} G(|x_{i_1}, \ldots, x_{i_n}\rangle)$. We recall that linear operators which act on n–registers can be represented as order 2^n square matrices

of complex entries. Usually such operators, as well as the corresponding matrices, are required to be unitary. In particular, this implies that the implemented operations are logically reversible (an operation is *logically reversible* if its inputs can always be deduced from its outputs).

All these notions can be easily extended to quantum systems which have $d > 2$ pure states. In this setting, the d–valued versions of qubits are usually called *qudits* [10]. As it happens with qubits, a qudit is typically implemented using the energy levels of an atom or a nuclear spin. The mathematical description — independent of the practical realization — of a single qudit is based on the d–dimensional complex Hilbert space $\mathbb{C}^d$. In particular, the pure states $|0\rangle, \left|\frac{1}{d-1}\right\rangle, \left|\frac{2}{d-1}\right\rangle, \ldots, \left|\frac{d-2}{d-1}\right\rangle, |1\rangle$ are represented by the unit vectors of the canonical orthonormal basis, called the *computational basis* of $\mathbb{C}^d$:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \left|\frac{1}{d-1}\right\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \cdots, \quad \left|\frac{d-2}{d-1}\right\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

As before, a *quantum register* of size n can be defined as a collection of n qudits. It is mathematically described by the Hilbert space $\otimes^n \mathbb{C}^d = \underbrace{\mathbb{C}^d \otimes \ldots \otimes \mathbb{C}^d}_{n \text{ times}}$. An *$n$–configuration* is now a vector $|x_1\rangle \otimes \ldots \otimes |x_n\rangle \in \otimes^n \mathbb{C}^d$, simply written as $|x_1, \ldots, x_n\rangle$, for x_i running on $L_d = \left\{0, \frac{1}{d-1}, \frac{2}{d-1}, \ldots, \frac{d-2}{d-1}, 1\right\}$. An n–configuration can be viewed as the quantum realization of the "classical" tuple $(x_1, \ldots, x_n) \in L_d^n$. The dimension of $\otimes^n \mathbb{C}^d$ is d^n and the set $\{|x_1, \ldots, x_n\rangle : x_i \in L_d\}$ of all n–configurations is an orthonormal basis of this space, called the *n–register computational basis*. Notice that the set L_d can also be interpreted as a set of truth values, where 0 denotes falsity, 1 denotes truth and the other elements indicate different degrees of indefiniteness.

In our definition of quantum P systems, all the elements of the model (multisets, the membrane hierarchy, input and output membrane, configurations, computations, and so on) are defined just like the corresponding elements of a classical energy–based P system, but for objects and rules. The objects are represented by the pure states of a quantum system. Hence, if the alphabet contains $d \geq 2$ elements, then without loss of generality we can put $A = \left\{|0\rangle, \left|\frac{1}{d-1}\right\rangle, \left|\frac{2}{d-1}\right\rangle, \ldots, \left|\frac{d-2}{d-1}\right\rangle, |1\rangle\right\}$, that is, $A = \{|a\rangle : a \in L_d\}$.

In a possible physical realization, we can think of a quantum system which is able to assume the above pure states. As stated above, such system will also be able to assume as a state any superposition of the kind:

$$c_0 \left|0\right\rangle + c_{\frac{1}{d-1}} \left|\frac{1}{d-1}\right\rangle + \ldots + c_{\frac{d-2}{d-1}} \left|\frac{d-2}{d-1}\right\rangle + c_1 \left|1\right\rangle$$

with $c_0, c_{\frac{1}{d-1}}, \ldots, c_{\frac{d-2}{d-1}}, c_1 \in \mathbb{C}$ such that $\sum_{i=0}^{d-1} \left|c_{\frac{i}{d-1}}\right|^2 = 1$. A multiset is simply a collection of quantum systems, each in its own state. In the most general setting, two or more quantum systems may become *entangled*, either because they are prepared in this way as input values for a computation, or because they are the result of the application of an operator on them. When two or more quantum systems are entangled, the state of each single system is tied to the state of the other systems. So, if we perform a measurement on a single system of an entangled pair, such operation will affect also the state of the other system. Formally, two or more quantum systems are entangled if their global state cannot be factorized as the tensor product of the states of the single systems. For example, $\frac{1}{\sqrt{2}}\left(\left|10\right\rangle - \left|01\right\rangle\right)$ is an entangled quantum state of the Hilbert space $\mathbb{C}^2 \otimes \mathbb{C}^2$.

Now let us turn to rules. As stated above, in this paper we propose two versions of quantum P systems. In the first version, rules are simply defined as unitary operators which transform the state of their input qudits. This is analogous to what happens with reversible logic gates, which act according to their truth table. If a rule acts on n quantum systems, we say that it computes an (n, d)–function, that is, a function $f : A^n \to A^n$. Notice that, since rules are defined by unitary operators, they are logically reversible. This means that f is a permutation on the set A^n.

As an example let us assume $d = 3$, so that $A = \left\{\left|0\right\rangle, \left|\frac{1}{2}\right\rangle, \left|1\right\rangle\right\}$. If we want to write the unitary operator which realizes the function $f : A \to A$ such that:

$$f(\left|0\right\rangle) = \left|\frac{1}{2}\right\rangle, \qquad f\left(\left|\frac{1}{2}\right\rangle\right) = \left|0\right\rangle, \qquad f(\left|1\right\rangle) = \left|1\right\rangle$$

we can operate as follows. The unitary matrix is an order $d^n = 3^1 = 3$ square matrix having complex entries. Each row and each column is associated with an element of A^n (in this case, A). If $f(\left|x_1, \ldots, x_n\right\rangle) = \left|y_1, \ldots, y_n\right\rangle$, then the element of the matrix whose row and column is associated with $\left|x_1, \ldots, x_n\right\rangle$ and $\left|y_1, \ldots, y_n\right\rangle$ respectively, is put to 1. All the other elements in the same row are put to 0. Since f is bijective, also all the other elements in the

same column are put to 0. Continuing with the example, the unitary matrix which corresponds to the function $f : A \to A$ described above is:

$$U_f = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Of course there exist also *genuine quantum functions*, i.e., functions that have no classical counterpart and are thus characterized by the fact that some input pure states (tensor product of vectors of the computational basis of $\mathbb{C}^d$) are transformed into non–trivial superpositions of pure states. An example of an operation of this kind is the $\sqrt{\text{NOT}}$ function, acting on the states of a single qubit, that can be thought of as the realization of a 1–register. Another genuine quantum gate is the *Hadamard* gate, also acting on quantum registers of size 1. Formally, the map $H : \mathbb{C}^2 \to \mathbb{C}^2$ is described by the following order 2 unitary matrix:

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

and the corresponding action on qubits is given by:

$$\begin{cases} H\left|0\right\rangle = \frac{1}{\sqrt{2}}\left|0\right\rangle + \frac{1}{\sqrt{2}}\left|1\right\rangle \\ H\left|1\right\rangle = \frac{1}{\sqrt{2}}\left|0\right\rangle - \frac{1}{\sqrt{2}}\left|1\right\rangle \end{cases}$$

Our second proposal for quantum P systems is more complicated, but is in some sense closer to physics laws. Objects are defined as in the first proposal, whereas the definition of rules is directly inspired from energy–based P systems. However, as we will see, differently from a classical energy–based P system, in a quantum P system free energy units are not just symbols which move into the system and cooperate with objects when the rules are applied. In quantum P systems, free energy units are true quanta of energy (for example, photons) which are necessary to transform a "low" energy state of a system to a higher energy state. On the other hand, when a quantum system decays to a lower energy state, it is understood that the difference of energy between the two states is released in the form of energy quanta (e.g., photons).

In order to become more formal, let us consider the set $\mathcal{E}_d = \left\{\varepsilon_0, \varepsilon_{\frac{1}{d-1}}, \varepsilon_{\frac{2}{d-1}}, \ldots, \varepsilon_{\frac{d-2}{d-1}}, \varepsilon_1\right\} \subseteq \mathbb{R}$ of real values; we can think to such quantities as energy values. To each element $v \in L_d$ (and hence to each object $|v\rangle \in A$)

we associate the energy level ε_v; moreover, let us assume that the values of $\mathcal{E}_d$ are all positive, equispaced, and ordered according to the corresponding objects: $0 < \varepsilon_0 < \varepsilon_{\frac{1}{d-1}} < \cdots < \varepsilon_{\frac{d-2}{d-1}} < \varepsilon_1$. If we denote by $\Delta\varepsilon$ the gap between two adjacent energy levels then the following linear relation holds:

$$\varepsilon_v = \varepsilon_0 + \Delta\varepsilon\,(d-1)\,v \qquad\qquad \forall\, v \in L_d \tag{1}$$

Notice that it is not required that $\varepsilon_0 = \Delta\varepsilon$.

If $\underline{x} = |x_1, \ldots, x_n\rangle \in A^n$ is an n–configuration, we define the *amount of energy associated to* $\underline{x}$ as $E_n(\underline{x}) = \sum_{i=1}^{n} \varepsilon_{x_i}$, where $\varepsilon_{x_i} \in \mathcal{E}_d$ is the amount of energy associated to the i–th element $|x_i\rangle$ of the configuration. Let us remark that the map $E_n : L_d^n \to \mathbb{R}^+$ is indeed a family of mappings parameterized by n, the size of the input. We say that a rule is *conservative* if, for any input configuration $\underline{x} = |x_1, \ldots, x_n\rangle \in A^n$, the corresponding output configuration $\underline{y} = |y_1, \ldots, y_m\rangle \in A^m$ is such that $E_n(\underline{x}) = E_m(\underline{y})$. Rules are again defined as (n, d)–functions, that is, functions of the kind $f : A^n \to A^n$. The difference with respect to the first proposal is that such functions are not necessarily bijections on A^n: they can be arbitrary mappings. This means that the linear operators which realize such functions are not necessarily unitary. Hence, we need a method to build and describe them.

In this paper we present a quantum realization of rules using an extension of the *Conditional Quantum Control* technique introduced in [2]. The technique is used to write quantum operators which describe the behavior of classical rules. Such operators are sums of "local" operators, each of which is a tensor product of suitable compositions of the operators $a^\dagger$ and a, which are the finite dimensional versions of creation and annihilation operators usually found in quantum mechanics. An equivalent formulation is also given, using spin–rising (J_+) and spin–lowering (J_-) operators.

In the following sections we interpret the d energy levels of a quantum system by the truncated quantum harmonic oscillator. Moreover, we introduce the creation and annihilation operators on $\mathbb{C}^d$, and we show how they can be used to transform the state of a single quantum system, as required by the rules which occur in energy–based P systems. An alternative description is also given, using spin–rising and spin–lowering operators. Finally, we show how the linear operators which correspond to rules can be built, using both a "brute force" approach and an extension of the Conditional Quantum Control.

3 A mathematical model for quantum rules

3.1 The d–levels single system Hamiltonian

In describing a computation device it is important, from the point of view of quantum mechanics, to give the Hamiltonian operator for the physical system that constitutes the computing machinery. As it is well known, the Hamiltonian operator describes the energy of the quantum system and allows one to derive its time evolution.

The quantum realization of d–valued one–input/one–output rules can be done by considering single quantum systems whose Hamiltonian on $\mathbb{C}^d$ is:

$$H = \begin{bmatrix} \varepsilon_0 & 0 & \dots & 0 \\ 0 & \varepsilon_0 + \Delta\varepsilon & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varepsilon_0 + (d-1)\Delta\varepsilon \end{bmatrix} \tag{2}$$

The energy eigenvalues $\varepsilon_k = \varepsilon_0 + k\Delta\varepsilon$ of H, starting from the ground energy state ε_0 and equispaced by the quantum of energy $\Delta\varepsilon$, are the ones of the infinite dimensional quantum harmonic oscillator truncated at the $(d-1)$-th excited level (see Figure 1).

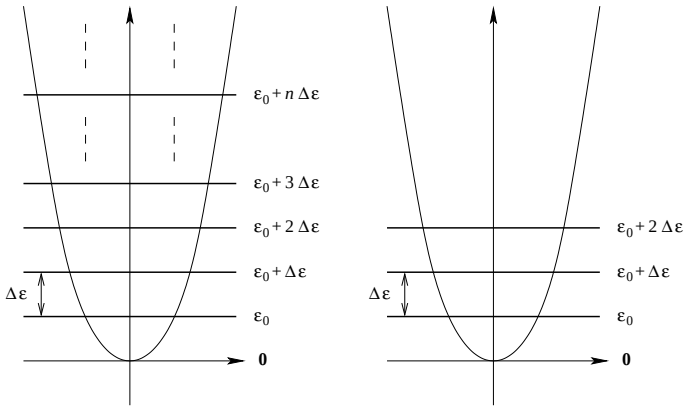

Figure 1: Energy levels of the infinite dimensional (on the left) and of the truncated (on the right) quantum harmonic oscillator

The unit vector $|H = \varepsilon_k\rangle = \left|\frac{k}{d-1}\right\rangle$, for $k \in \{0, 1, \dots, d-1\}$, is the eigenvector of the state of energy $\varepsilon_0 + k\Delta\varepsilon$. The spectral resolution of

the above truncated harmonic oscillator Hamiltonian (2) is:

$$H = \sum_{k=0}^{d-1} (\varepsilon_0 + k\Delta\varepsilon) P_{\varepsilon_k}$$

where each orthogonal projection $P_{\varepsilon_k} = P_{\frac{k}{d-1}}$ is the quantum realization of the sharp event "a measure of the system energy yields the value $\varepsilon_0 + k\Delta\varepsilon$".

We can now introduce the creation and annihilation operators on the d–dimensional Hilbert spaces $\mathbb{C}^d$. Formally, *creation* and *annihilation* operators on the Hilbert space $\mathbb{C}^d$ are respectively defined as:

$$a^\dagger = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & \sqrt{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{d-1} & 0 \end{bmatrix} \qquad a = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \sqrt{2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sqrt{d-1} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

The operators $a^\dagger$ and a are non–Hermitian, adjoints of each other, and satisfy the following commutation and anticommutation relations, respectively:

$$[a, a^\dagger] = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1-d \end{bmatrix} \qquad [a, a^\dagger]_+ = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d-3 & 0 \\ 0 & 0 & \cdots & 0 & d-1 \end{bmatrix}$$

Thus, if one excludes the case $d = 2$ where the boson anticommutation rule is satisfied, neither the fermion commutation rule $[a, a^\dagger] = \mathrm{Id}$ nor the anticummutation rule $[a, a^\dagger]_+ = \mathrm{Id}$ of the infinite dimensional case hold.

From these formulas it follows that the action of $a^\dagger$ on the vectors of the canonical orthonormal basis of $\mathbb{C}^d$ is the following:

$$a^\dagger \left| \frac{k}{d-1} \right\rangle = \sqrt{k+1} \left| \frac{k+1}{d-1} \right\rangle \qquad \text{for } k \in \{0, 1, \ldots, d-2\}$$
$$a^\dagger |1\rangle = \mathbf{0}$$

whereas the action of a is:

$$a \left| \frac{k}{d-1} \right\rangle = \sqrt{k} \left| \frac{k-1}{d-1} \right\rangle \qquad \text{for } k \in \{1, 2, \ldots, d-1\}$$
$$a |0\rangle = \mathbf{0}$$

Using $a^\dagger$ and a we can introduce the following operators:

$$N = a^\dagger a = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d-1 \end{bmatrix} \qquad aa^\dagger = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & d-1 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

The eigenvalues of the self–adjoint operator N are $0, 1, 2, \ldots, d-1$, and the eigenvector corresponding to the generic eigenvalue k is $|N = k\rangle = \left|\frac{k}{d-1}\right\rangle$. This corresponds to the notation adopted in [10], where the qudit base states are denoted by $|0\rangle, |1\rangle, \ldots, |d-1\rangle$, and it is assumed that a qudit can be in a superposition of the d base states:

$$c_0 |0\rangle + c_1 |1\rangle + \ldots + c_{d-1} |d-1\rangle$$

with $c_i \in \mathbb{C}$ for $i \in \{0, 1, \ldots, d-1\}$, and $|c_0|^2 + |c_1|^2 + \ldots + |c_{d-1}|^2 = 1$.

One possible physical interpretation of N is that it describes the *number of particles* of physical systems consisting of a maximum number of $d-1$ particles. In order to add a particle to the k particles state $|N = k\rangle$ (thus making it switch to the "next" state $|N = k+1\rangle$) we apply the creation operator $a^\dagger$, while to remove a particle from this system (thus making it switch to the "previous" state $|N = k-1\rangle$) we apply the annihilation operator a. Since the maximum number of particles that can be simultaneously in the system is $d-1$, the application of the creation operator to a full $d-1$ particles system does not have any effect on the system, and returns as a result the null vector. Analogously, the application of the annihilation operator to an empty particle system does not affect the system and returns the null vector as a result.

Another physical interpretation of operators $a^\dagger$ and a, by operator N, follows from the possibility of expressing the Hamiltonian (2) as follows:

$$H = \varepsilon_0 \,\mathrm{Id} + \Delta\varepsilon \,\mathrm{N} = \varepsilon_0 \,\mathrm{Id} + \Delta\varepsilon \,\mathrm{a}^\dagger\mathrm{a}$$

In this case $a^\dagger$ (resp., a) realizes the transition from the eigenstate of energy $\varepsilon_k = \varepsilon_0 + k\,\Delta\varepsilon$ to the "next" (resp., "previous") eigenstate of energy $\varepsilon_{k+1} = \varepsilon_0 + (k+1)\,\Delta\varepsilon$ (resp., $\varepsilon_{k-1} = \varepsilon_0 + (k-1)\,\Delta\varepsilon$) for any $0 \leq k < d-1$ (resp., $0 < k \leq d-1$), while it collapses the last excited (resp., ground) state of energy $\varepsilon_0 + (d-1)\,\Delta\varepsilon$ (resp., ε_0) to the null vector.

The collection of all linear operators on $\mathbb{C}^d$ is a d^2–dimensional linear space whose canonical basis is:

$$\{E_{x,y} = |y\rangle \langle x| \ : \ x, y \in L_d\}$$

Since $E_{x,y}\,|x\rangle = |y\rangle$ and $E_{x,y}\,|z\rangle = \mathbf{0}$ for every $z \in L_d$ such that $z \neq x$, this operator transforms the unit vector $|x\rangle$ into the unit vector $|y\rangle$, collapsing all the other vectors of the canonical orthonormal basis of $\mathbb{C}^d$ into the null vector. For $i, j \in \{0, 1, \ldots, d-1\}$, the operator $E_{\frac{i}{d-1},\frac{j}{d-1}}$ can be represented as an order d square matrix having 1 in position $(j+1, i+1)$ and 0 in every other position:

$$E_{\frac{i}{d-1},\frac{j}{d-1}} = (\delta_{r,j+1}\delta_{i+1,s})_{r,s=1,2,\ldots,d}$$

Each of the operators $E_{x,y}$ can be expressed, using the whole algebraic structure of the associative algebra of operators, as a suitable composition of creation and annihilation operators.

We can use the whole algebraic structure (in particular, the composition operation) of the associative algebra of operators to express any such operator (i.e., any order d complex matrix) as a linear combination of suitable compositions of creations and annihilations. Precisely, if we denote by $A^{p,q,r}_{u,v}$ the expression

$$\underbrace{v \cdots v}_{r}\underbrace{v^* \cdots v^*}_{q}\underbrace{v \cdots v}_{p}u \tag{3}$$

where $u, v \in \{a^\dagger, a\}$, v^* is the adjoint of v, and p, q, r are non negative integer values, then for any $i, j \in \{0, 1, \ldots, d-1\}$ we can express the operator $E_{\frac{i}{d-1},\frac{j}{d-1}}$ in terms of creation and annihilation as follows:

$$E_{\frac{i}{d-1},\frac{j}{d-1}} = \begin{cases} \frac{\sqrt{j!}}{(d-1)!}A^{d-2,d-1-j,0}_{a^\dagger,a^\dagger} & \text{if } i = 0 \\ \frac{\sqrt{j!}}{(d-1)!}A^{d-1,d-1-j,0}_{a,a^\dagger} & \text{if } i = 1 \text{ and } j \geq 1 \\ \frac{\sqrt{i!}}{(d-1)!\sqrt{j!}}A^{d-2-i,d-1,j}_{a^\dagger,a^\dagger} & \text{if } (i = 1,\ j = 0 \text{ and } d \geq 3) \text{ or} \\ & \quad (1 < i < d-2 \text{ and } j \leq i) \\ \frac{\sqrt{j!}}{(d-1)!\sqrt{i!}}A^{i-1,d-1,d-1-j}_{a,a} & \text{if } (i = d-2,\ j = d-1 \text{ and } d \geq 3) \\ & \quad \text{or } (1 < i < d-2 \text{ and } j > i) \\ \frac{1}{\sqrt{(d-1)!j!(d-1)}}A^{d-1,j,0}_{a^\dagger,a} & \text{if } i = d-2 \text{ and } j \leq d-2 \\ \frac{1}{\sqrt{(d-1)!j!}}A^{d-2,j,0}_{a,a} & \text{if } i = d-1 \end{cases}$$

3.2 The angular momentum interpretation of qudits

As it is well known, for a fixed integer $d \geq 2$ the angular momentum based on the Hilbert space $\mathbb{C}^d$ consists of the triple of self–adjoint operators $\mathbf{J} = (J_x, J_y, J_z)$. Moreover, for $j = \frac{d-1}{2}$, the real value $j(j+1)$ is an eigenvalue of the operator $\mathbf{J}^2 = J_x^2 + J_y^2 + J_z^2$. The matrix representation of the z

component of this angular momentum with respect to the orthonormal basis of its eigenvectors is:

$$J_z = \begin{bmatrix} \frac{d-1}{2} & 0 & \dots & 0 & 0 \\ 0 & \frac{d-3}{2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{3-d}{2} & 0 \\ 0 & 0 & \dots & 0 & \frac{1-d}{2} \end{bmatrix}$$

Thus, the z component of the angular momentum can assume d possible eigenvalues:

$$m = \frac{d-(2k+1)}{2} \qquad \text{for } k \in \{0, 1, \dots, d-1\}$$

with corresponding eigenvectors:

$$\left| J_z = \frac{d-(2k+1)}{2} \right\rangle = \left| \frac{k}{d-1} \right\rangle \tag{4}$$

Let us consider the two operators J_+ and J_- on the Hilbert space $\mathbb{C}^d$ which are obtained from the general angular momentum operators as:

$$J_+ = J_x + iJ_y \qquad\qquad J_- = J_x - iJ_y$$

The operators J_+ and J_- are non–Hermitian, adjoints of each other, and satisfy the canonical commutation relation $[J_+, J_-] = 2J_z$. In matrix form they can be expressed as follows:

$$J_+ = \begin{bmatrix} 0 & \sqrt{d-1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sqrt{2(d-2)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sqrt{2(d-2)} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sqrt{d-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

and

$$J_- = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 \\ \sqrt{d-1} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \sqrt{2(d-2)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{2(d-2)} & 0 & 0 \\ 0 & 0 & \cdots & 0 & \sqrt{d-1} & 0 \end{bmatrix}$$

That is, for $r, s \in \{1, 2, \ldots, d\}$, the element in position (r, s) of matrices J_+ and J_- is, respectively:

$$(J_+)_{r,s} = \sqrt{r(d-r)}\delta_{r,s-1}$$
$$(J_-)_{r,s} = \sqrt{s(d-s)}\delta_{r,s+1}$$

As it is well known, the action of operators J_+ and J_- on the vectors of the orthonormal basis of $\mathbb{C}^d$ formed by the eigenvectors of J_z is the following:

$$J_+ |J_z = m\rangle = \sqrt{j(j+1) - m(m+1)}\, |J_z = m+1\rangle \quad \text{for } m = -j, \ldots, j$$

and

$$J_- |J_z = m\rangle = \sqrt{j(j+1) - m(m-1)}\, |J_z = m-1\rangle \quad \text{for } m = -j, \ldots, j$$

Thus, we can interpret these operators as follows: the application of J_+ has the effect of changing the z component of the angular momentum to the next value. If applied to a system which has already a maximum value of J_z, J_+ leaves the system unchanged and returns as a result the null vector. Analogously, the application of J_- has the effect of switching the system to the previous value of the z component of the angular momentum. If applied to a system which has already a minimum value of J_z, J_- does not affect the system and returns as a result the null vector. Usually, J_+ and J_- are called the *spin–rising* and the *spin–lowering* operators, respectively.

The actions of J_+ and J_- on the vectors of the qudit orthonormal basis are the following:

$$J_+ \left| \frac{k}{d-1} \right\rangle = \sqrt{k(d-k)} \left| \frac{k-1}{d-1} \right\rangle \qquad \text{for } k \in \{1, 2 \ldots, d-1\}$$
$$J_+ |0\rangle = \mathbf{0}$$

and

$$J_- \left| \frac{k}{d-1} \right\rangle = \sqrt{(k+1)(d-(k+1))} \left| \frac{k+1}{d-1} \right\rangle \quad \text{for } k \in \{0, 1, \ldots, d-2\}$$
$$J_- |1\rangle = \mathbf{0}$$

In particular, let us note that J_+ switches a qudit to the *previous* element in L_d, whereas J_- switches it to the *next* element. The effect of operator J_+ is depicted on the left side of Figure 2 for a spin–1 system on the Hilbert space $\mathbb{C}^3$. On the right side of the same figure the annihilation action of

the same operator on a three–levels system is given for comparison with the previous behavior. A similar figure with respect to J_- can be drawn showing its spin–1 annihilation action with respect to the eigenstate creation behavior.

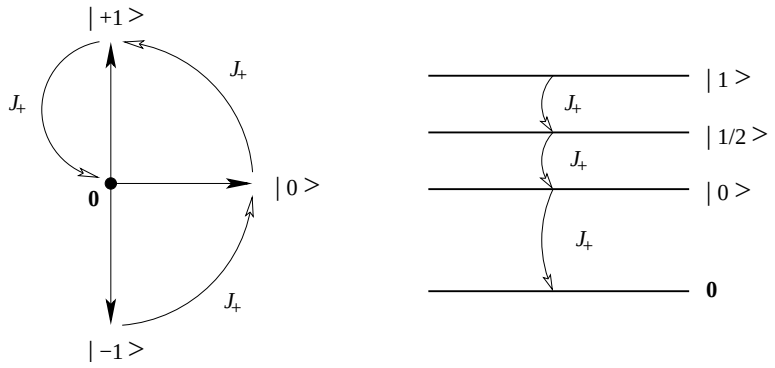

Figure 2: The effect of the spin–rising operator on a spin–1 system and the corresponding annihilation on three–level eigenstates

Let us note also that in the Boolean case (that is, when $d = 2$) it holds:

$$a^\dagger = J_- = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad \text{and} \qquad a = J_+ = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Therefore it holds also $N = J_- J_+$ and $N' = J_+ J_-$, whereas in general, for $d > 2$, such equalities do not hold.

We conclude this section by presenting the expressions that allow one to obtain the operators $E_{\frac{i}{d-1},\frac{j}{d-1}}$ in terms of spin–rising and spin–lowering. Let us consider the formal expression (3) applied to $u, v \in \{J_+, J_-\}$; moreover, let:

$$c_{r,s} = \frac{\prod_{k=r}^{s} \sqrt{k(d-k)}}{\prod_{k=1}^{d-1} k(d-k)}$$

where s, r are two non negative integers. Then, for $i, j \in \{0, 1, \ldots, d-1\}$ it

holds:

$$E_{\frac{i}{d-1},\frac{j}{d-1}} = \begin{cases} c_{1,j}\ A^{d-2,d-1-j,0}_{J_-,J_-} & \text{if } i = 0 \\ c_{2,j}\ A^{d-1,d-1-j,0}_{J_+,J_-} & \text{if } i = 1 \text{ and } j \geq 1 \\ c_{j+1,i}\ A^{d-2-i,d-1,j}_{J_-,J_-} & \text{if } (i = 1,\ j = 0 \text{ and } d \geq 3) \text{ or} \\ & \quad (1 < i < d-2 \text{ and } j \leq i) \\ c_{i+1,j}\ A^{i-1,d-1,d-1-j}_{J_+,J_+} & \text{if } (i = d-2,\ j = d-1 \text{ and } d \geq 3) \text{ or} \\ & \quad (1 < i < d-2 \text{ and } j > i) \\ c_{2,d-1-j}\ A^{d-1,j,0}_{J_-,J_+} & \text{if } i = d-2 \text{ and } j \leq d-2 \\ c_{1,d-1-j}\ A^{d-2,j,0}_{J_+,J_+} & \text{if } i = d-1 \end{cases}$$

4 Quantum realization of rules

Now that we have a mathematical model to interpret objects as vectors of the Hilbert space $\mathbb{C}^d$, and the quantum version of rules as linear operators $G : \otimes^n \mathbb{C}^d \to \otimes^n \mathbb{C}^d$ which implement (n,d)–functions, let us address the following problem.

Problem 1 *Given the truth table of an (n,d)–function $f : L_d^n \to L_d^n$, describe the linear operator $G_f : \otimes^n \mathbb{C}^d \to \otimes^n \mathbb{C}^d$ that provides a quantum realization of the rule as a formula containing only the linear operators* Id_2, $\mathrm{a}^\dagger$, a *and the algebraic operations* $+$, $-$, $\circ$, $\otimes$. ∎

If the (n,d)–function f is reversible then we already know how to build the corresponding (unitary) operator G_f. With the techniques we will introduce, it is also immediate to write a formula which describes G_f. However, we are interested to give a quantum description of *all* possible (n,d)–functions and thus, as we will see in the following, generally the operators we will obtain are not necessarily unitary.

In the next sections we expose two methods that can be used to build any (n,d)–function: the so called "brute force" method, and an extension of the Conditional Quantum Control method, originally proposed by Barenco, Deutsch, Ekert and Jozsa in [2].

4.1 The "brute force" method

We can write the global operator G_f as a sum of so called *local operators*, by a "brute force" procedure, where each local operator corresponds to a single row of the table which describes the (n,d)–function. Precisely, in order to

translate the generic table row:

$$(x_1, x_2, \ldots, x_n) \;\mapsto\; (y_1, y_2, \ldots, y_n)$$

meaning that the input n–tuple $(x_1, x_2, \ldots, x_n)$ is transformed by the function into the output n–tuple $(y_1, y_2, \ldots, y_n)$, we build the "local" operator:

$$E_{x_1,y_1} \otimes E_{x_2,y_2} \otimes \cdots \otimes E_{x_n,y_n}$$

where $E_{x,y} := |y\rangle \langle x|$, with $x, y \in L_d$, is the operator that transforms the single qudit vector $|x\rangle$ into the vector $|y\rangle$, and returns the null vector if it is applied to any other vector of the computational basis of $\mathbb{C}^d$.

Hence, the operator $E_{x_1,y_1} \otimes E_{x_2,y_2} \otimes \cdots \otimes E_{x_n,y_n}$ transforms the input vector $|x_1, x_2, \ldots, x_n\rangle \in \otimes^n \mathbb{C}^d$ into the output vector $|y_1, y_2, \ldots, y_n\rangle \in \otimes^n \mathbb{C}^d$, whereas it collapses all the other input vectors of the n–register computational basis to the null vector.

Since each operator $E_{x,y}$ can be expressed as an appropriate composition of creation and annihilation (resp., spin–rising and spin–lowering) operators, we can conclude that every local operator is a tensor product of suitable compositions of creation and annihilation (resp., spin–rising and spin–lowering) operators.

4.2 The generalized "Conditional Quantum Control" method

Let us now introduce a method derived from *Conditional Quantum Control* [2].

The quantum realization of a "controlled behavior" can be obtained by making use of the operators $P_X = E_{X,X} = |X\rangle \langle X|$, for $X \in \left\{0, \frac{1}{d-1}, \frac{2}{d-1}, \ldots, \frac{d-2}{d-1}, 1\right\}$. For simplicity, let us first consider the case of a $(2,2)$–function. For a reason that will be clear in a moment, we call *control qubit* and *target qubit* the first and the second input, respectively. If we want to realize a linear operator performing the condition: "if the control qubit is $|1\rangle$ then the operator O_1 is applied to the target qubit (and the control qubit is left unchanged)", then we can build the operator $N \otimes O_1$, where $N = E_{1,1} = |1\rangle \langle 1|$ checks for the condition "the control qubit is $|1\rangle$" and O_1 is the operator which acts on the target qubit $|x_2\rangle$. Note that if the control qubit is $|0\rangle$ then the operator $N \otimes O_1$ produces the null vector of $\mathbb{C}^2 \otimes \mathbb{C}^2$. Similarly, $N' \otimes O_0$, with $N' = E_{0,0} = |0\rangle \langle 0|$ realizes the condition "if the control qubit is $|0\rangle$ then the operator O_0 is applied to the target qubit $|x_2\rangle$ (and the control qubit is left unchanged)".

Notice that the method we are using here is a generalization of the Conditional Quantum Control method introduced in [2]. In fact recall that in Conditional Control $(n, 2)$–functions, 2^k functions $\delta_0, \ldots, \delta_{2^k-1}$ are stored in the memory of the control unit, the function δ_a being bijectively associated to the control input configuration labelled by the integer number $a \in \{0, \ldots, 2^k - 1\}$ (see Figure 3). In [2] these functions are described through unitary operators $U_0, U_1, \ldots, U_{2^k-1}$, defined on the Hilbert space $\otimes^{n-k}\mathbb{C}^2$; here we drop the requirement that such operators, as well as the global operator defined on $\otimes^n\mathbb{C}^2$, be unitary. Moreover, in the following we apply this method to realize d–valued operators.

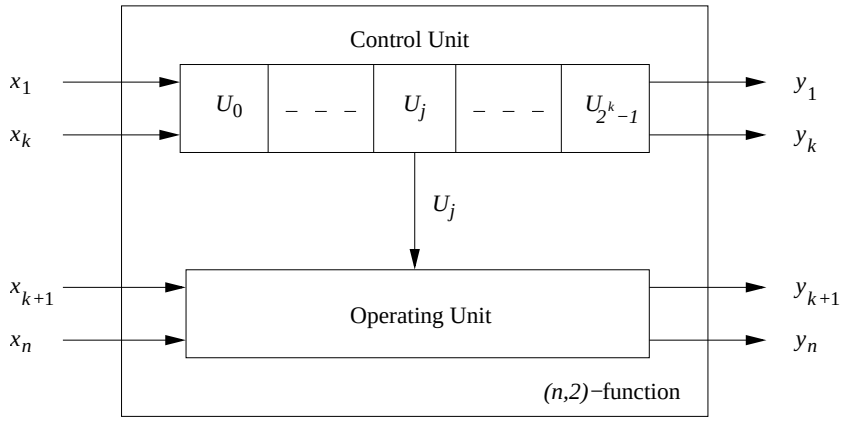

Figure 3: Conditional Quantum Control: the function is divided into a *control unit* and an *operating unit*. The input values of the control unit select a prescribed operator to be applied to the input values of the operating unit

Thus, when a configuration $|x_1, \ldots, x_k\rangle$ is fed to the control unit of a Conditional Control function two things happen:

1. the control configuration $|x_1, \ldots, x_k\rangle$ is returned unchanged into the output values of the control unit, and

2. the (non necessarily unitary) operator U_a bijectively associated to the control configuration is selected and applied to the input configuration $|x_{k+1}, \ldots, x_n\rangle$ of the operating unit, producing the output configuration $U_a |x_{k+1}, \ldots, x_n\rangle$.

The global operator on $[\otimes^k\mathbb{C}^2] \otimes [\otimes^{n-k}\mathbb{C}^2]$ which describes the behavior of

the function can thus be written as:

$$P_0 \otimes U_0 + P_1 \otimes U_1 + \ldots + P_{2^k-1} \otimes U_{2^k-1} \tag{5}$$

where $P_a = E_{a,a} = |a\rangle \langle a|$ is the orthogonal projection of the Hilbert space $\otimes^k \mathbb{C}^2$ which selects the a-th control configuration and collapses to the null vector all the other control configurations, and U_a is the corresponding operator on $\otimes^{n-k}\mathbb{C}^2$ which has to be applied to the target configuration. Making use of Dirac notation, expression (5) can be equivalently written as (see [2]):

$$|0\rangle \langle 0| \otimes U_0 + |1\rangle \langle 1| \otimes U_1 + \ldots + |2^k - 1\rangle\langle 2^k - 1| \otimes U_{2^k-1} \tag{6}$$

A further extension of the Conditional Quantum Control method to the d–valued case is immediate. If the (n, d)–function under consideration can be divided as a k–input/k–output *control unit* and an $(n-k)$–input/$(n-k)$–output *target* (also *operating*) *unit*, then any input configuration $|x_1, \ldots, x_k, x_{k+1}, \ldots, x_n\rangle$ can be splitted into a *control configuration* $|x_1, \ldots, x_k\rangle$ and a *target configuration* $|x_{k+1}, \ldots, x_n\rangle$. The control configuration is returned unchanged on the k output values of the control unit; as a side effect, it selects one of the d^k (non necessarily unitary) operators $U_0, U_1, \ldots, U_{d^k-1}$, defined on the Hilbert space $\otimes^{n-k}\mathbb{C}^d$, stored into the control unit. The selected operator is applied to the target configuration in order to produce the output values of the target unit. The global operator that describes the behavior of the (n, d)–function has now the form:

$$P_0 \otimes U_0 + P_1 \otimes U_1 + \ldots + P_{d^k-1} \otimes U_{d^k-1} = \sum_{X=0}^{d^k-1} P_X \otimes U_X$$

where $P_X = E_{X,X} = |X\rangle \langle X|$ is the orthogonal projection of the Hilbert space $\otimes^k \mathbb{C}^d$ which selects the X-th control configuration, and collapses to the null vector all the other configurations. If many of the operators U_i are identical then this expression is much shorter than the one obtained with the brute force method. On the other hand, it is clear that the method derived from Conditional Quantum Control cannot be used to describe every conceivable (n, d)–function, since there are functions which cannot be divided into a control unit and an operating unit.

5 Some problems and directions for future work

In this section we address some problems we have encountered while trying to define quantum P systems. Since this is a work in progress, still in its

early stage, we would like to share these problems with the community. We hope in this way to generate stimulating discussions on appropriate ways to define quantum P systems.

A first problem which comes to mind when speaking about quantum systems concerns the localization of objects. How can we be sure that an object will stay for a long time into a region surrounded by a membrane? Indeed, one notable feature of quantum systems is the so called "tunnel effect", thanks to which in every moment we have a positive probability that the object spontaneously (i.e., without the application of any rule) leaves the current region. How should we manage this situation? How can we control the computation (that is, the behavior of the system) under the assumption that every object can be anywhere with a non–zero probability?

The above problem is exacerbated by the fact that in classical P systems the objects can be moved as the effect of the application of a rule. Precisely, in classical energy–based P systems we can have a rule of the kind:

$$ae^k \rightarrow (b, p)$$

where $p \in \{\text{here}, \text{in}(name), \text{out}\}$ denotes the position of the new object b. This means that the rule has to transform the object a into the object b, using k units of free energy, and move b according to the prescribed position. In our current definition of quantum energy–based P systems we have only addressed the transformation of a into b. The object a will be represented as a pure state $\left|\frac{\ell}{d-1}\right\rangle$ of $\mathbb{C}^d$, with $\ell \in \{0, 1, \ldots, d-1-k\}$. Analogously, the object b will be represented as the pure state $\left|\frac{\ell+k}{d-1}\right\rangle$. The presence of k free energy units makes the transition from $\left|\frac{\ell}{d-1}\right\rangle$ to $\left|\frac{\ell+k}{d-1}\right\rangle$ possible. Currently, the movement of $\left|\frac{\ell+k}{d-1}\right\rangle$ to the new position p is assumed to occur in the same way as in classical P systems. However, this would imply the existence of a "magic" transportation mechanism that, notwithstanding the tunnel effect, is able to pick up and move a quantum system exactly as desired.

Another problem in the definition of a quantum P system derives from the fact that the presence of k units of free energy enables the transition from any state $\left|\frac{\ell}{d-1}\right\rangle$ of $\mathbb{C}^d$, with $\ell \in \{0, 1, \ldots, d-1-k\}$, to the state $\left|\frac{\ell+k}{d-1}\right\rangle$. Indeed, it is tempting to translate the classical rule $ae^k \rightarrow (b, p)$ into the quantum rule:

$$\left|\frac{\ell}{d-1}\right\rangle e^k \rightarrow \left(\left|\frac{\ell+k}{d-1}\right\rangle, p\right) \tag{7}$$

Now assume that a given region contains two rules of this kind, possibly

with different values of ℓ and k. The presence of a big number of free energy units (at least as many as the maximum of the two values of k) activates both rules. This occurs even if $k = 1$ for both rules, and the region contains one free energy unit. However, in this last case one rule or the other is correctly applied in a nondeterministic way, as it happens in classical P systems. When $k > 1$ for both rules, and the region contains some free energy units, we must avoid the undesirable situation in which some free energy units modify the first object and the remaining free energy units modify the second object. This occurs because when we have written the quantum rule (7) we have implicitly assumed that the k units of free energy must act *simultaneously*.

One possible solution to this problem would be to allow only transitions which involve just one unit of free energy. However, we should check whether these systems are computationally equivalent to the more general ones (we conjecture that the answer is affirmative).

The solution we have adopted in this paper involves the linear operators $E_{x,y} = |y\rangle \langle x|$, with $x, y \in L_d$. The classical rule $ae^k \rightarrow (b, p)$ is translated as:

$$\left(E_{\frac{\ell}{d-1},\frac{\ell+k}{d-1}}, p\right) \equiv \left(\left|\frac{\ell+k}{d-1}\right\rangle \left\langle\frac{\ell}{d-1}\right|, p\right)$$

In this way, the object $\left|\frac{\ell}{d-1}\right\rangle$ can only be transformed into $\left|\frac{\ell+k}{d-1}\right\rangle$, and this transformation requires k units of free energy to be performed.

Another observation concerning the use of quantum rules is the following. In a classical P system, a rule of the kind $ae^k \rightarrow (b, p)$ is applied *simultaneously* to every occurrence of a in the region, provided that enough free energy units are present. Clearly, this is a mathematical abstraction. In a real quantum system, there will be a subsystem — whose behavior is described by an appropriate linear operator — which is devoted to transform an instance of a into an instance of b, using k units of free energy. Hence, only one of such transformations at the time is possible. How does this affect the computational power of quantum P systems? Notice that P systems were originally inspired by the functioning of living cells, and in living cells we have the same problem: a prescribed biological mechanism is devoted to transform one (or a limited number of) instance(s) of a into one (or the corresponding number of) instance(s) of b. Hence, even in classical P systems the possibility to simultaneously apply a rule to every instance of a is a mathematical abstraction.

Concerning the power of quantum P systems we note that, in analogy with other models of quantum computers, there is the possibility to initial-

ize the system with a multiset of objects whose state is a superposition of classical (that is, pure) states. As a result, the computation will transform such input multiset to an output multiset which is obtained by linearity as a superposition of the results of the computation on every single classical state. As usual, when we measure the state of the systems which occur into the output multiset we will obtain a pure state as a result, according to the probability distribution which is induced by the coefficients of the superposition. Another interesting aspect of quantum P systems is their behavior when some quantum systems from the input multiset are in an entangled state.

Of course we advocate the study of the computational power of quantum P systems, by comparing them both against their classical counterparts and other quantum computational models. In particular, it would be useful to define a quantum version of counter machines, since they have proven to be a useful and powerful tool in the classical setting.

Last, but not the least, we pose the problem to write the linear operator which describes an entire quantum P system, starting from the linear operators which describe the single rules. Such global operator is important, from the point of view of Quantum Computing, for the existence of a physical system that behaves like the prescribed P system. Moreover, such operators are related with the Hamiltonian of the system, which describes the internal energy of the system. The difficulty of writing the global operator is due to the fact that in quantum P systems, at each computation step a maximally parallel set of rules is *nondeterministically* chosen and applied. This contrasts the deterministic nature of the application of the global operator to the global state of the quantum P system. One possible solution, in order to always select a maximally parallel set of rules, would be to distribute the free energy contained into each region to the rules of the region, sorted in increasing order with respect to the required amount of free energy units. However, this would be equivalent to introducing a priority between the rules of each region. We ask ourselves whether this is appropriate or not.

We hope that all these problems will generate stimulating discussion on quantum P systems. In this sense, they can all be considered a starting point for future work.

6 Conclusions

In this paper we have introduced two quantum versions of energy–based P systems. Both versions are defined just like classical energy–based P sys-

tems, but for objects and rules. Objects are represented as pure states in the Hilbert space $\mathbb{C}^d$, whereas the definition of rules differs between the two models. In the former, rules are defined as bijective functions — implemented as unitary operators — which transform the objects from the alphabet. In the latter, rules are defined as generic functions which map the alphabet into itself. Such functions are implemented using a generalization of the Conditional Quantum Control technique, and may yield to non–unitary operators. Finally, we have addressed some problems and outlined some directions for future work.

Acknowledgments

We gratefully thank Gheorghe Păun, who prompted us to define quantum P systems.

References

[1] G. Alford. Membrane systems with heat control. In *Pre–Proceedings of the Workshop on Membrane Computing*, Curtea de Arges, Romania, August 2002.

[2] A. Barenco, D. Deutsch, A. Ekert, R. Jozsa. Conditional Quantum Control and Logic Gates. *Physical Review Letters*, **74**, 1995, pp. 4083–4086.

[3] P. Benioff. Quantum Mechanical Hamiltonian Models of Discrete Processes. *Journal of Mathematical Physics*, **22**, 1981, pp. 495–507.

[4] P. Benioff. Quantum Mechanical Hamiltonian Models of Computers. *Annals of the New York Academy of Science*, **480**, 1986, pp. 475–486.

[5] D. Deutsch. Quantum Theory, the Church–Turing Principle, and the Universal Quantum Computer. *Proceedings of the Royal Society of London*, **A 400**, 1985, pp. 97–117.

[6] R. P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, **21**, No. 6–7, 1982, pp. 467–488.

[7] R. P. Feynman. Quantum Mechanical Computers. *Optics News*, **11**, 1985, pp. 11–20.

[8] R. Freund. Energy–Controlled P Systems. In *Membrane Computing*, Proceedings of the International Workshop WMC–CdeA 2002, Curtea de Arges, Romania, August 2002, LNCS 2597, Springer, 2002, pp. 247–260.

[9] P. Frisco. The conformon–P system: a molecular and cell biology–inspired computability model. *Theoretical Computer Science*, **312**, 2004, pp. 295–319.

[10] D. Gottesman. Fault–tolerant quantum computation with higher–dimensional systems. *Chaos, Solitons, and Fractals*, **10**, 1999, pp. 1749–1758.

[11] J. Gruska. *Quantum Computing*. McGraw–Hill, 1999.

[12] A. Leporati, C. Zandron, G. Mauri. Simulating the Fredkin Gate with Energy–Based P Systems. *Journal of Universal Computer Science*, **10**, No. 5, 2004, pp. 600–619. A preliminary version is contained in [16], pp. 292–308.

[13] A. Leporati, C. Zandron, G. Mauri. Universal families of Reversible P Systems. In *Machines, Computation and Universality* (MCU 2004), Saint–Petersburg, Russia, September 21–26, 2004. To appear in Lecture Notes in Computer Science.

[14] G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, **1**, No. 61, 2000, pp. 108–143. See also Turku Centre for Computer Science – TUCS Report No. 208, 1998.

[15] G. Păun. *Membrane Computing. An Introduction*. Springer–Verlag, Berlin, 2002.

[16] G. Păun, A. Riscos Nuñez, A. Romero Jiménez, F. Sancho Caparrini (Eds.). Second Brainstorming Week on Membrane Computing, Seville, Spain, February 2–7, 2004. Department of Computer Sciences and Artificial Intelligence, University of Seville TR 01/2004.

[17] G. Păun, Y. Suzuki, H. Tanaka. P Systems with energy accounting. *International Journal Computer Math.*, **78**, No. 3, 2001, pp. 343–364.

[18] The P systems Web page: `http://psystems.disco.unimib.it/`