Intrinsically universal
\(n\)-dimensional quantum cellular automata

Pablo Arrighi and Jonathan Grattage

University of Grenoble, Laboratoire LIG,
Bâtiment IMAG C, 220 rue de la Chimie,
38400 Saint-Martin-d’Hères, France

Abstract. We describe an \(n\)-dimensional quantum cellular automaton (QCA) capable of simulating all others, in that the initial configuration and the forward evolution of any \(n\)-dimensional QCA can be encoded within the initial configuration of the universal QCA. Several steps of the universal QCA then correspond to one step of the simulated QCA. The simulation preserves the topology in the sense that each cell of the simulated QCA is encoded as a group of adjacent cells in the universal QCA. The encoding is linear and hence does not carry any of the cost of the computation. Part of our proof consists of showing that any QCA can be presented in the more canonical, operational form of a Partitioned QCA, thereby showing an equivalence between many definitions of QCA that are present in the literature.

1 Introduction

1.1 Cellular automata and intrinsic universality

There are many models of distributed computation (CCS, \(\pi\)-calculus…), but often in such models the notion of space has little to do with our everyday understanding of the concept, such as our intuitive notion of relative distances between areas of a 3D space, etc. These models are not adequate to reason about simple space-sensitive synchronisation problems, such as ‘Machine self-reproduction’ [12, 50] or the ‘Firing squad’ problem [29, 32]. In contrast, cellular automata (CA) were invented to precisely model spatially distributed computation in space as we know it [45]. CA consist of an array of identical cells, each of which may take one of a finite number of possible states. The whole array evolves in discrete time steps by iterating a function \(\Delta\). Moreover this global evolution \(\Delta\) is shift-invariant (it acts the same way everywhere) and causal (information cannot be transmitted faster than some fixed number of cells per time step). This model of computation is therefore physics-like [27], as it shares some fundamental symmetries of theoretical physics: homogeneity (invariance of the physical laws in time and space), causality, and often reversibility. The way in which CA are programmed is also physics-like, often constructing ‘signals’ or ‘particles’ that ‘interact’ or ‘collide’ with one another. Often demonstrating that a CA is universal for computation feels very much like building an entire computer
from scratch, with wires and gates, in the ‘virtual world’ of the CA [15]. The most popular cellular automaton is Conway’s ‘Game of Life’, a two-dimensional CA which has been shown to be universal for computation, in the sense that any algorithm can be encoded within its initial state and then executed by the evolution of the CA. This was accomplished by simulating any Turing Machine (TM) within the automaton. As TM have long been regarded as the best definition of ‘what an algorithm is’ in classical computer science, this could have been the conclusion of the topic. Yet CA research has always been looking for more than just running any algorithm; it seeks to run distributed algorithms in a distributed manner, modelling phenomena together with their spatial structure, or making use of the spatial parallelism inherent to the model. These features, which are modelled by CA and not by TM, are all of interest. Thus the notion of what universality is and means has to be revisited in this context, taking us back to the original meaning of the word universal [1, 6, 15], namely the ability for one instance of a computational model to be able to simulate all other instances of the same computational model. The introduction of a partial order on CA via the notion of grouping [30] and subsequent generalisations of this notion [36, 43], have led to elegant and robust definitions of intrinsic universality, as an extremum of this partial order. There is now an impressive number of results relating to intrinsically universal CA; i.e. those that are capable of simulating all others efficiently and directly [13, 36]. They can, of course, also simulate those CA which are capable of simulating the TM. In the non-reversible case some of the best constructions are given in [35, 37]. Closer to our setting there are intrinsically universal Reversible CA constructions by Durand-Lose [16, 17]. Notice that the work on intrinsic universal Reversible CA in more than one dimensions was completed before the single dimension case, as the one-dimension case is harder due to the lack of space (e.g. to implement wire-crossings). Notice also that the difficulty is in having an \(n\)-dimensional reversible CA simulate all other \(n\)-dimensional reversible CA and not, say, the \((n-1)\)-dimensional reversible CA – otherwise we could use a history-keeping dimension, as in Toffoli [44].

1.2 QCA and intrinsic universality

Von Neumann provided the modern axiomatisation of quantum theory in terms of the density matrix formalism [49] in 1955, and the CA model of computation in 1966, but he did not bring the two together. Feynman did [19] in 1986, just as he was inventing the concept of Quantum Computation (QC). Besides wanting to extend these historic developments, let us now list the motivations that have brought people from different communities to the study of QCA (the first two are the original ones by Feynman). Reviews of QCA are found in refs. [2, 11, 52].

– Implementation perspective. QCA may provide an important path to realistic implementations of QC, mainly because they eliminate the need for an external, classical control over the computation and hence the principal source of decoherence. This route is continuously under investigation [10, 24, 33, 46, 48].
– **Simulation perspective.** QC were first invented as a means to efficiently simulate other quantum physical systems. This remains perhaps one of the most likely applications of QC. However, it may not be straightforward to encode the theoretical description of a quantum physical system into a QC in a relevant manner, i.e., so that the QC can provide an accurate and efficient simulation. QCA constitute a natural theoretical setting for this purpose, in particular in the case of Quantum Lattice-Gas Automata [8, 9, 18, 26, 31].

– **CA perspective.** One-dimensional CA consist of a line of cells, each of which may take one in a finite number of possible states, and which evolve in discrete time steps according to a local transition rule, applied synchronously and homogeneously across space. Hence they account for many of the symmetries of physics. Because CA are a physics-like model of computation (a term coined by Margolus in [27]) it is therefore natural to study their quantum extensions (as in ref. [28]).

– **Models of computation perspective.** Shaken by the advent of QC, theoretical computer science continues to ask ‘What is a computer, ultimately?’ QCA provide a model of QC which, just like CA, takes into account space as we know it. Hence they constitute a framework to model and reason about problems in spatially distributed QC.

– **Theoretical physics perspective.** QCA could provide helpful toy models for theoretical physics, as was advocated for example by [25], by providing bridges between computer science notions and modern theoretical physics, such as universality.

It is often said that modern computers are inadequate for predicting the outcome of some quantum physics experiments. Is it not the case that a quantum physical system should be able to simulate a physical system efficiently? Is it that the theory behind computer science is correct, but that the underlying assumptions are faulty in some way, and that models of computation are not independent from physics. Feynman argued along these lines in [19], giving rise to the field of QC. Shor [42] and Grover [21] unravelled the major impact that these models have on complexity theory. This is just another case of physics providing a new method of computation, but here some of the previously accepted theoretical results of computer science needed to be revisited, which goes against one of the foundational principles of computer science, *hardware-independence*. In fact, the computing process has become physical again. Many computer scientists have attempted to address this issue. In essence, this is what QC science is all about. Over the last twenty years there has been a number of quantisations of the classical models of computation, of which this work on QCA could be considered an example. Whilst computability was not as shaken by the advent of these models of QC, complexity theory has undergone quite an overhaul, see for example [7]. We can choose to take a less defensive attitude, and decide to lead our own
incursion into accepted physics. As the frontiers between disciplines are blurred, and the computing process is now physical again, the opposite question arises: ‘To what extent are physical processes...computational?’ This would be part of a bigger trend where theoretical physics departs from looking at ‘matter’ (particles interacting, scattering, forces, etc.) and seeks to look at ‘information’ (entropy, observation, information exchanges between systems, etc.), in an attempt to clarify its own concepts. There are many examples of this; we could mention for instance the huge impact that quantum information theory has had on the understanding of foundational concepts such as entanglement [14] and decoherence [38]. But coming back to theoretical computer science, and excluding information theory for that matter, what concepts have we that could be of use to theoretical physics?

The main concept in computation theory is universality. An instance of a model of computation is universal if it can simulate any other, and this may also be a useful notion in physics. Finding a minimal, universal physical phenomena will provide us with something simple to frame, so that the focus can be on reconciling the mechanics, and yet rich enough so as to have a guarantee that we can later fit some arbitrarily complex phenomena in this reconciled mechanics. Now, there are simple objections to these arguments:

– Firstly, the fact that a universal TM can simulate just about anything to an approximation is not enough. What is required is to be able to simulate each object independently in its own space. The universal physical phenomena we seek for should be some elementary unit of computation that can be plugged together to form a 3D network. Our notion of universality must account for space and interactions across space in a satisfactory manner.

– Secondly, the fact that the universal TM is so slow at simulating quantum physical phenomena suggests that the TM is not rich enough. The universal physical phenomena we seek should therefore be a universal model of quantum computation. Our notion of universality must account for the cost of simulation.

In this paper we formalise a notion of universality which fits both these criteria, namely intrinsic universality over QCA.

1.3 Relation to previous work

There have been several different non-axiomatic approaches to defining QCA [51, 47, 10, 33, 39, 23, 22, 11]. A subclass of QCA which is the most canonical of these non-axiomatic definitions is the Partitioned QCA. At the same time an axiomatisation of QCA emerged, first in the one-particle subcase [20, 31] and later in the general case [5, 4, 40], which consists solely of an enumeration of the properties which the global evolution of the QCA ought to have. The question of whether these general QCA can be brought down, without loss of generality, to more concrete, operational forms of [10, 11, 23, 22, 33, 39, 47, 51] is apparent.
It was shown [4] that any QCA can be put into the form given in [11], and are therefore equivalent in that one can be simulated by the other. In this work we will show that any QCA can be put into the form of a Partitioned QCA, so that we only need to find an intrinsically universal Partitioned QCA. Thus we reconcile all of the above mentioned non-axiomatic definitions of QCA, showing that they can all simulate one another, and show they are all equivalent to the axiomatic definition.

In the realm of QC, Watrous [51] has proved that QCA are universal in the sense of QTM. Then Shepherd, Franz and Werner [41] defined a class of QCA where the scattering unitary $U_i$ changes at each step $i$ (CCQCA). Via this construct they built a QCA of cell-dimension 12 which is universal in the circuit-sense. Universality in the circuit-sense had already been achieved by Van Dam [47], Cirac and Vollbrecht [48], Nagaj and Wocjan [33] and Raussendorf [39] – the latter uses a two-dimensional QCA but has this inspiring idea of programs crossing the data, with computation occurring in the interaction.

There is no previous work on intrinsically universal QCA before we addressed the one-dimensional case [3]. Given the crucial role of this notion in the classical CA theory, it was natural to tackle this issue in the quantum setting.

The necessary theoretical background to QCA is provided in section 2, and the notion of intrinsic simulation is transposed to this theory. In section 3 the various alternative definitions of QCA are shown to be equivalent to the simplest, i.e. Partitioned QCA. In section 4 the intrinsically universal QCA is constructed. Section 5 concludes with a discussion and ideas for future directions.

2 Theory

![Partitioned one-dimensional QCA with scattering unitary $U$](image_url)

Fig. 1. Partitioned one-dimensional QCA with scattering unitary $U$. Each line represents a quantum system, in this case a whole cell. Each square represents a scattering unitary $U$ which is applied to two cells. Time flows upwards.
Fig. 2. Intrinsic simulation of one QCA by another. The QCA defined by $U$ simulates the QCA defined by $V$. In this case two cells of the $U$-defined QCA are required to encode one cell of the $V$-defined QCA, and we need to run the $U$-defined QCA for four time steps to simulate one time step of the $V$-defined QCA. More generally the challenge is to define an initial configuration of the $U$-defined QCA so that it behaves just as the $V$-defined QCA with respect to the encoded initial configuration, after some fixed number of time steps. Such an encoding must hold the configuration of the $V$-defined QCA as well as a way of describing the scattering unitary $V$.

2.1 $n$-dimensional QCA

The fundamental definitions for $n$-dimensional QCA are now presented.

**Definition 1 (Finite configurations)** A (finite) configuration $c$ over $\Sigma$ is a function $c : \mathbb{Z}^n \rightarrow \Sigma$, with $(i_1, \ldots, i_n) \mapsto c(i_1, \ldots, i_n) = c_{i_1 \cdots i_n}$, such that there exists a (possibly empty) interval $I$ verifying $(i_1, \ldots, i_n) \notin I \Rightarrow c_{i_1 \cdots i_n} = q$. The smallest such interval $I$ is called interval domain of $c$, and is denoted $\text{idom}(c)$. The set of all finite configurations over $\Sigma$ will be denoted $\mathcal{C}_\Sigma^n$.

Whilst configurations hold the basic states of an entire array of cells, and hence denote the possible basic states of the entire QCA, the global state of a QCA may be a superposition of these. The following definition holds as $\mathcal{C}_\Sigma^n$ is a countably infinite set.

**Definition 2 (Superpositions of configurations)** Let $\mathcal{H}_{\mathcal{C}_\Sigma^n}$ be the Hilbert space of configurations. To each finite configuration $c$ is associated to a unit vector $|c\rangle$, such that the family $(|c\rangle)_{c \in \mathcal{C}_\Sigma^n}$ is an orthonormal basis of $\mathcal{H}_{\mathcal{C}_\Sigma^n}$. A superposition of configurations is then a unit vector in $\mathcal{H}_{\mathcal{C}_\Sigma^n}$.
Definition 3 (Unitarity) A linear operator \( G : \mathcal{H}_{c_f} \rightarrow \mathcal{H}_{c_f} \) is unitary if and only if \( \{ G|c\rangle \mid c \in c_f^2 \} \) is an orthonormal basis of \( \mathcal{H}_{c_f} \).

Definition 4 (Shift-invariance) Consider the shift operation, for \( k \in \{1, \ldots, n\} \), which takes configuration \( c \) to \( c' \) where for all \( (i_1, \ldots, i_n) \) we have \( c'_{i_1 \ldots i_k \ldots i_n} = c_{i_1 \ldots i_{k+1} \ldots i_n} \). Let \( \sigma_k : \mathcal{H}_{c_f} \rightarrow \mathcal{H}_{c_f} \) denote its linear extension to superpositions of configurations. A linear operator \( G : \mathcal{H}_{c_f} \rightarrow \mathcal{H}_{c_f} \) is said to be shift invariant if and only if \( G \sigma_k = \sigma_k G \) for each \( k \).

Definition 5 (Causality) A linear operator \( G : \mathcal{H}_{c_f} \rightarrow \mathcal{H}_{c_f} \) is said to be causal with radius \( \frac{1}{2} \) if and only if for any \( \rho, \rho' \) two states over \( \mathcal{H}_{c_f} \), and for any \( i \in \mathbb{Z} \), we have
\[
\rho|_N = \rho'|_N \Rightarrow G(\rho)G^\dagger|_{i_1 \ldots i_n} = G(\rho')G^\dagger|_{i_1 \ldots i_n}
\]
where \( \rho|_N \) means the restriction of \( \rho \) to the neighbourhood \( N \) in the sense of the partial trace, and \( G^\dagger \) is the Hermitian adjoint of \( G \), where \( N = \{i_1, i_1 + 1\} \times \ldots \times \{i_n, i_n + 1\} \).

In the classical case, the definition is that the letter to be read in some given cell \( i \) at time \( t+1 \) depends only on the state of the cells \( i \) to \( i+1 \) at time \( t \). Transposed to a quantum setting, we obtain the above definition. To know the state of cell number \( i \), we only need to know the states of cells \( i \) and \( i+1 \) before the evolution.

More precisely, this restrictive definition of causality is known in the classical case as a \( \frac{1}{2} \)-neighbourhood cellular automaton, because the most natural way to represent such an automaton is to shift the cells by \( \frac{1}{2} \) at each step, so that the state of a cell depends on the state of the two cells under it. This definition of locality is not so restrictive, since by grouping cells into “supercells” any CA with an arbitrary finite neighbourhood \( N \) can be made into a \( \frac{1}{2} \)-neighbourhood CA. The same method can easily be applied to QCA, so this definition of locality is essentially achieved without loss of generality.

The formal definition of \( n \)-dimensional QCA can now be given:

Definition 6 (QCA) An \( n \)-dimensional quantum cellular automaton (QCA) is an operator \( G : \mathcal{H}_{c_f} \rightarrow \mathcal{H}_{c_f} \) which is unitary, shift-invariant and causal.

This is the natural axiomatic quantisation of the notion of CA. It was given in [4, 5], but stems from an equivalent definition in the literature, phrased in terms of homomorphism of a \( C^* \)-algebra [40]. There have also been several non axiomatic approaches to QCA.

2.2 Intrinsic simulation of \( n \)-dimensional QCA

The notion of intrinsic simulation of one CA by another was discussed in section 1.1. The notion of quantising intrinsic simulation used here follows ref. [36]. The intuition is to say ‘\( G \) simulates \( H \)’ by translating the contents of each cell of \( H \)
into cells of $G$, running $G$, and then reversing the translation; this three step process amounts to running $H$. This translation should be simple (the cost of the computation will be carried over only by $G$), should preserve the topology (each cell of $H$ is encoded into cells of $G$ in a way which preserves neighbours), and should be faithful (no information should be lost in translation). This latter requirement relates to a precise notion in quantum theory, that of isometry, i.e. an inner product preserving evolution with $\text{Enc}I\text{Enc} = I$. This same requirement is in line with the translation being a physical process. The following definitions are thus derived.

**Definition 7 (Isometric coding)** Consider $\Sigma_G$ and $\Sigma_H$, two alphabets with distinguished quiescent states $q_G$ and $q_H$, and such that $|\Sigma_H| \leq |\Sigma_G|$. Consider $\mathcal{H}_{\Sigma_G}$ and $\mathcal{H}_{\Sigma_H}$ the Hilbert spaces having these alphabets as their basis, and $\mathcal{H}_{\Sigma_F}$, $\mathcal{H}_{\Sigma_F'}$ the Hilbert spaces of finite configurations over these alphabets.

Let $E$ be an isometric linear map from $\mathcal{H}_{\Sigma_H}$ to $\mathcal{H}_{\Sigma_G}$ which preserves quiescence, i.e. such that $E|q_H\rangle = |q_G\rangle$. It trivially extends into an isometric linear map $\text{Enc} = (\bigotimes_{\Sigma_F} E)$ from $\mathcal{H}_{\Sigma_F}$ into $\mathcal{H}_{\Sigma_F'}$, which we call an isometric encoding.

Let $D$ be an isometric linear map from $\mathcal{H}_{\Sigma_G}$ to $\mathcal{H}_{\Sigma_H} \otimes \mathcal{H}_{\Sigma_G}$ which also preserves quiescence, in the sense that $D|q_G\rangle = |q_H\rangle \otimes |q_G\rangle$. It trivially extends into an isometric linear map $\text{Dec} = (\bigotimes_{\Sigma_F} D)$ from $\mathcal{H}_{\Sigma_F'}$ into $\mathcal{H}_{\Sigma_H} \otimes \mathcal{H}_{\Sigma_F'}$, which we call an isometric decoding.

The isometries $E$ and $D$ define an isometric coding if the following condition is satisfied:

$\forall \psi \in \mathcal{H}_{\Sigma_F'}, \exists \phi \in \mathcal{H}_{\Sigma_F'} \quad |\psi\rangle \otimes |\phi\rangle = \text{Dec}(\text{Enc}|\psi\rangle)$.

(The understanding here is that $\text{Dec}$ is morally an inverse function of $\text{Enc}$, but we may leave out some garbage $|\phi\rangle$ on the way.)

**Definition 8 (Direct simulation)** Consider $\Sigma_G$ and $\Sigma_H$, two alphabets with distinguished quiescent states $q_G$ and $q_H$, and two QCA $G$ and $H$ over these alphabets. We say that $G$ directly simulates $H$, if and only if there exists an isometric coding such that

$\forall i \in \mathbb{N}, \forall \psi \in \mathcal{H}_{\Sigma_F'}, \exists \phi \in \mathcal{H}_{\Sigma_F'} \quad (G^i|\psi\rangle) \otimes |\phi\rangle = \text{Dec}(H^i(\text{Enc}|\psi\rangle))$.

Unfortunately this is not enough for intrinsic simulation. Often we want to say that $G$ simulates $H$ even though the translation:

- takes several cells of $H$ into several cells of $G$;
- demands several steps of $G$ in order to simulate several steps of $H$.

Hence the notion of grouping is required, introduced here:

**Definition 9 (Grouping)** Let $G$ be a QCA over alphabet $\Sigma$. Let $s$ and $t$ be two integers, $q'$ a word in $\Sigma' = \Sigma^s$. Consider the iterate global evolution $G^t$ up to a grouping of each hypercube of $s^n$ adjacent cells into one supercell. If this operator can be considered to be a QCA $G'$ over $\Sigma'$ with quiescent symbol $q'$, then we say that $G'$ is an $(s,t,q')$-grouping of $G$. 
Definition 10 (Intrinsic simulation) Consider $\Sigma_G$ and $\Sigma_H$, two alphabets with distinguished quiescent states $q_G$ and $q_H$, and two QCA $G$ and $H$ over these alphabets. We say that $G$ intrinsically simulates $H$ if and only if there exists $G'$ some grouping of $G$ and $H'$ some grouping of $H$ such that $G'$ directly simulates $H'$.

In other words, $G$ intrinsically simulates $H$ if and only if there exists some isometry $E$ which translates supercells of $H$ into supercells of $G$, such that if we then iterate $G$ and translate back, the whole process is equivalent an iteration of $H$. This understanding is captured by Fig. 3.

![Diagram](image)

Fig. 3. The notion of intrinsic simulation made formal.

3 The structure of QCA

The axiomatics of QCA were introduced in section 2.1, but in abstract, mathematical, terms for which it seems difficult to find an intrinsically universal instance. However, the structure theorem of ref. [4] states that any such QCA can be simulated by a more operational QCA, and this will be developed here.

3.1 Block representation

Theorem 1 (n-dimensional QCA multi-layered block representation)

Let $G$ be an $n$-dimensional QCA with alphabet $\Sigma$. Let $E$ be an isometry from $\mathcal{H}_\Sigma \rightarrow \mathcal{H}_\Sigma \otimes \mathcal{H}_\Sigma$ such that $E|\psi_x\rangle = |q\rangle \otimes |\psi_x\rangle$. This mapping can be trivially extended to whole configurations, yielding a mapping $E : \mathcal{H}_{C^2} \rightarrow \mathcal{H}_{C^2^2}$. There then exists an $n$-dimensional QCA $H$ on alphabet $\Sigma^2$, such that $HE = EG$, and $H$ admits a $2^n$-layer block representation. Moreover $H$ is of the form

$$H = (\bigotimes S)(\prod K_x)$$  \hspace{1cm} (1)

where:
– \((K_x)\) is a collection of commuting unitary operators all identical up to shift, each localised upon each neighbourhood \(N_x\);

– \(S\) is the swap gate over \(\mathcal{H}_\Sigma \otimes \mathcal{H}_\Sigma\), hence localised upon each node \(x\).

In section 1.3 we have explained how several authors, given the apparent lack of operationality of the axiomatics of QCA, have defined them directly as quantisations of block representations of CA. Amongst those various definitions only the one by Perez-Delgado and Cheung [11] is not two-layered. It stands out at this stage, as it just directly posits, after some interesting informal arguments, that they are of a form very akin to the one given by Equation (1).

3.2 Down to two layers: Block QCA

In the literature most quantisations of block representations of CA are two-layered; cf. [47, 10, 33, 39, 40, 23]). This is captured by the definition of a Block QCA (BQCA):

**Definition 11 (Block QCA)** A block \(n\)-dimensional quantum cellular automaton (BQCA) is defined by two unitary operators \(U_0\) and \(U_1\) such that \(U_i : \mathcal{H}_\Sigma^{\otimes 2^n} \rightarrow \mathcal{H}_\Sigma^{\otimes 2^n}\), and \(U_i|qq \ldots qq\rangle = |qq \ldots qq\rangle\), i.e. each takes \(2^n\) cells into \(2^n\) cells and preserves quiescence. Consider \(G_i = (\otimes_{2^n} U_i)\) the operator over \(\mathcal{H}\). The induced global evolution is \(G_0\) at odd time steps, and \(\sigma G_1\) at even time steps, where \(\sigma\) is a translation by one in all directions (Fig. 4).

![Fig. 4. Block QCA. The elementary unitary evolutions \(U_0\) and \(U_1\) are alternated repeatedly as shown.](image-url)

The equivalence of the QCA and BQCA axiomatics is interesting. BQCA are unitary, causal, and shift-invariant, and hence fall under the axiomatics and
Theorem 1 (strictly speaking we need to group each hypercube of $2^n$ adjacent cells into a supercell, see Definition 9.) Whether BQCA can simulate any QCA of that form is less trivial.

In the form given by Theorem 1, each cell $x$ at time $t$ is successively involved in $2^n$ computations governed by a local unitary $K$, whose aim is to compute the next state of a cell within a radius $\frac{1}{2}x$ at time $t+1$. In 2 dimensions a cell $x$ relates to the cells West, North-West and North to work out its North-West successor, and then with the cells North, North-East, East of it to compute the North-East successor. Similarly for the South-East and the South-West successors. To mimic this with a BQCA, each original cell can be encoded into four cells, arranged so that the original cell $x$ starts in the North-West quadrant of the four cells. The first layer of the BQCA applies $K$ to compute the North-West successor of $x$. The second layer of the BQCA moves the original cell $x$ in the North-West quadrant. Each full application of the evolution of the BQCA corresponds only to one layer ($\bigotimes K$), hence it will take four steps for this BQCA to simulate one step of the QCA. Fig. 5 gives a sketch of the method used.

**Fig. 5.** Sketch of a BQCA simulating a QCA. The original cell $x$ is coded into four cells, at the centre ($E$). It starts by considering the North-West as at time 0 it will compute its North-West successor, and then move clockwise. At time 1 it will compute its North-East successor etc.

There are some issues to be noted here. Whereas cell $x$ is turning clockwise in the example, the cell to its North is turning anticlockwise. Hence we need some ancillary data coding for the path to be taken by the original cell $x$ within the four coding cells. Also, Theorem 1 finishes with a Swap between the ‘computed tape’, where the results have been stored, and the ‘uncomputed tape’, (i.e. what remains of the original cell after having computed all of its successors) which is not shown in the sketch. Hence the number of layers of $K$ so far computed has to be tracked, so that the Swap occurs at the appropriate step. The Swap also needs to know where the results have been stored in order to move them correctly. All of this has to be arranged spatially and efficiently, and one such method is shown by Fig. 6 and Fig. 7.
Fig. 6. BQCA simulating a QCA. The grey areas denote the neighbourhood where the action of $k_x$, the first layer of the BQCA, will be significant — i.e., a group of four cells where it will perform a $K_x$ operation to work out a successor. Where this successor will be stored is indicated by $(R_x)$. At the next step we see that $R_x$ has appeared, and also that registers have been reshuffled thanks to the second layer of the BQCA, which acts according to the rotation-direction mark. The second layer also increases the clock count and includes the final swapping step, which only happens at time 3. There it ensures that $R_0$ becomes $A$, $R_1$ becomes $B$, etc. Which registers are to be swapped with one another can be worked out from the rotation and arrow marks. Each step is made formal by Fig. 7.

Fig. 7. Operations used in fig. 6. The $k$ applies a $K$ operation whenever some data is present (data carries an extra bit to distinguish it from $|q\rangle$, say). The $U$ operation reshuffles the data by rotating it in the direction given by the indicator in the top left (clockwise or anticlockwise), and increments the index counter. Lastly, $cswap$ acts as the identity in all cases except when the index is 3, when it swaps the result of the computations with the data, ready for the next round.
BQCA can therefore simulate QCA, up to a relatively simple encoding. We have shown this for two dimensions, but this construct clearly generalises to $n$-dimensions. Hence QCA (as in Definition 6) provide a rigorous axiomatics for BQCA (as in Definition 11), and BQCA provide a convenient operational description of QCA.

### 3.3 Down to one scattering unitary: Partitioned QCA

How previous research has defined quantisations of partitioned representations of CA [51, 47, 22] was discussed in section 1.3. An interesting question to consider now is whether QCA (as in Definition 6) provide a rigorous axiomatics for Partitioned QCA (PQCA, Definition 12), and if PQCA provide a convenient operational description of QCA.

**Definition 12 (Partitioned QCA)** A partitioned $n$-dimensional quantum cellular automaton (PQCA) is defined by a unitary operators $U$ such that $U : 2^n \otimes \Sigma^m \rightarrow 2^n \otimes \Sigma^m$, and $U|qq\ldots qq\rangle = |qq\ldots qq\rangle$, i.e. that takes $2^n$ cells into $2^n$ cells and preserve quiescence. Consider $G = (\bigotimes_{2^n} U)$, the operator over $\mathcal{H}$. The induced global evolution is $G$ at odd time steps, and $\sigma G$ at even time steps, where $\sigma$ is a translation by one in all directions, see Fig. 1.

[Fig. 8. PQCA simulating a BQCA. The QCA is decorated with control qubits following a simple encoding procedure (left), which allow the scattering unitary $U$ (centre) to act as either $U_0$ or $U_1$, according to the layer (right). The black box can be any unitary.]

Following our previous results (section 3.2), it is only required to show that PQCA can simulate BQCA. Both PQCA and BQCA are two-layered, the only difference is that for BQCA those two-layers may be different (e.g. compare Fig. 1 and 4). So if a PQCA is $U$-defined, with a $U$ capable of performing $U_0$ and $U_1$ alternatively as controlled by some ancillary, this will suffice. This was done for one-dimension in [3] and is given here for two dimensions in Fig. 8, but it is clear that our construct generalises to $n$-dimensions.

Therefore we can conclude that PQCA are the most canonical and yet general operational description of QCA. More generally, this demonstrates that a well-
axiomatised, and yet concrete, operational notion of $n$-dimensional QCA is now available.

4 Intrinsic universality

4.1 Requirements of universality

In section 2.1 the formal definition of $n$-dimensional QCA was discussed, and in section 2.2 a formal definition for the notion of intrinsic simulation was provided, in section 3 it was noted that they have a simple circuit-like structure, and the equivalence between their various forms was discussed. Ultimately the picture one needs to have in mind in order to follow this section just that of Fig. 2: the aim is to find a particular $U$-defined QCA which is capable of intrinsically simulating any $V$-defined QCA, whatever the $V$. In order to describe that $U$-defined QCA two things must be first defined:
- The cell structure must be specified, such as how are the vertical lines of Fig. 1 composed, and what is the dimensionality of $U$. By the definition of QCA they are finite dimensional quantum systems of some fixed dimension $d$, but for clarity they will be decomposed into subsystems of dimension $d_i$, which will be named according to their function (e.g. the vertical lines of Fig. 1 are ‘buses’). Subsystems have names written in bold.
- How $U$ acts upon a pair of these cells, and more precisely upon the subsystems making up the pair of cells. This is presented informally, but a formal circuit description of $U$ is also provided. We shown that such a $U$ is indeed unitary.
FIRST we discuss the mechanism used to solve this problem, then we address the QCA components in turn. Finally, these concepts are brought together to give an intrinsically universal QCA.

4.2 Circuit universality versus intrinsic universality in higher dimensions

Intrinsic universality refers to the ability for one CA to simulate any other CA in a way which preserves the spatial structure of the simulated CA. A CA that can simulate any Turing Machine, and hence run any algorithm, is called TM-universal; this is also known as computation-universality. Additionally, circuit universality is the ability of one CA to simulate any circuit. e.g. NAND gates for classical circuits and CA, or TOFFOLI gates for reversible circuits and CA, etc. Informally, in the quantum setting, this requires a QCA which is capable of simulating a unitary evolution expressed as a combination of a universal set of quantum gates, such as the standard gate set: CNOT, PHASE, and HADAMARD. The relationships between these notions of CA universality is interesting [15]. A computation universal CA is also a circuit universal CA, because circuits are simply finitary computations. Moreover, an intrinsic universal CA is also a computation universal CA, because it can simulate any CA; including computation universal CA. Hence intrinsic universality implies computation universality,
which implies circuit universality. Is this an equivalence?
In one-dimension we see immediately that this does not result directly in an
equivalence. Intuitively, computation universality requires more than circuit uni-
versality, namely the ability to loop the computation, which is not trivial for
CA. Similarly, intrinsic universality requires more than computation universal-
ity, such as the ability to simulates multiple communicating TM. In the classical
setting there are formal results to distinguish them [36].
In \( n \)-dimensions we commonly assume that circuit universality implies intrinsic
universality, and hence all of these notions are equivalent [36]. Strictly speaking
this is not true. For example, consider a 2-dimensional CA which runs one-
dimensional CA in parallel. If the one-dimensional CA is circuit/computation
universal, but not computation/intrinsically universal, then this is also true for
the 2-dimensional CA. In the QCA setting it appears that the 2-dimensional
constructions in [11] and [39] are indeed circuit universal but not intrinsically
universal. However, this remains a useful assumption.

CA admit a block representation, and for reversible CA these blocks are per-
mutations. In section 3 we saw that QCA also admit a block representation,
where the blocks are unitary matrices. Thus we can express the evolution of any
(Quantum/Reversible) CA as an infinite (quantum/reversible) circuit of (quan-
tum/reversible) gates repeating across space. We can say that if a CA is circuit
universal, and it has the ability of wiring together different circuit-pieces in dif-
ferent regions of space, then it can simulate the block representation of any CA,
and hence it can simulate any CA in a way which preserves its spatial structure.
It is intrinsically universal.

This is the route we will follow to construct our intrinsically universal \( n \)-dimen-
sional QCA. First we see how to construct the ‘wires’ which can carry information
across different regions of space (here these are signals which can be redirected
or delayed by using barriers, with each signal holding a qubit of information).
Secondly, we see how to construct the ‘circuit-pieces’, i.e. how to implement
gates and combine them. One and two qubit gates are implemented as obstacles
to and collisions of these signals.

To complete the proof formally, we show that since any \( n \)-dimensional QCA can
be expressed as a PQCA, we can flatten this infinitely repeating two-layered
circuit into space (so that at the beginning all the signals carrying qubits find
themselves in circuit-pieces implementing one of the scattering unitary of the
first layer, and then they all synchronously exit and travel to circuit-pieces im-
plementing the scattering unitary of the second layer, etc.).

An algorithm for performing this flattening could be provided, however we do
not describe the process in too high a level of detail here to maintain clarity,
as per the corresponding classical literature. However, a good intuition of the
flattening process for a 2-dimensional PQCA is presented in Fig. 9 to 12, and
this clearly applies to \( n \)-dimensions.
4.3 Signals carrying qubits, barriers

First we need to understand how to code signals travelling along the cardinal directions of space. Each cell requires two axis subsystems, \textit{x-axis} and \textit{y-axis} which can be either empty, or hold a single qubit. The direction of propagation does not need to be signed; it suffices to know that a particle is travelling along some axis (\textit{e.g.} the \textit{x-axis}) and to know the parity of its position (\textit{e.g.} on a left cell or a right cell) to know the direction in which it is travelling along the axis.
Fig. 11. Flattening a PQCA into a UQCA: Step 3. Within the central square of Fig. 10, the incoming signals are bunched together so as to undergo a circuit which implements $V$. Then they are then dispatched towards the four corners. This diagram does not make explicit a number of signal delays, which may be needed to ensure that they arrive synchronously at the beginning of the circuit implementing $V$.

This is because whenever a unitary interaction $U$ acts upon a square of four cells, there is only one choice of cell for the propagation of the signal (a signal at the bottom right travelling along the $x$-axis can only move to the bottom left). The ability to redirect these signals is also required, and is achieved by ‘bouncing’ them off a barrier. This is all the structure required for our intrinsically universal $U$-defined QCA.

The movement operation is shown in Fig. 13 (see also Fig. 22), which acts on a partition of four cells. The effect of the barrier subsystem is given in Fig. 14, and acts on individual cells; note that for now we assume that only one axis subsystem is carrying a qubit. Collisions, when two qubits are occupying the
same cell, are discussed later. The movement permutation is applied to the four cell partition first, followed by the single cell operations.

Fig. 13. Basic movement of signals: Each cell is denoted by a single square split into three. The central circle holds the barrier subsystem, while the top-left triangle holds the $x$-axis subsystem, and the bottom-right triangle holds the $y$-axis subsystem (note grey arrows). Movement leaves the barrier subsystem unchanged, and acts as a simple permutation of the axis subsystems, shown in the diagram by the variables $a$ to $h$. This operation is self-inverse on the basis states, and hence unitary.

Fig. 14. Barrier subsystem, $\times$: when there is only one occupied axis, the action simply exchanges the axis subsystems. The barrier subsystem remains unchanged.

Using only these operations we can easily delay (Fig. 15) and swap (Fig. 16) signals. For signals to be correctly synchronised, each operation takes 24 time-steps, and each qubit operates in a $4 \times 16$ cell tile so that they can be plugged together as in Fig. 12.

### 4.4 Collisions and derived gates

Two qubit-carrying signals may meet and collide. No extra structure is required to enable this, it is simply modelled by both the $x$-axis and the $y$-axis subsystems of a single cell holding a qubit. To allow a universal set of gates to be implemented by the QCA, extra meaning must be attached to the collision of two qubits.

| Name   | Size | Function                        |
|--------|------|---------------------------------|
| $x$-axis | 3    | Empty, or holding a qubit signal: $|\epsilon\rangle$, $|0\rangle$, $|1\rangle$. |
| $y$-axis | 3    | Empty, or holding a qubit signal: $|\epsilon\rangle$, $|0\rangle$, $|1\rangle$. |
| barrier | 2    | Empty, or holding an axis-change barrier: $|\epsilon\rangle$, $\times$. |

Table 1. Subsystems of a cell dealing with qubit carrying signals and barriers.
Fig. 15. An ‘identity circuit’ tile, a $4 \times 16$ tile taking 24 time-steps made from two consecutive delay tiles. The dotted line shows the trajectory of the signal, and the arrow denotes the entry point and direction of signal propagation.

Fig. 16. The ‘swap circuit’ tile, an $8 \times 16$ tile taking 24 time-steps which permutes the two inputs. Note that no cell contains two signals at the same time-step.

signals: it is interpreted as the application of a two qubit gate. Indeed collisions between two signals will apply either the Hadamard operation on both carried qubits, or the cPhase gate (controlled-phase, with the phase change defined as $e^{i\pi/4}$) on both carried qubits, depending upon whether or not there is a barrier. If a barrier is present the signals are also deflected, as expected; otherwise the signals continue on their original trajectory. Note that if we do not want a two qubit gate to be applied, but only want to cross-over the two signals, then the swap configuration of Fig. 16 can be used. The operations for interpreting collisions are given in Fig. 17, for when the barrier subsystem is empty, and Fig. 18, for the case when a barrier is present.

Fig. 17. A collision when the barrier subsystem is empty also acts as a controlled phase (cP) operation on the qubits. If both axis signals are $|1\rangle$, then a global phase of $e^{i\pi/4}$ is added to the configuration. All subsystems remain unchanged, and in all other cases this operation is the identity. This is implemented by the $cP^*$ gate of the B circuit in Fig. 22.

From these building blocks, a two qubit cPhase gate tile, and the one qubit Phase gate and Hadamard gate tiles can be constructed, as given in Fig. 19,
A collision when a barrier is present causes the Hadamard ($H$) operation to be applied to both qubits, causing the cell to move into a superposition of four cells. The signals are also deflected by the barrier, which is denoted by the exchanging of the axis subsystems. Normalisation factors of $\frac{1}{\sqrt{2}}$ have been omitted for clarity. This is implemented by the $H^*$ gate of the $B$ circuit in Fig. 22.

Fig. 18.

20, and 21 respectively. Each gate is implemented as a $4 \times 16$ grid for single qubit operations, and an $8 \times 16$ grid for two qubit operations, with the signals entering on the second of the four rows. This is so these circuit tiles may be easily plugged together to form any circuit. The identity ‘gate’ is modelled by the delay configuration, given in Fig. 15. To ensure that all signals are synchronised, each takes exactly 24 time-steps. When wiring together non-contiguous circuits several delay gates may be required in order to ensure all signals enter the second circuit at the same time-step. Notice that our chosen set of quantum gates is indeed universal as we can recover the standard set (CNOT, H, Phase) via $\text{CNOT}\ket{\psi} = (I \otimes H)(\text{CPhase})^4(I \otimes H)\ket{\psi}$.

Fig. 19.

The ‘CPhase circuit’ tile, which applies the controlled-phase operation to the two input qubits. This is achieved by delaying the second signal and redirecting the first, causing a collision at the highlighted cell. The qubits are then synchronised so that they exit at the same time along their original paths.

4.5 The scattering unitary

An intuitive explanation of what the scattering unitary $U$ does to squares of four such cells has been provided in the previous sections. A formal description of $U$ will now be given, which shows that it is indeed unitary. The scattering unitary
Fig. 20. The ‘Phase gate’ tile. This tile makes use of a signal, set to $|1\rangle$, which loops inside the grid every 8 time-steps, ensuring that it will collide with the signal that enters the configuration and causing it to act as the control qubit to a cPhase operation. This therefore acts as a phase rotation on the input qubit, which passes directly through. After 24 time-steps the auxiliary signal has returned to its origin. The collision cell is highlighted in grey.

Fig. 21. The ‘Hadamard gate’ tile, which applies the Hadamard operation to the input signal. Like the ‘Phase circuit’ tile, this tile uses a looping signal, ensuring that it will collide with the signal that enters the tile. The auxiliary signal in this circuit loops every 4 time-steps, and after the collision, which applies the Hadamard operation to both signals, the input signal is deflected by the barrier while the auxiliary continues looping. The input signal is then rerouted to exit along its original trajectory. The collision cell is shown in grey.

$U$ takes four cells as inputs. Each of these decomposes into three subsystems: $x$-axis, $y$-axis, and barrier. Therefore it could be given as a $18^4 \times 18^4$ matrix of complex numbers, yet fortunately it decomposes as in Fig. 22. This circuit is essentially split into two sections: a permutation of the axis subsystems, and an application of a $B$ operation on every cell, controlled by their own barrier subsystem. The permutation directly implements the neighbourhood movement rule given in Fig. 13, while the $B$ operation implements the single-cell rules for barriers and collisions given in Figs. 14, 17, and 18. The controlled-$B$ operation can be further decomposed as in Fig. 23, and is made up of three gates, all controlled by the barrier subsystem, as follows:

- **cP* gate:** This gate applies the cPhase operation to the axis subsystem on the subspaces where they are both not empty (i.e. they contain either $|0\rangle$ or $|1\rangle$). This operation is only applied if the cell does not contain a barrier; the barrier subsystem $= \epsilon$. This fully implements the rule given in Fig. 17.
- **Axis permutation:** This gate permutes the values of the $x$-axis and $y$-axis subsystems of the cell, and is only applied if the cell contains a barrier. This implements the signal deflection rule given in Fig. 14, and the collision signal deflection of the Hadamard collision rule given in Fig. 18.
The scattering unitary $U$ as a quantum circuit, with time flowing upwards. The circuit takes four cells and gives back four cells.

This circuit encodes the actions of the scattering unitary that act upon only one cell. This completes the definition of the scattering unitary.

– $H^*$ gate: This gate applies the Hadamard operation to both qubit signals stored in axis subsystems, and, like the $cP^*$ operation, is the identity if either of these subsystems are empty. It is only applied if the cell contains a barrier, and this operation, along with the axis permutation above, implements fully the Hadamard collision rule given in Fig. 18.

This completes the formal definition of the scattering unitary $U$. As the evolution of $U$ has been given as a combination of smaller unitary matrices via tensors, composition and the control-construct, it is indeed unitary as required by Definition 12.

5 Conclusion

Main claim. In summary, we have first shown that

Theorem 2 (PQCA are universal) Given any $n$-dimensional QCA $H$, there exists an $n$-dimensional PQCA $G$ which simulates $H$.

This proof demonstrates that all the non-axiomatic definitions of QCA [51, 47, 10, 33, 39, 23, 22, 11] are equivalent to one another and to the axiomatic definition, in the sense that they all simulate each other. A two-dimensional QCA capable of simulating all others with linear overhead was then constructed. This intrinsically universal QCA is a Partitioned QCA (Fig. 1) of cell-dimension 18 whose scattering unitary is given explicitly (Fig. 22, 23 and section 4.5).
construction is given in two-dimensions, but it generalises to $n$-dimensions. The main, formal result can be stated thus:

**Theorem 3** There exists $G$ an $n$-dimensional $U$-defined QCA which is intrinsically universal QCA, as follows. Let $H$ be a $n$-dimensional $V$-defined QCA such that $V$ can be expressed as a quantum circuit $C$ made of gates from the set $H$, CNOT, PHASE. Then $G$ is able to intrinsically simulate $H$.

Any finite-dimensional unitary $V$ can always be approximated by a circuit $C(V)$ with an arbitrary small error $\varepsilon = \max_{\langle \psi \rangle} | |V\langle \psi \rangle - C\langle \psi \rangle||$. Assume that $G$ simulates the $C(V)$-defined QCA instead, for a region of $s$ cells and over a period $t$. The error with respect to the $V$-defined QCA will be bounded by $st \varepsilon$. This is due to the general statement that errors in quantum circuits increase at most proportionally with time and space [34]. Combined with Thm. 2, this means that $G$ is intrinsically universal, up to this unavoidable approximation.

**Discussion & future work.** So far QC has dealt with theoretical physics supporting theoretical computer science for the purposes of more secure and efficient computing. By doing so it has shaken the foundations of Computer Science, and hence QC scientists have reconsidered and redeveloped information theory, complexity theory etc. However, information theory also plays a crucial role in the foundations of theoretical physics. Generally speaking, there is a tendency to abstract away the foundational theoretical problems from particles, matter, forces and cast them in terms of information exchanges only. We believe that universality, among the many concepts that computer science has developed, will become an essential, simplifying methodology in this respect. For example, if the foundational theoretical problem crucially involves some notion of interaction, universality still makes it possible to abstract away particles, matter, forces, and cast the problem in terms of information exchanges together with some universal information processing. This work can be therefore viewed as an attempt to export the notion of universality to theoretical physics; a rigorous step on the quest for a *universal physical phenomena*, with some simplified mechanics. As the spatial arrangement of interaction is important in physics, this is another reason why intrinsic universality is preferred over computation universality. Although this QCA is universal, it probably is not minimal. In comparison, the simplest known nearest-neighbour intrinsically universal classical CA has cell dimension 4 [37]. We will pursue our research in two directions: finding a minimal intrinsically universal QCA, and determining similar universal physical phenomena for a more complex mechanics.

**Acknowledgements**

P.J.A would like to thank Jérôme Durand-Löse, Jacques Mazoyer, Nicolas Ollinger, Guillaume Theyssier and Philippe Jorrand.
References

1. J. Albert and K. Culik. A simple universal cellular automaton and its one-way and totalistic version. Complex Systems, 1:1–16, 1987.
2. P. Arrighi. Quantum cellular automata. Thèse d’habilitation à diriger les recherches, University of Grenoble, 2009.
3. P. Arrighi, R. Fargetton, and Z. Wang. Intrinsically universal one-dimensional quantum cellular automata in two flavours. Fundamenta Informaticae, 21:1001–1035, 2009.
4. P. Arrighi, V. Nesme, and R. Werner. Unitarity plus causality implies locality. Arxiv preprint arXiv:0711.3975, 2007.
5. P. Arrighi, V. Nesme, and R. F. Werner. Quantum cellular automata over finite, unbounded configurations. In Proceedings of MFCS, Lecture Notes in Computer Science, volume 5196, pages 64–75. Springer, 2008.
6. E. R. Banks. Universality in cellular automata. In SWAT ’70: Proceedings of the 11th Annual Symposium on Switching and Automata Theory (swat 1970), pages 194–215, Washington, DC, USA, 1970. IEEE Computer Society.
7. E. Bernstein and U. Vazirani. Quantum complexity theory. In Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, pages 11–20. ACM New York, NY, USA, 1993.
8. I. Bialynicki-Birula. Weyl, Dirac, and Maxwell equations on a lattice as unitary cellular automata. Physical Review D, 49(12):6920–6927, 1994.
9. B. M. Boghosian and W. Taylor. Quantum lattice-gas model for the many-particle Schrödinger equation in d dimensions. Phys. Rev. E, 57(1):54–66, 1998.
10. G. K. Durand and J. E. Williams. Entanglement dynamics in one-dimensional quantum cellular automata. Phys. Rev. A, 68(4):042311, Oct 2003.
11. D. Cheung and C. A. Perez-Delgado. Local Unitary Quantum Cellular Automata. ArXiv pre-print arXiv:0709.0006.
12. E. G. Codd. Cellular Automata. Academic Press, New York, 1978.
13. M. Delorme. An Introduction to Cellular Automata, Cellular Automata: a Parallel Model. In Mathematics and Its Applications. Kluwer, 1999.
14. W. Dürr, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. Phys. Rev. A, 62:062314, 2000.
15. B. Durand and Z. Roka. The Game of Life: universality revisited Research Report 98-01. Technical report, Ecole Normale Supérieure de Lyon, 1998.
16. J. O. Durand-Lose. Reversible cellular automaton able to simulate any other reversible one using partitioning automata. In In latin’95, number 911 in Lecture Notes in Computer Science, pages 23024–4. Springer, 1995.
17. J. O. Durand-Lose. Intrinsic universality of a 1-dimensional reversible cellular automaton. In Proceedings of STACS 97, Lecture Notes in Computer Science, page 439. Springer, 1997.
18. J. Eakins. Quantum cellular automata, the EPR paradox and the Stages paradigm. In Proceedings of NATO ARW, The Nature of Time: Geometry, Physics and Perception, 2003.
19. R. P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21(6):467–488, 1982.
20. G. Grössing and A. Zeilinger. Quantum cellular automata. Complex Systems, 2(2):197–208, 1988.
21. L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. Found. Phys Phys Rev Lett, 79:325, 1993.
22. S. Inokuchi and Y. Mizoguchi. Generalized partitioned quantum cellular automata and quantization of classical CA. *International Journal of Unconventional Computing*, 1:0312102, 2005.

23. I. G. Karafyllidis. Definition and evolution of quantum cellular automata with two qubits per cell. *Journal reference: Phys. Rev. A*, 70:044301, 2004.

24. S. Lloyd. A potentially realizable quantum computer. *Science*, 261(5128):1569–1571, 1993.

25. S. Lloyd. A theory of quantum gravity based on quantum computation. Arxiv preprint quant-ph/0501135, 2005.

26. P. Love and B. Boghosian. From Dirac to Diffusion: decoherence in Quantum Lattice gases. *Quantum Information Processing*, 4:335–354, 2005.

27. N. Margolus. Physics-like models of computation. *Physica D: Nonlinear Phenomena*, 10(1-2), 1984. partitioning, neighbourhood, energy.

28. N. Margolus. Parallel quantum computation. In *Complexity, Entropy, and the Physics of Information: The Proceedings of the 1988 Workshop on Complexity, Entropy, and the Physics of Information Held May-June, 1989, in Santa Fe, New Mexico*, page 273. Perseus Books, 1990.

29. J. Mazoyer. A Six-State Minimal Time Solution to the Firing Squad Synchronization Problem. *Theoretical Computer Science*, 50:183–238, 1987.

30. J. Mazoyer and I. Rapaport. Inducing an order on cellular automata by a grouping operation. In *Proceedings of STACS’98*, in *Lecture Notes in Computer Science*, pages 116–127. Springer, 1998.

31. D. A. Meyer. From quantum cellular automata to quantum lattice gases. *J. Stat. Phys.*, 85:551–574, 1996.

32. E. F. Moore. The firing squad synchronization problem. In *in Sequential Machines, Selected Papers*, pages 213–214. Addison-Wesley, 1964.

33. D. Nagaj and P. Wocjan. Hamiltonian Quantum Cellular Automata in 1D. Arxiv preprint arXiv:0802.0886, 2008.

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

35. N. Ollinger. The Quest for Small Universal Cellular Automata. In *Proceedings of ICALP ’02, Lecture Notes in Computer Science*, pages 318–329. Springer, 2002.

36. N. Ollinger. Universals in cellular automata a (short) survey. In B. Durand, editor, *First Symposium on Cellular Automata ”Journées Automates Cellulaires” (JAC 2008)*, Uzès, France, April 21-25, 2008. *Proceedings*, pages 102–118. MCCME Publishing House, Moscow, 2008.

37. N. Ollinger and G. Richard. A Particular Universal Cellular Automaton. In Turlough Neary, Damien Woods, Anthony Karel Seda, and Niall Murphy, editors, *CSP*, pages 267–278. Cork University Press, 2008.

38. J. P. Paz and W. H. Zurek. Environment-induced decoherence and the transition from quantum to classical. *Lecture Notes in Physics*, pages 77–140, 2002.

39. R. Raussendorf. Quantum cellular automaton for universal quantum computation. *Phys. Rev. A*, 72(022301), 2005.

40. B. Schumacher and R. Werner. Reversible quantum cellular automata. Arxiv pre-print quant-ph/0405174, 2004.

41. D. J. Shepherd, T. Franz, and R. F. Werner. A universally programmable quantum cellular automata. *Phys. Rev. Lett.*, 97(020502), 2006.

42. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, pages 303–332, 1999.

43. G. Theyssier. Captive cellular automata. In *Proceedings of MFCS 2004*, in *Lecture Notes in Computer Science*, pages 427–438. Springer, 2004.
44. T. Toffoli. Computation and construction universality of reversible cellular automata. *J. of Computer and System Sciences*, 15(2), 1977.
45. T. Toffoli and N. Margolus. *Cellular Automata Machine A New Environment for Modelling*. MIT Press, Cambridge MA, 1987.
46. J. Twamley. Quantum cellular automata quantum computing with endohedral fullerenes. *Phys. Rev. A*, 67(5):52318–52500, 2003.
47. W. Van Dam. Quantum cellular automata. Master thesis, University of Nijmegen, The Netherlands, 1996.
48. K. G. H. Vollbrecht and J. I. Cirac. Reversible universal quantum computation within translation-invariant systems. *New J. Phys Phys Rev A*, 73:012324, 2004.
49. J. von Neumann. *Mathematical foundations of quantum mechanics*. Princeton University Press, 1955.
50. J. Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.
51. J. Watrous. On one-dimensional quantum cellular automata. *Complex Systems*, 5(1):19–30, 1991.
52. K. Wiesner. Quantum Cellular Automata. Arxiv preprint arXiv:0808.0679, 2008.