A Computational Role for the One-dimensional Lieb-Shultz-Mattis Spin Model

by

Michael Biafore

Submitted to the Department of Physics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1994

Author . . . .
Department of Physics
December 29, 1993

Certified by . . . .
Tommaso Toffoli
Principal Research Scientist
Thesis Supervisor

Accepted by . . . .
George F. Koster
Chairman, Physics Graduate Committee

A Computational Role for the One-dimensional Lieb-Shultz-Mattis Spin Model

by

Michael Biafore

Submitted to the Department of Physics
on December 29, 1993, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Regarding individual quantum-mechanical degrees of freedom as elements of a parallel computation, a simple but fully interacting quantum many-body system is analyzed in order to derive a fundamental quantum-mechanical limit to massively parallel computation.

Exact solutions for all eigenvalues and eigenvectors of an operator $\hat{\Gamma}$ that the characterizes parallel computational velocity are found. A symmetry between the Hamiltonian and $\hat{\Gamma}$ is described, and its consequences investigated. The largest eigenvalue of the computational velocity operator is found to scale with system size, N, as $\gamma_{\mathrm{max}} \sim 2N/\pi < N$, a result which implies the impossibility of using additional quantum-mechanical parallel processors to obtain the same ideal speedup as in the classical case. A variational argument shows that this is a special case of a more fundamental limit to parallel quantum computation.

Thesis Supervisor: Tommaso Toffoli

Title: Principal Research Scientist

For my parents,

Louis Pasco

and

Edith Margaret Biafore

Contents

Chapter 1

Introduction

"Information", as Rolf Landauer has put it, "is physical". But to go beyond this general observation to specific results, we must find ways to analyze informational processes, "computation", using the familiar tools of theoretical physics. This thesis investigates the problem of analyzing computation—especially computation in which the nature of the information-bearing degrees of freedom is quantum-mechanical—with tools drawn from scattering theory, density matrix theory and the quantum theory of magnetism.

The central result, from which the thesis takes its title, is the formulation of the first exactly solvable model of "quantum computation". More specifically, I show that there is a class of cellular automata which is both powerful enough computationally to simulate any possible computer and simple enough physically to be identified with an exactly solvable quantum spin model: the Lieb-Shultz-Mattis model[104, 117], sometimes known as the one-dimensional XY model. The exact solution lets us obtain a rigorous picture of how computational properties, like the computation rate, are related to physical properties, such as the excitation energy.

Together with the other results presented here, the proposed model falls within the domain of a small interdisciplinary field, the theory of "physical computation". Physical computation seeks to reconstruct the conventional theory of computation on a foundation of physically realistic models. Eventually, it is hoped that such models will permit the extension of physical principles to computational phenomena, much as the development of physical chemistry in the late nineteenth and early twentieth century saw the extension of thermodynamical principles (by van't Hoff, Ostwald and Arrhenius) and quantum-mechanical principles (by Pauling and Slater) to chemical phenomena.

In addition to this admittedly philosophical motivation, there is a considerably more prosaic one. The end of the era of computation based on the metal-oxide/semiconductor field-effect transistor (MOSFET) appears to be within sight[42, 134, 66, 11]. Consequently, many researchers are investigating the possibility of basing the next generation of computers on nanometer-scale devices whose operation depends directly on physical effects like resonant tunneling, ballistic electron transport and Coulomb blockade effects. By developing computational primitives with a more physical form, a form more compatible with the way we understand these effects, we can hope to facilitate the systematic search for ways to apply these effects to computing.

From a slightly broader perspective, this thesis can be viewed as advocating a certain class of lattice-gas[111, 56, 63] cellular automata as the most appropriate vehicle for describing computation as a physical process. Members of this class are referred to as "few-body automata" because the primitive computational operation is identified with the scattering of a few particles at the vertices of the lattice-gas. The few-body model is based on concepts derived from the theory of quantum N-body scattering and the theory of reduced N-body density matrices. Few-body automata can be applied when the physical interactions are "regular" in the sense used in scattering theory. That is, few-body automata apply to interactions sufficiently nonsingular that there is some finite time far enough before and after the interaction that the evolution is effectively the "free" evolution. Not all interactions fall in this class; in particular, the Coulomb interaction does not.

It is the few-body model that provides, among other things, the route to formulating the first physically realistic model of quantum computation—computation in which the information-bearing degrees of freedom obey the Schrödinger equation. Unlike previous models of quantum computation, which have been roundly criticized by Landauer[96] and others for their Rube-Goldberg-like Hamiltonia, the model proposed here has a form familiar to physicists. Under very special conditions, the Hamiltonian of the $1D$ XY model, to which our model is very closely related, may even describe experimentally accessible systems. About twenty years ago, it was shown experimentally that the $1D$ XY model accurately describes the low-temperature heat capacity and transverse electric susceptibility of praseodymium trichloride[65] and praseodymium ethyl sulphate[52].

The plan of this thesis is described in more detail in §1.2. Since this thesis falls within the domain of the relatively unknown theory of physical computation, we begin in §1.1 with a brief history of the theory of physical computation. That history began when physicists like P. Bridgman and R. Landauer first questioned the physical realizability of certain operations that the conventional theory of computation has borrowed from abstract set theory.

1.1 History of physical computation

Set theory first attracted the skepticism of physicists in 1934, when P. W. Bridgman[25], better known for his experimental work on the properties of matter at very high pressures, wrote an extensive article criticizing set theory on the grounds that it was implicitly based on physically unrealizable procedures. Just two years later, Church and Turing made some of these same procedures[1] as the basis for the modern theory of computation. But Bridgman did more than object, he offered a physically-motivated alternative. By carefully considering the physical operations involved in determining which objects in the universe belonged to a given set, Bridgman was able to resolve some of the famous paradoxes of set theory, such as Russell's paradox[2]. Although Bridgman's method of resolving these paradoxes was ignored by workers in the theory of computation, his "operationalist" epistemology[24]—basically, the idea that all meaningful physical concepts must ultimately be clearly defined in terms of experimental operations—has so thoroughly pervaded physics that we take it for granted.

[1] In particular, one such procedure on infinite sets, known as "diagonalization[73, 103]" is used extensively in the theory of computation.

[2] Does the set that contains all sets not containing themselves contain itself, or not?

For thirty years, no serious attempts were made to reconcile operationalist physical theory with non-operationalist computation theory, until Landauer pointed out the consequences of this inconsistency. Landauer[94] advocated (and continues to advocate[98]) the need for what he calls a "self-consistent theory" in which, on the one hand, physical laws are expressed only in terms of computationally executable operations and on the other, the theory of computation is expressed only in terms of operations that are physically realizable.

This thesis addresses only the latter half of Landauer's program, which calls for bringing the theory of computation into the scope of systems treatable by theoretical physics. There are many precedents for the historical process by which a discipline that initially developed its own free-standing conceptual framework gradually acquires a foundation linked to theoretical physics. A particularly relevant example can be found in the development of physical chemistry during the last quarter of the nineteenth century.

To some extent, the analogous development of physical computation has already begun. Substantial progress has been made in understanding the thermodynamics of computation[19, 17]. In 1961, Landauer[93] proved wrong the widely held belief that each binary logical operation required the dissipation of energy $kT \ln 2$. He correctly identified the locus of dissipation as logically irreversible operations, those that compress the phase space of the information-bearing degrees of freedom and hence increase the entropy of the remaining non-information-bearing degrees of freedom. Because all the logical primitives in common use at the time had $N \geq 2$ inputs but only a single output, it was structurally impossible for them to be logically reversible. The single binary output could not possibly carry enough information to reconstruct the input. Logically reversible operations with N inputs and N outputs were known, the simplest nontrivial example being the 2-input/2-output operation that exchanges its inputs. However, it was not evident that universal computation could be performed using only reversible operations. In fact, it was widely believed that at least some irreversible operations would be required.

About a decade after Landauer's pioneering paper, Bennett[16] exhibited a reversible Turing machine, thereby showing that completely reversible computation was possible. Bennett[17, 18, 107] also gave the definitive account of Maxwell's Demon, showing that the demon incurs a thermodynamic cost in recording velocity information about the particle he is currently deciding to block (or not). The cost is incurred at the point where the demon, having finite memory, must eventually overwrite (i.e. erase) information pertaining to previous particles. The resulting increase entropy is sufficient to offset the 'paradoxical' decrease in entropy due to the demon's intervention.

E. Fredkin arrived at his interest in reversible logic from a different point of departure. Driven by the conviction that all physics was at root a computational phenomenon, he began searching for universal logic gates that possessed the nearly ubiquitous physical property of microscopic reversibility. The simplest such gate, having 3-inputs and 3-outputs, is now known as the Fredkin gate[54].

Fredkin also saw that if one wants to find models of computation that resemble physics as closely as possible, it is natural to start with cellular automata. Cellular automata were introduced around 1950 by S. Ulam[146] and J. von Neumann[147], and independently by K. Zuse[158]. Cellular automata, like many physical systems and quite unlike other models of computation, are spatially uniform. This raised a question—"Is it possible to combine the reversibility and computation-universality of Bennett's Turing machine with the spatial locality of a cellular automaton"? Initially, it was believed that this had been proven impossible, until Toffoli[142] exhibited the first computation-universal, reversible cellular automaton.

Fredkin[54] also realized that physics and computation could be brought closer not only by endowing models of computation with physical properties (like reversibility), but by endowing physical models with computational properties (like computation-universality). He showed that—with appropriate boundary conditions and initial conditions—a classical gas of infinitely hard billiard balls could be interpreted as a universal computer. In the course of implementing Fredkin's billiard ball model as a cellular automaton, Margolus[111] introduced a new kind of cellular automaton, which he called partitioning automaton.

Although special cases of partitioning cellular automata had been reinvented numerous times, particularly by Monte Carlo investigators[71, 72] and in the early lattice gas papers by J. Hardy, O. De Pazzis and Yves Pomeau[63] and U. Frisch, B. Hasslacher and Yves Pomeau[56], Margolus[111] was the first to recognize them as a separate class of cellular automata (CA) whose form made it particularly straightforward to incorporate conservation laws of all sorts, not just the momentum conservation laws of interest to lattice gas theorists, the fermion commutation relations of lattice QED, or the fine-grained entropy conservation that leads to reversible cellular automata.

Like conventional (von Neumann/Ulam/Zuse) cellular automata, partitioning cellular automata incorporate the notion of distance between sites only through the drastically simplified relation of adjacency. But, as Landauer[96] has remarked,

> *Real particles have an interaction that falls off with distance, but is not all that selective, and is not limited to nearest neighbors or next nearest neighbors.*

In this thesis, we bring models of computation a step closer to being treatable within the scope of physics by defining an extension of partitioning cellular automata called *few-body automata*. Few-body automata address Landauer's objection by replacing the stylized nearest-neighbor interactions of cellular automata with realistic physical interactions among a few conserved particles. In particular, we use the few-body form to show in Chapter 5, *Parallel Quantum Computation*, that one can construct a physically-realistic model of quantum computation. The resulting model is realistic in the sense that it is closely related to the one-dimensional XY spin model of Lieb, Shultz and Mattis. Their model, in turn, has been successfully applied to predicting the low-temperature heat capacity and electric susceptibility of praseodymium trichloride and praseodymium ethyl sulphate.

1.2 Plan

The central theme that has guided us in defining few-body automata (§2.6) is the analogy between a reversible N-input/N-output logic gate and the quantum scattering matrix ($\widehat{S}$ matrix) pertaining to N conserved particles, where typically $N = 2, 3$ or 4. It is well-known that the $\widehat{S}$ matrix formalism is valid only if the interaction potential satisfies certain regularity conditions at $r = 0$ and $r = \infty$. In §2.1, we use similar arguments to identify sufficient conditions on the interactions under which the few-body automaton abstraction—that is, the analogy between N-input/N-output logic gates and the $\widehat{S}$ matrix—is physically valid.

Eventually, we hope that such analogies will play a role in the development of physical computation as a subdiscipline, much as van't Hoff's analogy between dilute chemical solutions and mixtures of ideal gases guided early work in physical chemistry.

In response to the model presented in Chapter 5, Landauer[3] has raised the question of whether N-body automata with $N = 2$ are too restricted to be capable of universal-computation. In §2.8, I show that 2-body automata are indeed capable of universal computation, and that in fact no more than 5.81 bits of state (i.e., 56 states) per site are required. I also show that if Wolfram[151] is correct in conjecturing that rules with Wolfram codes 357 and 824 are computation-universal, then 3.58 bits of state will suffice.

In Chapter 3, *Logical Completeness*, we make a brief excursion into the theory of logical completeness, a part of classical circuit theory that seeks to determine whether or not a circuit element or binary function is "universal"—as the familiar NAND gate is—in the sense that all other binary functions can be composed from it. We show how the few-body format, in conjunction with graphs of a certain partial order on input states, can be used to determine whether or not the physical interaction governing the evolution of a particular few-body automaton has the two logical properties—nonmonotonicity and nonlinearity—that are prerequisites of logical completeness.

In Chapter 4, *Inverse Quantum Scattering*, we show how restricting ourselves to

[3] Personal email communication.

the few-body form permits us, at least in principle, to start from the logical function we want and analytically derive the form of interaction potential that will provide it.

Compared to the amount of progress made in putting the theory of computation on a sound thermodynamic footing, substantially less progress has been made in viewing computation as a quantum mechanical process. Benioff[13, 14] presented the first model of quantum computation by describing a time-dependent Hamiltonian operating on spin-$\frac{1}{2}$ particles that represented Boolean variables. The time-evolution of the system simulated the operation of a computation-universal Turing machine. Similar models, but also requiring time-dependent Hamiltonians, were elaborated on by Deutsch[44], Zurek[156] and Peres[131]. A notable improvement was made by Feynman[51], who gave the first time-independent Hamiltonian model of quantum computation. But none of the models are very satisfactory from the point of view of providing a model of quantum computation that is physically-realistic, because the resulting Hamiltonians are quite contrived and therefore bear little resemblance to actual physical systems.

Bennett has stressed the "...unphysical idealizations in existing models and the importance of specifying a quantum computer more concretely than by merely inventing a Hamiltonian[19]." Landauer has been even more critical of the physical relevance of these models:

> *In describing quantum computers we have felt free to invent arbitrary Hamiltonians, without the need to explain how the physical universe permits the construction of these Hamiltonians. Classical mechanics is accompanied by a tradition which does not encourage discussion of phase space mappings in such a sweeping way; we feel the need to draw pictures of the pieces, as if we were writing a patent disclosure*[96].

In an effort to make a more realistic model of quantum computation, Margolus[112] applied Feynman's method[51] of producing a time-independent Hamiltonian to the two-dimensional billiard ball cellular automaton, and succeeded in providing a model of quantum computer in which the Hamiltonian is both time-independent and short-ranged. Unfortunately, the rate at which useful computation occurs is diffusive rather than linear—that is, one has to wait a time N^2 rather than N before there is a reasonable probability that the quantum automaton will have made N forward steps in the computation. Because diffusive behavior severely limits the length of practical computations, it is questionable as to whether we can meaningfully assert that such computation is universal. Some progress on this problem was made in a one-dimensional analog of this quantum cellular-automaton. In [115], Margolus and I were able to show that a computation-rate operator, as defined in [112], commutes with the Hamiltonian. This means that the computation-rate is a constant of the motion, but we were not able to explicitly compute any eigenvalues of the computation-rate operator in this model.

Chapter 5, *Parallel Quantum Computation*, contains the result from which this thesis takes its title. In that chapter, I present an exactly solvable one-dimensional quantum cellular automaton with a Hamiltonian related to the one considered in [115]. All of the interesting long-range properties of the system can be described by an effective Hamiltonian $\widehat{H}_{\text{sync}}$ for the subsystem responsible for synchronizing communication between the cells. The computation within each cell is still performed by a contrived subsystem—but a very small one. The specific structure of this subsystem (see §2.8) is dictated by computational considerations, and is of less interest physically than the system described by $\widehat{H}_{\text{sync}}$.

Remarkably, $\widehat{H}_{\text{sync}}$ turns out to be identical to the Hamiltonian of a certain anisotropic Heisenberg antiferromagnet first solved by Lieb, Shultz and Mattis[104], sometimes referred to as the 1D XY model. In addition, experimental measurements of the heat capacity C_V and transverse electric susceptibility $\chi_\perp$ of praseodymium trichloride and praseodymium ethyl sulphate by D. R. Taylor, D. J. Thouless et al.[65] and J. T. Folinsbee, J. P. Harrison et al.[52] agree well with theoretical predictions[82, 27] derived from the 1D XY model. Consequently, this model of quantum computation begins to answer Landauer's objection that models of quantum computation rely on Hamiltonians so contrived that they lose all physical relevance:

After all, parts are obtained from a stockroom, or from a particle physicist.

> *They are not just degrees of freedom whose time-evolution is controlled by a cleverly chosen Hermitean operator. Most physicists would very likely be satisfied by a Hamiltonian which has a sufficiently familiar and simple structure, e.g. a Hamiltonian for a reversible cellular automaton which resembles an Ising lattice Hamiltonian. But it really is a very drastic assumption, about the real world, to assume that even simple Hermitean operators are* physically realizable *Hamiltonians.*(Landauer[99]).

About the time Benioff was beginning to explore spin models of quantum computation, interest in computing with microscopic degrees of freedom was increasing in the engineering community as well. In fact, although we have thus far emphasized the importance we attach to bringing the theory of computation within the scope of physical theory, it is only fair to mention that most of the recent impetus—or more candidly, the funding—comes from sponsors with significantly more practical concerns. The expectation is that as we begin to understand computation as a physical process, particularly in nanometer scale structures where quantum effects are important, we will be better able to understand whether or not these phenomena can be used as the basis of a new generation of computers.

Responding to perceived limits in downscaling conventional microprocessor architectures [11, 42, 134], a number of device physicists [10, 9, 50, 130] have proposed an alternative architecture based on cellular automata. Specifically, they envision dense arrays of nanometer-scale devices communicating with one another via *direct physical interactions* rather than through wired interconnections. The devices might consist of semiconductor heterostructures, molecular switches, or some other nanometer-scale structure capable of changing state in response to interactions with neighboring structures. For certain interactions among cells, the array might be made to function as a cellular automaton in which each device acts as a single cell.

Various groups have proposed that devices be coupled via resonant quantum tunneling[10, 9, 48], Coulomb interaction effects[105, 130], or soliton switching in π-conjugated polyenes[28]. If such proposals are ever realized, the resulting cellular automata machines would differ in an important way from the cellular automata machines constructed by Toffoli and Margolus[144, 145, 116]. The dynamics of the cellular automaton would no longer be determined by an arbitrarily definable lookup table; instead, it would be completely determined by the particular physical interactions[119] between cells. Consequently, we might expect few-body automata to be of some use in understanding how we might extract computation from arrays of such nanoscale devices.

As an exercise in applying few-body automata, Chapter 6, *Ballistic Computation in an Array of Quantum Dots*, describes a 4-body automaton that simulates the ballistic computation of the billiard ball cellular automaton[111, 113] in an array of idealized quantum dots. The resulting "gedanken" computer provides a computation-universal cellular automata architecture for the single-electron quantum-dot device proposed by Obermayer *et al.*[130, 129, 140, 128, 108]. For a number of practical reasons, the resulting construction is not easily amenable to error-correction (or even fabrication), but it serves well enough as a simple example of how few-body cellular automata might be applied to more realistic nanometer scale devices.

Chapter 2

Few-body Automata

In this chapter we define N-body automata, determine the conditions under which they are physically meaningful, and prove that they can simulate all other cellular automata, including computation-universal ones, at a reasonable cost.

The definition of few-body automata depend on the analogy alluded to in §1.2 between the vertex rule $\mathcal{S} : Z_2^N \to Z_2^N$ that maps incoming to outgoing states of a lattice gas (Fig. 2-1a shows the case $N = 3$) and an N-particle scattering matrix $\widehat{S}$ (Fig. 2-1c). The computational primitive is therefore the vertex rule $\mathcal{S}$, a reversible logic gate f with N inputs and N outputs (Fig. 2-1(b)).

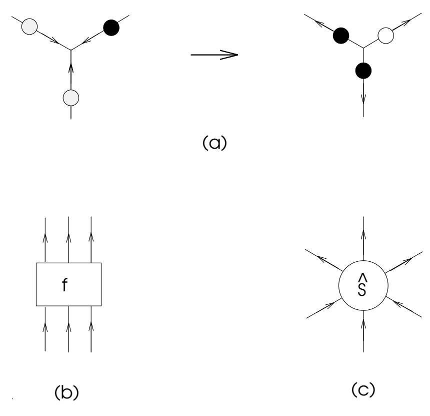

Figure 2-1: (a) Lattice gas vertex with three incoming and three outgoing particles, (b) Reversible three-input/three-output logic gate, (c) Three-particle $\widehat{S}$-matrix.

In §2.1 and §2.2 we determine the physical conditions under which this computational primitive is physically well-defined. In §2.3, we show how a regular pattern of interactions at the vertices combine to give the structure of few-body automata. Finally, in §2.6 and the remainder of the chapter, we give the formal definition of few-body automata and ascertain the expense, in time and space, at which they can simulate all other cellular automata, including those that are computation universal. We thereby establish that, in the course of giving few-body automata a form closer to that of physical phenomena, we have not sacrificed their ability to compute. This result is used implicitly throughout the remainder of the thesis, particularly in Chapter 5, where we appeal to the existence of the universal 2-body automata constructed in §2.8 and dispense with any further discussion of its detailed structure.

2.1 Asymptotic States

The notion of an N-input/N-output gate is a computational abstraction whose utility depends on the assumption that arbitrary networks of such gates can be composed by letting the output of one gate serve as a well-defined input to the next. Physically, this corresponds to the assumption that the outputs have attained an *asymptotic state*—a state that obtains sufficiently long before or after an interaction that its evolution is indistinguishable from the free evolution. We will find that this assumption, reasonable as it seems, is only valid if the physical interactions going on inside the gate satisfy certain regularity conditions. The appropriate methods for determining the regularity conditions under which asymptotic states exist have been developed in nonrelativistic quantum scattering theory.

Consider a system of N particles governed by the nonrelativistic Schrödinger equation with Hamiltonian $H = H_0 + V$, where V is some unspecified interaction. A state $|\psi_{in}\rangle$ of the corresponding Hilbert space $\mathcal{H}$ is said to be the *in-asymptote* of another state $|\psi\rangle$ if

$$\lim_{t\to-\infty} U(t)|\psi\rangle = \lim_{t\to-\infty} U_0(t)|\psi_{\text{in}}\rangle, \tag{2.1}$$

where $U(t)$ is the time-evolution operator of the fully interacting system and $U_0(t)$ is the time-evolution operator for the free Hamiltonian. Similarly, a state $|\psi_{out}\rangle$ of the

Hilbert space is said to be the *out-asymptote* of $|\psi\rangle$ if

$$\lim_{t\to\infty} U(t)|\psi\rangle = \lim_{t\to\infty} U_0(t)|\psi_{out}\rangle. \tag{2.2}$$

In order to understand what the computational abstraction of the vertex rule $\mathcal{S}$ means physically, we would like to find an operator $\widehat{S} : \mathcal{H} \to \mathcal{H}$ that maps the in-asymptotes to the out-asymptotes, but without explicit reference to what goes on in between, just as the cellular automaton rule $\mathcal{S}$ maps the N input states at each vertex of a lattice gas to the N output states. In quantum scattering theory the operator $\widehat{S}$, known as the *scattering matrix*, is defined as the product of two auxiliary operators

$$\widehat{\Omega}_\pm \equiv \lim_{t\to\mp\infty} U^\dagger(t)U_0(t), \tag{2.3}$$

called Møller operators. The Møller operators $\widehat{\Omega}_+$ and $\widehat{\Omega}_-$ transform the asymptotic-in and asymptotic-out states $|\psi_{\text{in}}\rangle$ and $|\psi_{\text{out}}\rangle$ into the actual states $|\psi_{\text{in}}\pm\rangle$ of the fully-interacting system that have $|\psi_{\text{in}}\rangle$ and $|\psi_{\text{out}}\rangle$ as their in- and out-asymptotes:

$$\widehat{\Omega}_+|\psi_{\text{in}}\rangle = |\psi_{\text{in}}+\rangle, \tag{2.4}$$

$$\widehat{\Omega}_-|\psi_{\text{out}}\rangle = |\psi_{\text{in}}-\rangle. \tag{2.5}$$

The limit (2.3) that defines the Møller operators only exists if the interaction potential is sufficiently non-singular. Below, we derive sufficient conditions. When these are satisfied, asymptotic states exist and the N-input/N-output abstraction is physically valid. To avoid a proliferation of $\pm$ and $\mp$ indices, consider the case of $\widehat{\Omega}_-$; the conditions for the existence of $\widehat{\Omega}_+$ are the same.

The Møller operator $\widehat{\Omega}_-$ exists only if, for all $|\psi_{\text{out}}\rangle \in \mathcal{H}$, the limit

$$\lim_{t\to\infty} U^\dagger(t)U_0(t)|\psi_{\text{out}}\rangle \tag{2.6}$$

"converges in the norm"[1] to to some $|\psi\rangle$ in the Hilbert space. To convert the question

[1]A sequence of vectors $\phi(t)$ in a Hilbert space is said to converge in the norm to ϕ_0 if $\lim_{t\to\infty} ||\phi -$

of convergence in Hilbert space to one in the reals, express this operator as the integral of its derivative

$$\widehat{U}^{\dagger}(t)\widehat{U}_0(t)|\psi_{\text{out}}\rangle = |\psi_{\text{out}}\rangle + i\int_0^{\infty} \widehat{U}^{\dagger}(t')\widehat{V}\widehat{U}_0(t')|\psi_{\text{out}}\rangle dt', \tag{2.7}$$

where $\widehat{V} = \widehat{H} - \widehat{H}_0$.

The limit (2.6) then exists if the integral converges, which is certainly the case if the integral is absolutely convergent

$$\int_0^{\infty} \|\widehat{U}^{\dagger}(t')\widehat{V}\widehat{U}_0(t')|\psi_{\text{out}}\rangle\| dt' = \int_0^{\infty} \|\widehat{V}\widehat{U}_0(t')|\psi_{\text{out}}\rangle\| dt' < \infty, \tag{2.8}$$

the last equality following from the unitarity of $\widehat{U}^{\dagger}$. For two-body interactions, (2.8) can be established in a basis composed of Gaussians

$$\langle\vec{r}|\psi_{\text{out}}\rangle = e^{-(\vec{r}-\vec{r_0})^2/2\sigma^2} \tag{2.9}$$

in the relative coordinate $\vec{r}$. The norm of the integrand in (2.8) satisfies

$$\|\widehat{V}\widehat{U}_0(t')|\psi_{\text{out}}\rangle\|^2 = \int |V(\vec{r})|^2 |\langle\vec{r}|\widehat{U}_0(t')|\psi_{\text{out}}\rangle|^2 d\vec{r}. \tag{2.10}$$

The second factor is just the modulus of a freely propagating Gaussian

$$|\langle\vec{r}|\widehat{U}_0(t')|\psi_{\text{out}}\rangle|^2 = \left(\frac{m^2\sigma^4}{t'^2 + m^2\sigma^4}\right)^{3/2} e^{-(\vec{r}-\vec{r}_0)^2/(\sigma^2 + \frac{t'^2}{m^2\sigma^2})}; \tag{2.11}$$

so

$$\|\widehat{V}\widehat{U}_0(t')|\psi_{\text{out}}\rangle\|^2 \leq \left(\frac{m^2\sigma^4}{t'^2 + m^2\sigma^4}\right)^{3/2} \left(\int |V(\vec{r})|^2 d\vec{r}\right), \tag{2.12}$$

and therefore (2.8) converges if the integral over the squared potential is finite.

But the contribution to

$$\int |V(\vec{r})|^2 d\vec{r} \sim \int |V(\vec{r})|^2 4\pi r^2 dr \tag{2.13}$$

$\phi_0\| \to 0$.

near $r = 0$ will be finite if

$$\lim_{r\to 0} r^2 |V(r)|^2 < r^{-1+\epsilon} \tag{2.14}$$

for some $\epsilon > 0$. Similarly, the contribution to (2.13) for $r \to \infty$ will be finite if

$$\lim_{r\to \infty} r^2 |V(r)|^2 < r^{-1-\epsilon} \tag{2.15}$$

for some $\epsilon > 0$. Therefore, for the Møller operators to exist, it is sufficient[2] that the potential satisfies the regularity conditions

$$\lim_{r\to 0} r^{\frac{3}{2}-\epsilon} V(r) = 0 \tag{2.16}$$

and

$$\lim_{r\to \infty} r^{\frac{3}{2}+\epsilon} V(r) = 0. \tag{2.17}$$

Potentials which are weak enough that the scattered states attain a well-defined asymptotic value are called *regular*, as distinguished from singular, potentials. Not every physically reasonable interaction satisfies these regularity conditions. For example, the unscreened attractive Coulomb potential admits no asymptotic states. This is manifested, for example, in the fact that its phase shifts[139, p.261],

$$\delta_l = -\ln(2kr) + \arg\Gamma(l + 1 + i\frac{me_1e_2}{k}), \tag{2.18}$$

which characterize the effect of the scattering interaction on states with angular momentum l, are not well-defined since they do not approach a limit as $r \to \infty$. That is, the interaction is so strong that, at least by the definition of "free" in (2.3), the particles *never* behave freely. Already, we see that the physical assumptions implicit in the computational abstraction of an N-input, N-output gate are by no means trivial: some interactions do not support the asymptotic states on which the abstraction is based. But the validity of the N-input/N-output abstraction depends on more than

[2]But not necessary; conditions that are both necessary and sufficient are not known. Simon[136] has shown that (2.22) can be weakened somewhat. Asymptotic states can still be defined even when $V(r)$ is as singular as $r^{-2+\epsilon}$ near $r = 0$.

the existence of asymptotic states; it also requires that the system be free of pathological states having one kind of asymptote (in or out), but not the other. When this pathology is absent, the system is said to be *asymptotically complete.*

2.2 Asymptotic Completeness

If the ranges $\mathrm{Ran}(\Omega_+)$ and $\mathrm{Ran}(\Omega_-)$ of the Møller operators are identical subsets of the full Hilbert space, the system is said to have the property of *asymptotic completeness.* When this condition holds, the Møller operator $\widehat{\Omega}_-$ has a well-defined inverse—equal to $\widehat{\Omega}_-^\dagger$, since $\widehat{\Omega}_-$ is norm-preserving—on the range of $\widehat{\Omega}_+$. As a consequence, it is possible to define an $\widehat{S}$-matrix

$$\widehat{S} \equiv \widehat{\Omega}_-^\dagger \widehat{\Omega}_+. \tag{2.19}$$

The $\widehat{S}$ matrix connects the asymptotic-in and asymptotic-out states,

$$\widehat{S}|\psi_{\mathrm{in}}\rangle = |\psi_{\mathrm{out}}\rangle, \tag{2.20}$$

without explicit reference to the intervening interaction. Asymptotic completeness is also required in order to establish that $\widehat{S}$, as defined in (2.19), is unitary [3].

Proving a system asymptotically complete is notoriously difficult. For the case of three-body scattering, L. D. Fadeev[47] was able to establish asymptotic completeness when all pairwise interactions V_{ij} of the three particles satisfy the regularity condition (2.16) at $r = 0$, in addition to the more stringent condition

$$\lim_{r\to\infty} r^{3+\epsilon} V_{ij}(r) = 0 \tag{2.21}$$

at $r = \infty$. More recently, Hepp[70] has shown that, under the same conditions, the general N-body case—the case we associate with an N-input/N-output logical gate—is also asymptotically complete.

[3] In the axiomatic S-matrix theory of strong interactions[33, 74], no such proof is required because the existence and unitarity of $\widehat{S}$ is assumed.

Consequently, we see that the practice of breaking computation down into a composition of functions (or gates) is not without physical limits. Even though the composition of an arbitrary binary function from more primitive binary functions is a mathematical tautology, from an operationalist perspective the decomposition is physically meaningless unless the interactions V_{ij} used to implement the primitive functions satisfy both

$$\lim_{r \to 0} r^{\frac{3}{2}-\epsilon} V_{ij}(r) = 0, \tag{2.22}$$

and

$$\lim_{r \to \infty} r^{3+\epsilon} V_{ij}(r) = 0. \tag{2.23}$$

Therefore, few-body automata, which are spatially regular compositions of N-body interactions, can only be realized by the class of physical interactions that satisfy these conditions. Within this class, our goal is to probe the latent computational properties—such as the computation-universality—of physical interactions.

2.3 Few-body Automata as a Probe

In order to probe the latent computational properties of an interaction, we must define an appropriate "probe", a probe which, when applied, stimulates the physical interaction to compute, but without introducing spurious effects that overwhelm and obscure the computational properties to be probed. Just as NMR (nuclear magnetic resonance) and related techniques let us probe the local magnetic properties of systems by applying appropriate constraints and driving forces consisting of static and time-dependent external fields, few-body automata are a way of probing the computational properties of an interacting system by imposing appropriate constraints on the pattern of interactions in space and time.

To understand what motivates the definition of few-body automata, suppose we are given a collection of conserved, interacting particles. Since they are conserved, but are assumed to have some nontrivial interaction, it is natural to view them as information-bearing degrees of freedom. Next, we would like to understand the conditions under which the interaction is sufficiently "rich" to provide the basis for a computation. In particular, suppose we want to understand the computational properties—that is, view as a computation—a physical process involving six conserved, interacting particles. In the absence of further assumptions about the interactions, we must regard this as an irreducible twelve-legged process, as in Fig. 2-2.

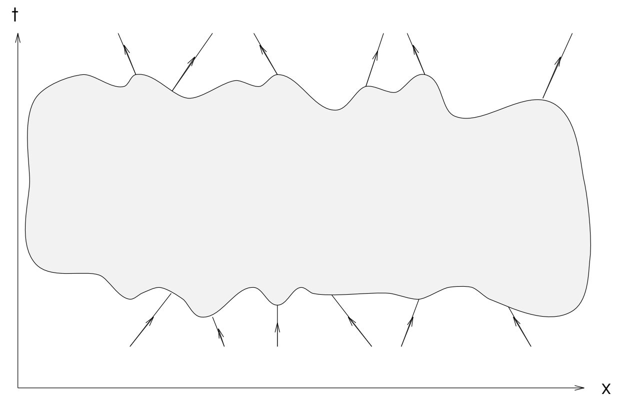

Figure 2-2: Generic process with six input states and six output states.

However, if the interactions are *regular*, we have seen that it is valid to view the six-particle process as a composition of few-body scattering events, as in Fig. 2-3.

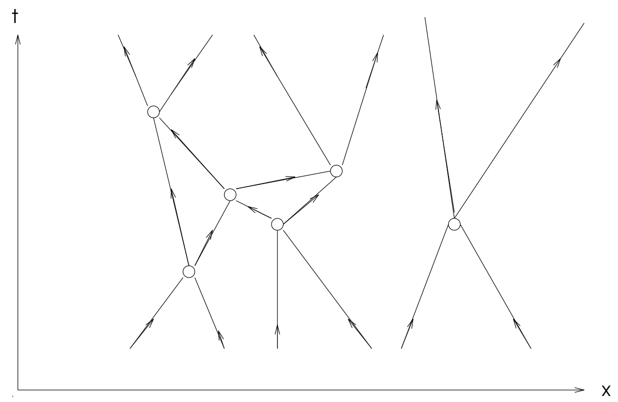

Figure 2-3: Process with six input states and six output states when all interactions are regular.

But the scattering events still occur too irregularly in space and time for us to analyze them using the tools available from the theory of lattice-gas cellular automata.

Instead, we discretize the times and places where events can occur, until the pattern of interactions resembles that in Fig. 2-4. This is now isomorphic to the space-time structure of a special kind of cellular automaton, a lattice gas[56, 63, 111].

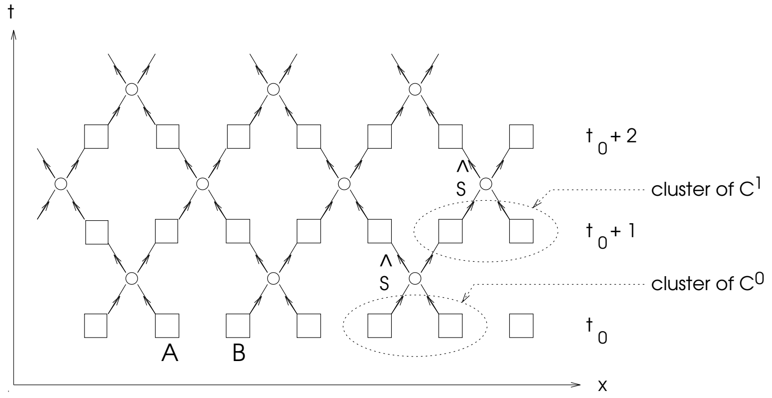

Figure 2-4: Interaction pattern of a 2-body CA in one dimension is isomorphic to that of a one-dimensional lattice gas.

In this thesis, I will only analyze computations that involve physical interactions satisfying the regularity conditions (2.22) and (2.23), and which conform to the space-time structure of a lattice-gas (or partitioning[111]) cellular automaton (see Fig. 2-4 and Fig. 2-5). Any physical implementation of a cellular automaton that satisfies these regularity and structural conditions is called a *few-body automaton.*

Except for the formal simulation theorems of §2.6–2.8, we confine our attention almost entirely to few-body automata in one or two dimensions. The pattern of interactions in one dimension is shown in Fig. 2-4 and the two-dimensional pattern in Fig. 2-5. In Fig. 2-5, the cells of the cellular automaton are represented as dotted boxes, each of which can take on a finite number of states. The open balls indicate the $\widehat{S}$ operator that transforms the four in-asymptotic states into four out-asymptotic states. Note that the four out-states become inputs to four different interactions at the next discrete time step. On the following time step, however, the original grouping recurs, so that the time dependence has two distinct phases. The alternating groupings are called *clusterings*, and are denoted $\mathcal{C}^0$ and $\mathcal{C}^1$ in the formal proofs of §2.6–2.8. In two dimensions, each clustering consists of $2{\times}2$ clusters of cells; in one dimension (Fig. 2-4), the clusters contain two cells.

In order to identify this pattern of interactions with a particular binary cellular automaton, someone must supply a mapping $\xi : \mathcal{H} \rightarrow \{0,1\}$ from states in the Hilbert space $\mathcal{H}$ to binary values. Following Landauer, we will not pursue the details of how information-bearing degrees are linked to the information they bear. Often the association is obvious. In schemes like Likharev's single-electron logic[5, 105], the presence of an electron is associated with the binary value 1, its absence with the binary value 0.

2.4 Cluster Separation and Liouville equation

As Landauer[96] has remarked, one of the unphysical features of cellular automata is that

> *Real particles have an interaction that falls off with distance, but is not all that selective, and is not limited to nearest neighbors or next nearest neighbors.*

That is, conventional cellular automata do not incorporate any notion of distance between cells, only adjacency. In contrast, N-body automata incorporate the notion of distance by setting a scale Λ for the separation between N-body clusters. More generally, in any physical realization of an N-body automaton, Λ would be determined by the range of the physical interaction between information-bearing degrees of freedom.

For example, although we have seen that the Coulomb interaction is not regular, the screened Coulomb, or Yukawa interaction, $e^{-r/\Lambda}/r$, is regular. The scale Λ, the Debye screening length, will be important to us in Chapter 6. There, it determines the required separation of clusters in the few-body automaton which was used to solve the problem of how to extract universal computation from Coulomb-induced shifts in the energy levels of neighboring quantum dots.

The simplest example of an N-body automaton is the one-dimensional 2-body automaton shown schematically in Fig. 2-6. The linear array of cells indicates one possible placement of cells in the x-y plane. The cells are to be thought of as some sort of interacting nanometer-scale structure. In the case shown, the cells take on one of three physical states, representing two logical states: 1 (solid circles), 0 (open circles), and a third, physically "unoccupied", state (crossed circles $\otimes$) that is computationally quiescent[4]. Possible physical states might be a soliton localized near a particular monomer in a polyacetylene chain[28], or a charge state localized on a metallic particle in a linear array of metal-insulator-metal (MIM) tunnel junctions[105]. A second, less fundamental, scale can also be defined if we let λ represent the diameter of a cluster.

Since N-body automata entail the evolution of mutually isolated clusters containing precisely N particles, the physical state of the cluster, which determines the corresponding computational state in the N-body cellular automaton, can be characterized by an N-body reduced density matrix.

Let $\widehat{\rho}^{(I)}$ denote the density matrix for the entire N-body automaton, the collection of all clusters. The equation of motion of $\widehat{\rho}^{(I)}$ is the quantum Liouville equation

$$i\hbar\frac{\partial}{\partial t}\widehat{\rho}^{(I)} = [\widehat{V}^{(I)}, \widehat{\rho}^{(I)}]. \tag{2.24}$$

where the potential

$$\widehat{V}^{(I)} = \sum_{c,i\neq j}(\widehat{V}^{(I)}_{\mathrm{ext},ci} + \widehat{V}^{(I)}_{ci,cj}) + \sum_{c\neq c',i,j}\widehat{V}^{(I)}_{ci,c'j} \tag{2.25}$$

contains terms describing the interaction of the site i in cluster c with an external field, interactions between sites i and j within c and a term describing the possibility of residual interactions between sites in different clusters c and c'.

For a particular cluster c_0, tracing over all degrees of freedom outside c_0 in the full quantum Liouville equation (2.24) yields the generalized master equation

$$i\hbar\frac{\partial}{\partial t}\widehat{\rho}^{(I)}_{c_0} = [\widehat{V}^{(I)}_{c_0}, \widehat{\rho}^{(I)}_{c_0}] + \mathrm{Tr}_{c\neq c_0 \text{ or } c'\neq c_0}\{[\sum_{c\neq c',i,j}\widehat{V}^{(I)}_{ci,c'j}\ ,\ \widehat{\rho}^{(I)}]\}. \tag{2.26}$$

[4]By *computationally quiescent*, we mean that this state "evolves" into itself.

This is the equation of motion for the reduced N-body density matrix $\widehat{\rho}_{c_0}^{(I)}$ that describes the state of the N-body cluster c_0.

Each term of (2.26) has direct computational significance. The term

$$\mathrm{Tr}_{c\neq c_0}\{[\sum_{c\neq c',i,j} \widehat{V}^{(I)}_{ci,c'j}\ ,\ \widehat{\rho}^{(I)}]\} \tag{2.27}$$

is an incoherent driving term. For ideal few-body automata, it vanishes since it is due to intercluster effects on the information-bearing degrees of freedom. More realistically, it must be treated as a noise source and its effects minimized through error correction.

In the ideal limit of no interaction with other clusters, the commutator $[\widehat{V}_{c_0}^{(I)}, \widehat{\rho}_{c_0}^{(I)}]$ determines the coherent evolution $\widehat{S}$ of the state of the cluster and hence the logical function $\mathcal{S}$ at the N-body vertex.

2.5 Formal Definition

Although the separation of few-body clusters is crucial to any physical implementation, in the remainder of this chapter we are concerned exclusively with the computational, rather than the physical properties of N-body automata. Specifically, we will define few-body automata formally—as mathematical objects like Turing machines—in order to obtain numerical estimates of the minimum number of states they require to achieve computation-universality. We need to do this in order to assure ourselves that this number is not so exorbitant that each cell of the few-body automaton would have to be physically enormous. Fortunately, we will find that this is not the case; but proof is fairly tedious, and the impatient reader may wish to glance at Tables 2.1 and 2.2, which summarize the results, and proceed to Chapter 3.

2.6 Few-body Automata in d Dimensions

We begin by focusing on the simulation of d-dimensional conventional CA by d-dimensional N-body automata where $N = 2^d$. These d-dimensional N-body automata employ two clusterings $\mathcal{C}^0$ and $\mathcal{C}^1$ directly analogous to those illustrated in Figs. 2-4 and 2-5. The clusters making up $\mathcal{C}^0$ and $\mathcal{C}^1$ consist of d-dimensional cubes with two cells along each edge.

$$\mathcal{C}^0 = \{\mathcal{C}_{2i_1,2i_2,\ldots 2i_d} | i_k \in Z\}, \tag{2.28}$$

and

$$\mathcal{C}^1 = \{\mathcal{C}_{2i_1+1,2i_2+1,\ldots 2i_d+1} | i_k \in Z\}, \tag{2.29}$$

where

$$\mathcal{C}^0_{i_1,i_2,\ldots i_d} = \{\sigma_{i_1+\tau_1,i_2+\tau_2,\ldots,i_d+\tau_d} | i_k \in Z\,, \tau \in \{0,1\}\}. \tag{2.30}$$

Definition 2.6.1 *A d-dimensional 2^d-*body *cellular automaton is a pair $\langle \widehat{S}, \widehat{Q} \rangle$, where $\widehat{Q}$ is the finite set of states that each cell can occupy and the automorphism $\widehat{S} : \widehat{Q}^{2^d} \rightarrow \widehat{Q}^{2^d}$ is a local evolution operator which synchronously updates each cluster of 2^d cells belonging to the clusterings $\mathcal{C}^0$ and $\mathcal{C}^1$.*

Our goal is to find least upper bounds on the number of states required to attain computation universality in the limiting case $N = 2$. The importance of this limit has two sources. First, the physics of two interacting devices is both experimentally the most accessible and theoretically the most tractable. Secondly, from a computational perspective the case of $N = 2$, where each cell has only a single neighbor aside from itself, is the most difficult limit in which to achieve computation universality. Therefore, the upper bounds we will obtain on the number of states required for $N = 2$ automatically represent upper bounds for all N-body automata with $N \geq 2$.

In addition, these bounds can be used to evaluate the practicality of using specific nanometer-scale devices as cells of a CA . Some of the proposed coupling mechanisms (such as single-electron tunneling[105]) are limited by the underlying physics

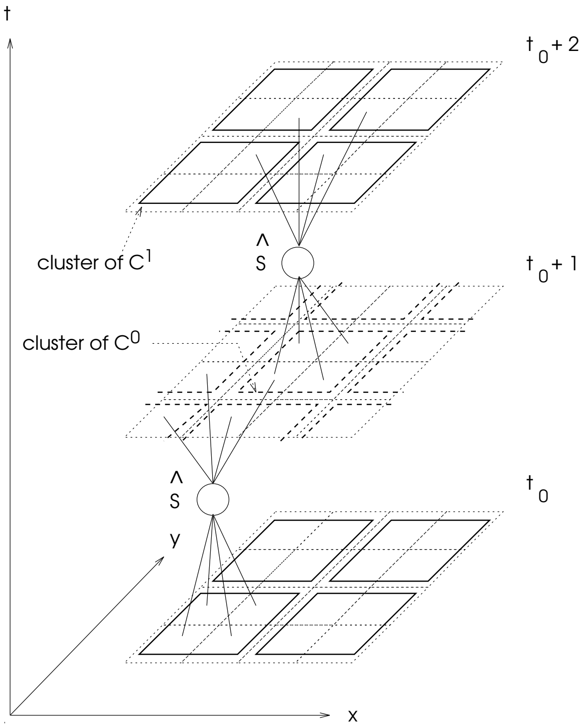

Figure 2-5: Partitioning cellular automaton with 2×2 cluster neighborhoods. Clusters of $\mathcal{C}^0$ are shown as solid squares; clusters of $\mathcal{C}^1$ are dashed; the cells are shown as dotted boxes.

to providing only a single bit per cell [21, 105, 130]; none of them can provide more than a few bits per cell. Consequently, although one can easily show that universal computation is possible even in the limit of 2-body cellular automata, the question of practical relevance is, "How many bits per cell will be required"?

To approach this question, in section 2.7 we establish a general theorem describing the efficiency with which a 2^d-body automaton in d-dimensions can simulate an arbitrary conventional cellular automaton (CA). By applying the simulation theorem to known computation-universal CA of conventional type, we find that it is possible to perform universal computation in a 2-body automaton with no more than 5.81 bits per cell. We then show how 4-body automata can be systematically simulated by 2-body automata. Using this technique, we are able to lower the bound further and obtain a universal 2-body automaton requiring just 2 bits per cell.

Simulation by Contraction Automata

We will show how any d-dimensional conventional cellular automaton $\langle U, Q, r\rangle$ can be simulated by a 2^d-body automaton $\langle \widehat{S}, \widehat{Q}\rangle$.

Conventional Cellular Automata

To begin, we define explicitly what we mean by "conventional" (or "von Neumann") cellular automata, and explain why they are not a satisfactory tool for analyzing the CA -like properties of arrays of interacting devices. Conventional CA [147] usually reside on a d-dimensional cartesian lattice, and we restrict our attention to this case. The neighborhood of a given cell is defined to consist of all cells within some radius r. (For example, in $d = 2$, the case $r = 1$ gives the well-known Moore neighborhood [145].) The next state of the cell then depends on an evolution operator U with domain $Q^{(2r+1)^d}$, where Q is the finite set of states each cell may occupy.

Definition 2.6.2 *On a d-dimensional rectangular lattice, a* conventional cellular automaton *with radius r is a triple $\langle U, Q, r\rangle$, where Q is the finite set of states that each cell can occupy, r is the radius of the neighborhood and $U : Q^{(2r+1)^d} \to Q$ is the local* *evolution operator,*

$$a^{t+1}_{i_1,i_2,\ldots,i_d} = U \begin{pmatrix} a^t_{i_1,i_2,i_3\ldots,i_d} \\ a^t_{i_1+1,i_2,i_3\ldots,i_d} \\ a^t_{i_1-1,i_2,i_3\ldots,i_d} \\ \vdots \\ a^t_{i_1+r,i_2,i_3\ldots,i_d} \\ a^t_{i_1-r,i_2,i_3\ldots,i_d} \\ \vdots \\ a^t_{i_1+r,i_2+r,i_3+r\ldots,i_d} \\ a^t_{i_1-r,i_2-r,i_3-r\ldots,i_d} \end{pmatrix}. \qquad (2.31)$$

Unfortunately, for the application device that physicists are proposing, where cells are coupled by real physical interactions (such as Coulomb dipole interactions [130, 21]), conventional CA are not a suitable model. Note that cellular automata are characterized by three forms of discreteness:

1. each cell can take on only a finite number of states,
2. the state of the cell changes at discrete time intervals and,
3. state of each cell at the following time-step is completely determined by a well-defined (usually small) subset of nearby cells, its “neighbors”.

If we simply arrange nanometer-scale devices in a dense rectangular array, as in a conventional cellular automata, then in general, properties 2 and 3 will be lost. Because the cells are coupled by physical interactions, state transitions occur continuously rather than at controlled discrete time-intervals. And although the effect of distant cells decreases with separation (e.g., the Coulomb dipole potential[78] falls off as $\sim 1/r^2$), in most cases the next state of each cell will depend on a large, rapidly fluctuating subset of all cells. The resulting dynamics may be of interest in its own right, but other than its spatial discreteness, it has little in common with cellular automata. Until we find a format to constrain the dynamics of such device arrays so that they have the essential discreteness of a cellular automaton, we will not be able to analyze their computational properties within the framework of cellular automata theory.

For certain kinds of interactions between cells, we can impose the required constraints by imposing the structure of *N-body* automata.

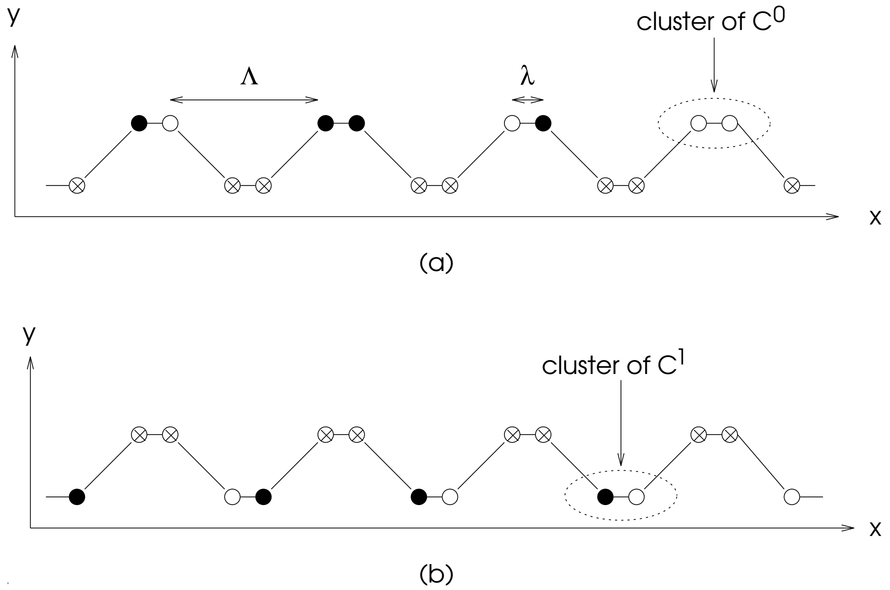

Figure 2-6: (a) One-dimensional 2-body automaton with cells of the $\mathcal{C}^0$ clustering occupied, solid cells indicate state $\sigma = 1$, open cells indicate $\sigma = 0$, crossed cells indicate the quiescent, or unoccupied, state. (b) 2-body automaton with cells of the $\mathcal{C}^1$ clustering occupied.

Contraction Automata

The simplest way to prove the simulation theorem is to introduce an auxilliary form of cellular automaton, the *contraction* cellular automaton, and first prove that any conventional CA can be simulated by a contraction automaton. Then we complete the proof by showing that any contraction automaton can be simulated by a 2^d-body cellular automaton.

Contraction automata are a straightforward generalization to higher dimensions of the *one-way* automata introduced by Culik [2]. Their sole function here is to systematically compress the neighborhood volume from the $(2r+1)^d$ cells of a conventional CA with radius r to the 2^d cells available to a 2^d-body automaton, trading off a reduction in the volume of the neighborhood for an increase in the size of the state set[75].

Definition 2.6.3 *A d-dimensional* contraction automaton *is a pair $\langle U, Q\rangle$ where Q is the state set and $U : Q^{2^d} \to Q$ is the local evolution operator,*

$$a^{t+1}_{i_1,i_2,\ldots,i_d} = U\begin{pmatrix} a^t_{i_1,i_2,i_3\ldots,i_d} \\ a^t_{i_1+1,i_2,i_3\ldots,i_d} \\ a^t_{i_1,i_2+1,i_3\ldots,i_d} \\ a^t_{i_1+1,i_2+1,i_3\ldots,i_d} \\ \vdots \\ a^t_{i_1+1,i_2+1,i_3+1\ldots,i_d+1} \end{pmatrix}. \tag{2.32}$$

Each cell has a neighborhood consisting of the 2^d cells having coordinates equal to or one greater than the corresponding coordinate of the cell. For $d = 1, 2$ and 3, the neighborhoods for contraction automata are shown in figure 2-7.

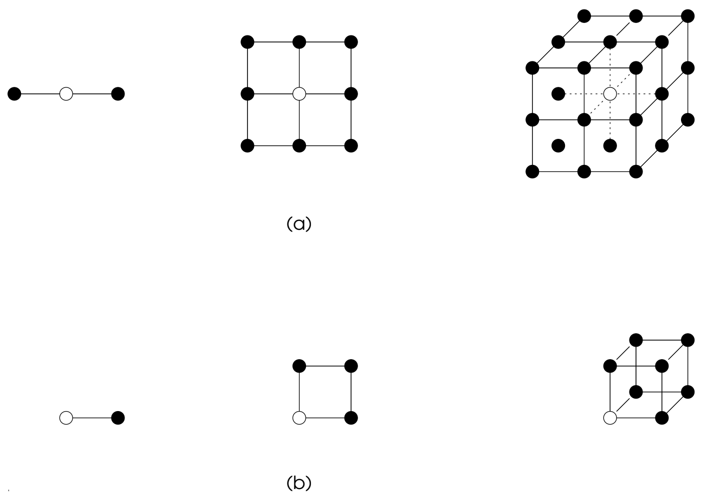

Figure 2-7: Comparison of the neighborhoods of the open cell in (a) conventional CA ($r = 1$ as shown) and (b) contraction automata for $d = 1, 2, 3$.

Lemma 2.6.1 *Any d-dimensional cellular automaton $\langle U, Q, r\rangle$ can be simulated by a d-dimensional contraction automaton $\langle U', Q'\rangle$. The simulation can be performed* *in the same amount of space with a slowdown of at most $2r$ and employs $|Q'| = \sum_{j=1}^{2r} |Q|^{j^d}$ states per cell.*

POINT: For the sake of concreteness, we present the construction for the case $d = 2$ and $r = 1$; generalization to any $d \geq 1$ and $r \geq$ is straightforward.

First, Q' is enlarged to include the tensor products of Q needed to encode a contracted representation of the $(2r+1)^d$ cells in the conventional CA neighborhood.

$$Q' \equiv Q \cup (\otimes_{i=1}^{2^d} Q) \cup \ldots \cup (\otimes_{i=1}^{(2r)^d} Q) = \bigcup_{j=1}^{2r} Q^{j^d} \tag{2.33}$$

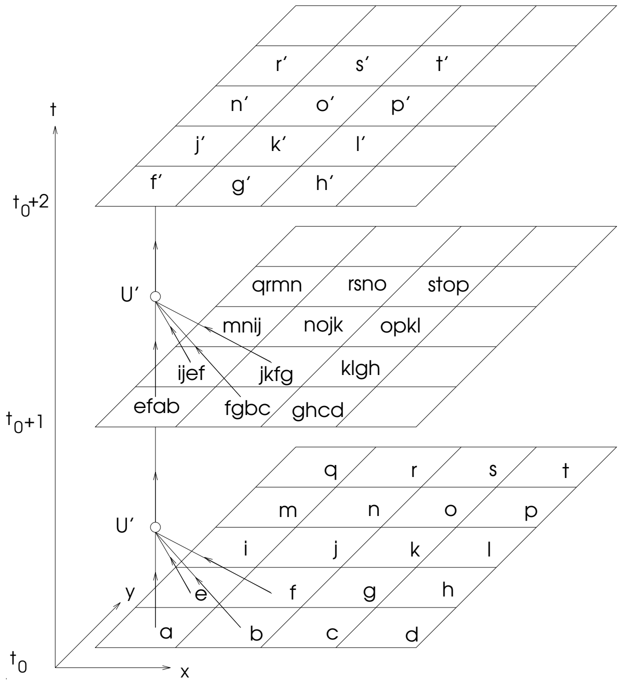

Figure 2-8: Simulation of an arbitrary two-dimensional cellular automaton by a two-dimensional contraction cellular automaton.

Since $r = 1$ and $d = 2$ in figure 2-8, the only new states introduced into Q' are $e \times f \times a \times b, f \times g \times b \times c, \ldots \in Q^4$. The product symbols will henceforth be omitted for the sake of brevity.

By construction, the local evolution rule U' has two distinct parts. First, it performs a sequence of contraction steps after which each cell is in a state encoding the states of all $(2r+1)^d$ cells of a neighborhood of $\langle U, Q, r\rangle$. In a single additional step, U' then simulates the action of U on that neighborhood.

For example, in the lower left corner of figure 2-8, the local evolution operator U' acts at time t_0 to produce the contracted state *efab* at time t_0+1. After U' has been applied everywhere, the neighbors of the corner cell are in the states *efab, fgbc, ijef* and *jkfg*. These cells contain enough information for U' to produce the contracted state $ijk\times efg\times abc \in Q^9$ which summarizes an entire 3×3 neighborhood of cell f at t_0. (That is, the accumulated neighborhood information is not centered around the cell originally in state a, but around the cell originally in state f.) But instead of recording the state of the neighborhood in an element of Q^9, we let U' be defined to simulate U on that neighborhood, e.g.,

$$U'\begin{pmatrix} efab \\ fgbc \\ ijef \\ jkfg \end{pmatrix} = U(f,e,g,i,j,k,a,b,c) = f'. \tag{2.34}$$

In the general case, $2r-1$ applications of U' are required to contract the state information of the $(2r+1)^d$ neighbors of a cell of $\langle U, Q, r\rangle$ into the 2^d-cell neighborhood of $\langle U', Q'\rangle$. Since an additional application of U' is required to simulate U on that neighborhood, the simulation proceeds at a rate $2r$ times slower than the conventional automaton being simulated. The net effect is that an exact simulation of $\langle U, Q, r\rangle$ is performed, except for a computationally unimportant drift of the simulated configuration with velocity $(-r,-r,...-r)$ per simulated step. ■

The principal cost of simulation manifests itself not in the slowdown, but in the number of states $|Q'|$ required to simulate $\langle U, Q, r\rangle$. From (2.33), we see that $|Q'| = \sum_{j=1}^{2r}|Q|^{j^d}$. Unless both r and $|Q|$ are small integers, $|Q'|$, which is bounded below by

$$|Q'| > \int_0^r e^{x^d \ln|Q|}dx, \quad \text{when } \ d,r\geq 1, \ \ |Q|\geq 2, \tag{2.35}$$

grows rapidly.

2.7 Simulation by N-body Automata

Theorem 2.7.1 *Any* d*-dimensional cellular automaton* $\langle U, Q, r\rangle$ *can be simulated by a 2-phase,* 2^d*-body automaton* $\langle \widehat{S}, \widehat{Q}\rangle$*. The* 2^d*-body automaton takes* $2r$ *steps to simulate each step of* $\langle U, Q, r\rangle$*, needs at most* 2^d *times as much space, and employs* $|\widehat{Q}| = \sum_{j=1}^{2r} |Q|^{j^d}$ *states.*

Proof. As above, the construction is given explicitly for $d = 2$, but in such a way that the generalization to any $d \geq 1$ is clear. Since Lemma 2.6.1 assures us that for any $\langle U, Q, r\rangle$ there exists a contraction automaton $\langle U', Q'\rangle$ that simulates it, it suffices to first construct the 2^d-body cellular automaton $\langle \widehat{S}, \widehat{Q}\rangle$ from $\langle U', Q'\rangle$ and then show that $\langle \widehat{S}, \widehat{Q}\rangle$ can simulate any finite volume V of $\langle U', Q'\rangle$ using a volume no larger than $2^d V$.

Let $\widehat{Q} = Q'$. By definition 2.6.3, in $\langle U', Q'\rangle$ each cell belongs simultaneously to 2^d neighborhoods, whereas (by definition 2.6.1) in $\langle \widehat{S}, Q'\rangle$ each cell can belong to only a single neighborhood at any given time. Therefore, the initial state of $\langle \widehat{S}, Q'\rangle$ must contain at least 2^d copies of the initial state of $\langle U', Q'\rangle$ and so requires at least a volume $2^d V$ (see figure 2-9 for $d = 2$ and figure 2-10 for $d = 1$). Since the same argument holds at each subsequent step of the simulation of $\langle U', Q'\rangle$ by $\langle \widehat{S}, Q'\rangle$, the local evolution operator

$$\widehat{S} : (Q')^{2^d} \to (Q')^{2^d} \tag{2.36}$$

must be constructed to produce 2^d copies of the value U' would produce on the same input; that is,

$$\widehat{S}\begin{pmatrix} a^t_{x_1,x_2,x_3\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1\ldots,x_d+1} \end{pmatrix} \equiv \begin{pmatrix} U'\begin{pmatrix} a^t_{x_1,x_2,x_3\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1\ldots,x_d+1} \end{pmatrix} \\ \vdots \\ U'\begin{pmatrix} a^t_{x_1,x_2,x_3\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1\ldots,x_d+1} \end{pmatrix} \end{pmatrix}. \tag{2.37}$$

From figures 2-9 and 2-10, we see that in fact a volume $2^d V$ is both necessary and sufficient. As indicated by the dashed boxes in figure 2-10, the 2^d-fold copies of the simulated configuration of $\langle U', Q' \rangle$ shift at the rate of $(1/2, 1/2, ..., 1/2)$ per operation of $\widehat{S}$; after $2T$ steps of simulation, the simulated configuration must be read relative to the translated origin at $(T, T, ..., T)$.

Since Lemma 2.6.1 asserts that for any $\langle U, Q, r \rangle$ there exists a contraction automaton $\langle U', Q' \rangle$ that simulates it with slowdown $2r$ and $|Q'| = \sum_{j=1}^{2r} |Q|^{j^d}$, it follows that for any $\langle U, Q, r \rangle$ there exists a 2^d-body automaton $\langle \widehat{S}, Q' \rangle$ that simulates $\langle U, Q, r \rangle$ with the same number of states and the same slowdown in a volume 2^d-times larger.
∎

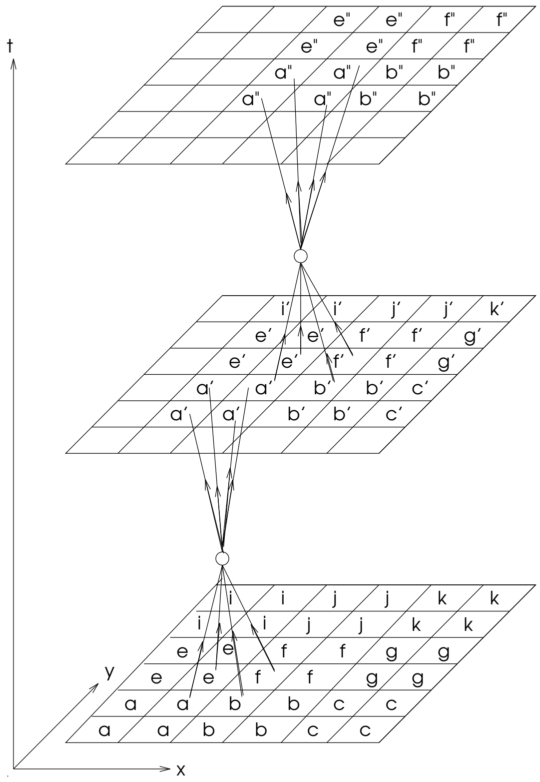

Figure 2-9: Simulation of a contraction automaton by a 2-phase, 2^d-body scattering automaton, for $d = 2$.

2.8 Universal 2-body Automata Requiring Few States

In the Introduction, we have emphasized the importance of obtaining strong numerical bounds on the minimum number of bits per cell required to perform universal computation in a few-body cellular automaton. Fabricators of cellular-automata device-arrays can use these bounds to assess the feasibility of using various physical couplings between device-cells without losing the potential for universal computation. Coupling mechanisms that are physically incapable of supporting enough bits per cell to be universal must be abandoned, or at best redirected toward special-purpose applications. Two of the most stringent cases, with clusters of size $N = 2$ and $N = 4$, occur for $d = 1, 2$ respectively ($N = 3$ does not occur on a square lattice, but can occur on a triangular lattice). If we apply Theorem 2.7.1 to conventional computation-universal CA having small neighborhood radii and small state sets, we can set an upper bound on the number of bits per cell required. For $d = 1$ ($N = 2$), figure 2-10 shows explicitly how the final stage of the simulation proceeds. The number of bits per cell required to simulate various conventional CA are discussed below and summarized in Tables 2.1 and 2.2.

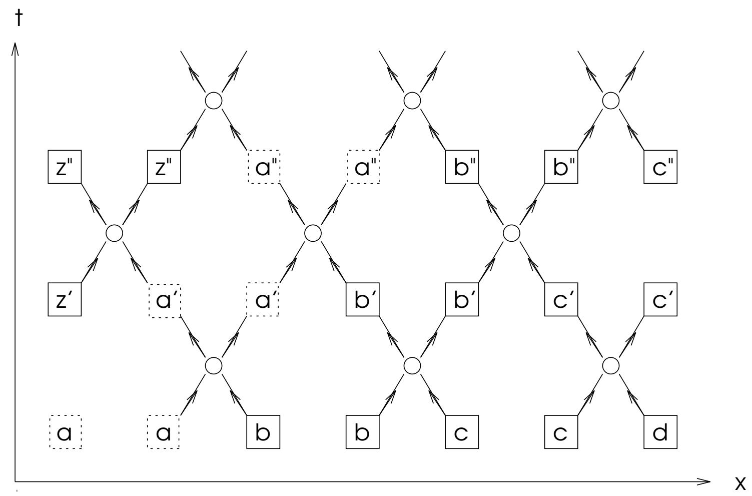

Figure 2-10: Simulation of a one-dimensional contraction (i.e., one-way) automaton by a 2-body automaton.

From the time von Neumann first introduced his two-dimensional, self-reproducing cellular automaton with 29 states, many researchers [26, 35, 6, 75] have attempted to reduce both the state count $|Q|$ and the neighborhood radius r of computation-universal rules. The earliest work focused on decreasing the number of states in von Neumann's automaton; Codd [35] reduced $|Q|$ from 29 to 8; Banks [6] later reduced $|Q|$ to 4. From Theorem 2.7.1, we see that by simulating the von Neumann, Codd and Banks constructions we obtain 4-body automata that are computation-universal and require $\log_2 |Q'| = 19.43$, 12.00 and 8.02 bits per cell respectively.

Banks also discovered a rule in $d = 2$ which, like Conway's well-known "Life" rule, has the fewest possible states, $|Q| = 2$, the smallest possible radius, $r = 1$, but is still provably computation universal. The 4-body cellular automata that simulate these rules each require just 3.32 bits per cell. Together with Margolus's one-bit version of Fredkin's universal billiard ball model [111] (which was, however, implemented from the start as a 4-body CA), the Banks and Conway rules represent the smallest possible conventional cellular automata and constitute the smallest possible universal cellular automata in $d = 2$.

In contrast, for $d = 1$ the problem of minimizing $|Q|$ while retaining universality remains open. A. R. Smith has proven a series of theorems on the simulation of (m, n)-Turing machines[5] by one-dimensional conventional cellular automata. By applying his theorems to the small $(6, 6)$- and $(4, 7)$-Turing machines that Minsky [118] has shown to be universal, Smith has produced universal one-dimensional cellular automata with neighborhood radii $r = 1, 2$ and 3.

Theorem 2.8.1 (Smith [76]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ with $|Q| = \max(m, n) + 1$ and $r = 3$ that simulates it in real-time.*

For the $(6, 6)$-Turing machine of [118], this yields a conventional CA with $|Q| = 7$, $r = 3$. Applying Theorem 2.7.1 yields a 2-body CA with $|Q'| = 137,256$ or $\log_2 |Q'| = 17.07$ bits per cell.

[5] That is, Turing machines employing m tape symbols and n head states.

Theorem 2.8.2 (Smith [76]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ with $|Q| = m+n$ and $r = 2$ that simulates it in real-time.*

For the $(4,7)$-Turing machine of [118], this yields a conventional CA with $|Q| = 11$, $r = 2$. Applying Theorem 2.7.1 yields a 2-body CA with $|Q'| = 16{,}104$ or $\log_2 |Q'| = 13.98$ bits per cell.

Theorem 2.8.3 (Smith [76]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ that simulates it in 2 times real-time and has $|Q| = m + 2n$ and $r = 1$.*

For the $(4,7)$-Turing machine of [118], this yields a conventional CA with $|Q| = 18$, $r = 1$. Applying Theorem 2.7.1 yields a 2-body CA with $|Q'| = 342$ or $\log_2 |Q'| = 8.42$ bits per cell.

Culik at al. [2] have reduced the state set to $|Q| = 14$ with $r = 1$; the resulting universal CA yields a universal 2-body CA with $|Q'| = 210$ or $\log_2 |Q'| = 7.71$ bits per cell.

Recently, Lindgren et al. [106] have strengthened Theorems 2.8.2 and 2.8.3.

Theorem 2.8.4 (Lindgren and Nordahl [106]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ that simulates it in 2 times real-time and has $|Q| = m + n + 2$ and $r = 1$.*

Applying Theorems 2.7.1 and 2.8.4 to Minsky's $(4,7)$-Turing machine would only reduce the state set to $|Q| = 13$, but Lindgren et al. reduce this further to $|Q| = 7$ and $r = 1$ by simulating the Turing machine head with composite objects propagating in a periodic background. Applying Theorem 2.7.1 to their construction yields a universal 2-body CA requiring just 5.81 bits per cell.

Smaller lower bounds on the required number of bits per cell in $d = 1$ have not yet been obtained, but several members of Wolfram class 4, widely conjectured to be universal, would yield lower bounds. The two $r = 1$ rules, W^3_{357} and W^3_{824}, (where

(d=2) Rule	$\lvert Q\rvert$	r	Universal	$\log_2 \lvert Q'\rvert$
von Neumann [147, 26]	29	1	Y	19.43
Codd [35]	8	1	Y	12.00
Banks [6]	4	1	Y	8.02
Conway("Life") [20]	2	1	Y	3.32
Banks [6]	2	1	Y	3.32

Table 2.1: Simulation of two-dimensional conventional automata by 4-body automata.

W_c^k denotes the k-ary rule with Wolfram [151] code c) both have $|Q| = 3$, and hence would yield $\log_2 |Q'| = 3.58$ bits per cell if their universality can be established.

Since we are primarily interested in minimizing N for universal N-body automata, we can obtain 2-body automata with even fewer states by applying the contraction operation within the 2×2 clusters of universal 4-body automata (see figure 2-11). It is not difficult to see that

Claim 2.8.1 *Any 4-body automaton $\langle \widehat{S}, \widehat{Q} \rangle$ can be simulated by a 2-body automaton $\langle \widetilde{U}, \widetilde{Q} \rangle$ at the cost of a slowdown of 2 and augmentation of the state set $|\widetilde{Q}| = 2|\widehat{Q}|$.*

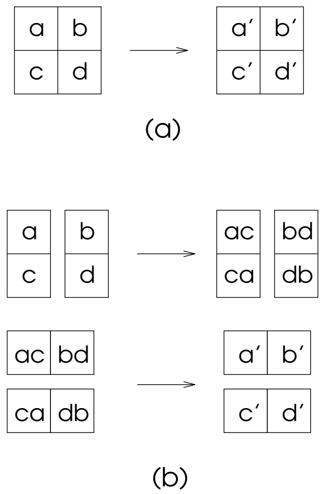

Figure 2-11: Simulation of a 4-body automaton by a 2-body automaton.

(d=1) Rule	$\|Q\|$	r	Universal	$\log_2 \|Q'\|$
Smith [76]	7	3	Y	17.07
Smith [76]	11	2	Y	13.98
Smith [76]	18	1	Y	8.42
Lindgren et al. [106]	4	2	Y	8.41
Albert et al. [2]	14	1	Y	7.71
Lindgren et al. [106]	7	1	Y	5.81
W^3_{357}, W^3_{824}	3	1	—	3.58
W^2_{110}	2	1	—	2.58

Table 2.2: Simulation of one-dimensional conventional automata by 2-body automata.

2.9 Summary

In order to put to rest any fears that few-body automata may have lost the ability to perform universal computation, or may incur unacceptable costs to do so, in this chapter we have explicitly shown that there is at least one completely straightforward way in which a few-body automaton can simulate any conventional cellular automaton, including the ones that perform universal computation. The costs of this simulation are calculated in detail, and found not to require an exhorbitant number of states or amount of space. This does not rule out the possibility that there are other ways to simulate conventional cellular automata with few-body automata that are even more efficient.

In particular, we have tried to find the a good upper bound on the number of bits per cell required to perform universal computation in 2-body automata, because this is the type of few-body automaton we will employ in the Lieb/Shultz/Mattis model of quantum computation discussed in Chapter 5.

By applying Theorem 2.7.1 to universal conventional CA , we obtained an upper bound on the number of bits per cell required for universal computation when only two-body interactions are permitted. The lowest upper bound we have established, based on a conventional CA constructed by Lindgren [106], is 5.81 bits per cell. If the hypothesized universality [151] of Wolfram rules 357 and 824 can be proven, the bound can immediately be reduced to 3.58 bits per cell. If any binary CA with $r = 1$ can be shown to be universal (and there is some disagreement about the possibility;

compare [106] with [151, p.31]) then the required amount of state can be reduced to 2.58 bits per cell.

In $d = 2$ we have shown how to construct a computation-universal 2-body automaton requiring 2 bits per cell. This requirement can probably be met by many of the proposed schemes for computing with nanometer-scale arrays. The construction of a computation-universal 2-body automaton that only requires a single bit per cell remains an open problem.

Chapter 3

Logical Completeness

In this chapter we return to the problem of determining the logical properties of physical interactions. Recall from §2.3 that we follow Landauer in assuming that we are given a mapping ξ from the physical states of information-bearing degrees of freedom to the logical values they represent. One of the benefits of few-body automata is that ξ induces a natural mapping from the physical evolution determined by the scattering matrix $\widehat{S}$ (or master equation (2.26)) to the vertex rule $\mathcal{S}$ of a lattice-gas cellular automaton. The vertex function can be interpreted as an N-input/N-output logic gate. Each of its N outputs can be regarded as the output of an N-input gate—an ordinary binary logic gate—and its logical properties can therefore be analyzed using the tools of classical circuit theory. By this route, we can analyze the logical properties of $\mathcal{S}$ and, via ξ, ultimately the logical properties of the physical interactions that determine $\widehat{S}$. In particular, we will use the Post-Glushkov theorem[132, 60] to give a new graphical method for determining when, under interpretation ξ, a physical interaction is "rich" enough to support computation or, in terms of classical circuit theory, "logically complete". The Post-Glushkov theorem states that, given ready access to constants of 0 and 1, a gate is logically complete unless it is either linear or monotonic.

In Sections 3.2 and 3.3 we define what 'linear' and 'monotonic' mean in this context. Then we show how both properties can easily be detected in a graph of the *partial order* on input states. Finally, we use the partial order technique to identify precisely which transitions from input to output—physically, which elements of the $\widehat{S}$ matrix—give the billiard ball cellular automaton[111] its logical completeness, and hence computation-universality.

3.1 Partial Orders

A partial order is like the familiar ordering relationships, such as "$<$", which defines an order on any pair of distinct real numbers. The only difference is that for a partial order there exist pairs of objects which are not related at all; neither one is "smaller" than the other.

The partial order "$\lhd$" that plays a role in determining the logical properties of an N-input binary gate is a partial order on the vector space of all possible inputs; that is, on all N-vectors with binary components. If $A = (a_1, a_2, ..., a_N)$, $a_i = \xi(\psi_{\text{in}}^{(i)}\rangle) \in \{0, 1\}$, represents a possible input state and $B = (b_1, b_2, ..., b_N)$ represents another possible input, then we say

$$A \lhd B \tag{3.1}$$

if and only if for all i, $a_i = 1$ implies $b_i = 1$. That is, $A \lhd B$ only if B has a 1 in every position that A does. Figure 3-1 is a graph of the partial order that $\lhd$ induces in the case N=2. The graph of a partial order is called (somewhat infelicitously) a *lattice*. Each possible input state is represented by a node. The "largest" inputs are further up the lattice. If an input B satisfies $B \lhd A$, then there is a path from B to A consisting only of upward links.

The principal advantage of the lattice representation is that it provides a unified representation for simultaneously determining whether the corresponding gate has the two properties, nonmonotonicity and nonlinearity, which are necessary and sufficient for the gate to be logically complete. If, in addition, it is physically possible to use the outputs of one gate as the inputs to another, then the gate is universal in the sense that any computable binary function can be constructed from a collection of such gates.

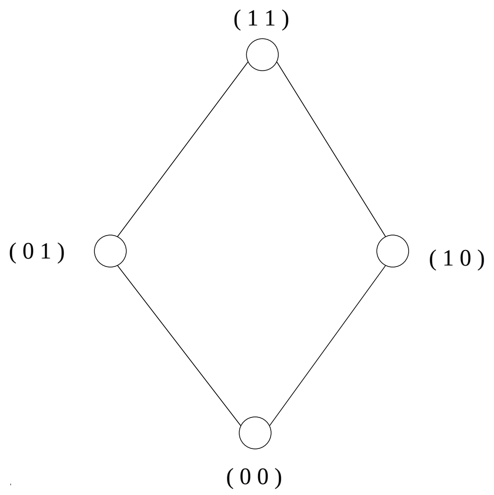

Figure 3-1: Partial order $\lhd$ on input states with two binary components.

3.2 Linearity

It is straightforward to verify that every binary function of n binary inputs can be written in the canonical form

$$f(x_1, x_2, ...x_N) = \bigotimes_{i=0}^{2^N-1} a_i \wedge (x_{j_1} \wedge x_{j_2} \wedge ... \wedge x_{j_k}), \tag{3.2}$$

where $\{j_p\}$ are the positions of the nonzero bits in the binary expansion of the integer i, $\otimes$ is the exclusive-or (parity) operator, and $\wedge$ is the logical AND (conjunction) operator.

DEFINITION. A gate is *linear* if its canonical form contains no conjunctions of two or more x_i

$$: f(x_1, x_2, ...x_N) = a_0 \otimes (a_1 \wedge x_1) \otimes (a_2 \wedge x_2) \otimes ... \otimes (a_N \wedge x_N). \tag{3.3}$$

That is, if all inputs but one are held fixed and the remaining input is inverted, then the output is inverted. The output is therefore the linear superposition (mod 2) of the inputs.

3.3 Monotonicity

Definition. A gate $f : (Z_2)^N \rightarrow Z_2$ is *monotonic* if and only if, for any two input vectors A and B,

$$(A \lhd B) \wedge (f(A) = 1) \quad \Rightarrow \quad f(B) = 1. \tag{3.4}$$

That is, if the input increases, as measured by the partial order $\lhd$ on binary vectors, then the output either remains the same or increases.

The monotonicity properties of a gate are trivial to verify in the lattice representation. Since the binary input vectors are ordered from smallest at the bottom to largest at the top of the lattice, to determine whether or not a gate is monotonic it suffices to search the links in the lattice from bottom to top. If a link is found with a 1 at the bottom node and a 0 at the top node, then and only then is the gate nonmonotonic.

3.4 Graphical Algorithm for Determining Completeness

Seeing whether or not a gate is nonlinear from the partial order graph requires a little more effort than seeing that it is nonmonotonic. The best way to grasp the trick is to understand how the canonical form (3.2) can be derived using the lattice of binary input states.

The method consists of successively approximating the actual output until an exact canonical expression is found after $N+1$ iterations. Starting with the smallest input (according to the partial order $\lhd$), we write down an initial approximation $f_0(x_0, ..., x_{N-1}) = 0$, which has the canonical form (3.2). In general, approximations that produce the correct output for smaller values of the input will produce incorrect values for some of the larger values further up the lattice. Working our way up the lattice, these cases are corrected. A new approximation is produced by taking the exclusive-or ($\otimes$) of the previous approximation with a term that identifies the case in which the previous approximation gives the wrong output. Since the output is binary, the exclusive-or operation flips the output value for that case, thereby correcting it. Because successive approximations proceed from smallest to largest inputs, this cannot cause any of the previously correct output values for smaller inputs to become incorrect. It can, however, cause output values for larger inputs that were accidentally correct to become incorrect but, being larger inputs, these cases will be corrected at a later stage of the procedure.

All of the essential points occur in the following simple example. Using the lattice on two binary inputs (Fig. 3-1), we show how to construct the canonical form (3.2) of the logical OR operation.

In Fig. 3-2, the input vectors are shown in parentheses and the correct output value of OR are shown in square brackets. White circular nodes indicate binary value 0 in the current approximation and black circular nodes indicate binary value 1. Since OR has two inputs, the graph of the partial order has three levels; the 0^{th} level has one node, (00), the 1^{st} level has two nodes and the second level has a single node.

Starting with the (00) node, in Fig. 3-2(a) we find that the initial approximation, $f_0^{\text{OR}} \approx 0$ is correct since the actual output is [0].

In Fig. 3-2(b) we move to node $(x_0 x_1) = (01)$ on the 1^{st} level. The previous approximation f_0^{OR} gives 0 for the output when (01) is the input, while the actual output is [1]. To correct the situation we form the term x_1 which describes the input $(x_0 x_1) = (01)$, and take the exclusive-or of this term with the previous approximation to produce a new approximation

$$f_1^{\text{OR}} = x_1 \otimes f_0^{\text{OR}} = x_1 \tag{3.5}$$

that gives the correct output of the OR gate for both inputs (00) and (01). This is indicated in Fig. 3-2(b) by the fact that the approximation at node (00) (white circle) agrees with the desired output [0], and similarly the approximation at (01) (black circle) matches the output [1] of OR on that input.

In Fig. 3-2(b) we see that node $(x_0x_1) = (10)$ still disagrees with the approximation $f_1^{\rm OR} = x_1$ since the correct output [1] is not matched by the white node. To correct this, we form the new approximation by taking the exclusive-or of x_0 with the previous approximation, obtaining

$$f_2^{\rm OR} = x_1 \otimes x_0. \tag{3.6}$$

As shown in Fig. 3-2(c), this now agrees with the actual output of OR on inputs (00), (01) and (10), but the approximation $f_2^{\rm OR} = x_1 \otimes x_0$ is still in disagreement on input (x_1x_1).

The final approximation,

$$f_3^{\rm OR} = x_1 \otimes x_0 \otimes (x_0 \wedge x_1), \tag{3.7}$$

which brings the node (11) into agreement, is formed by taking the exclusive-or of the (11) term $x_0 \wedge x_1$ with $f_2^{\rm OR}$.

All nodes are now agreement with the actual output of OR, and the final approximation is the exact canonical form of OR

$$f_{\rm OR}(x_0, x_0) = x_1 \otimes x_0 \otimes (x_0 \wedge x_1). \tag{3.8}$$

Because of the factor $(x_0 \wedge x_1)$, this is not of the form (3.3), and so the logical OR function is not linear.

The general procedure for determining nonlinearity in the lattice representation is now easy to describe. A function is nonlinear if, in the "bottom-up" approximation procedure, the first-order approximation $f_1(x_0, ..., x_{N-1})$ is not identical to the final exact canonical form.

Stated another way, suppose we are given two things: some structure which evolves its N asymptotic input states $|\psi_{\rm in}^{(}i)$ into N asymptotic output states $|\psi_{\rm out}^{(}i)$, and a mapping xi that associates binary values with each of these states. Then we can determine whether or not a particular output n of the structure is logically complete by drawing the lattice representation of the partial order on N binary inputs and labeling each node with the value of the output that is associated with that input. If there is at least one bottom-up path with a $1 \to 0$ transition, output n of the structure is nonmonotonic. Then, to determine whether or not output n is linear, look at the nodes on the 2nd level. Each node will have several predecessors on the 1st level. If there is even a single node of the second level whose output value is not equal to the parity of the number of its predecessors on the 1st level whose output value is 1, then the gate is nonlinear. Similarly for all higher levels, if even a single node does not have the output value predicted by the parity of its 1st level predecessors, output n of the structure is nonlinear; otherwise, it is linear.

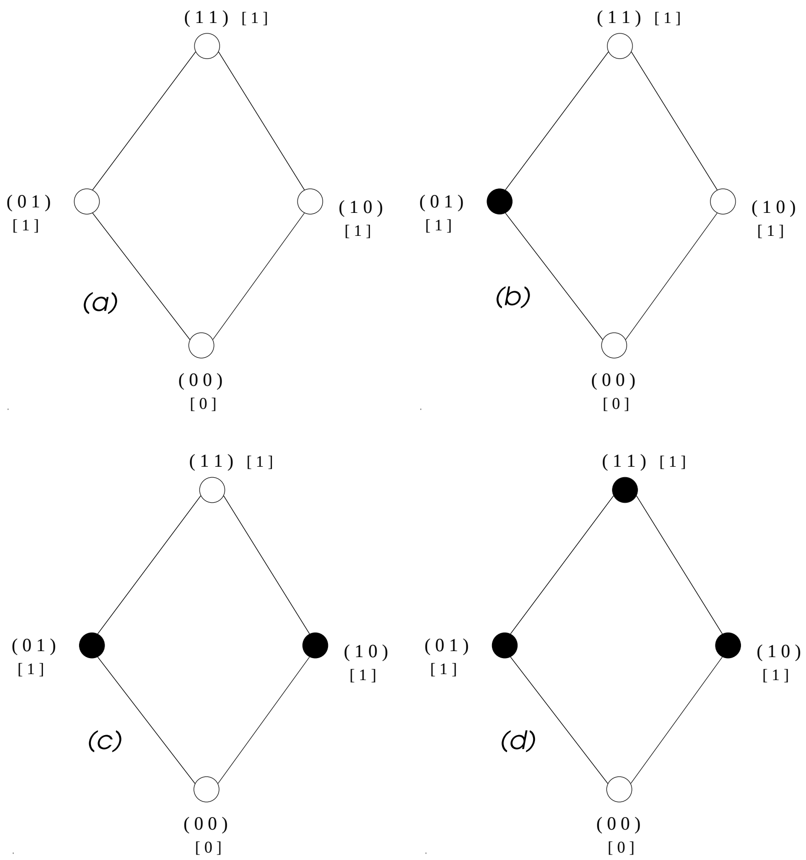

Figure 3-2: Lattice procedure for finding the canonical form of the OR gate output f^{OR}. Input vectors are in parentheses, values of f^{OR} are in square brackets, and values of successive approximations are unbracketed. The successive approximations to $f^{\mathrm{OR}}(x_0, x_1)$ are: $f_0^{\mathrm{OR}} = 0$, $f_1^{\mathrm{OR}} = x_0$, $f_2^{\mathrm{OR}} = x_0 \otimes x_0$, and the final canonical form $f^{\mathrm{OR}} = x_0 \otimes x_0 \otimes (x_0 \wedge x_0)$.

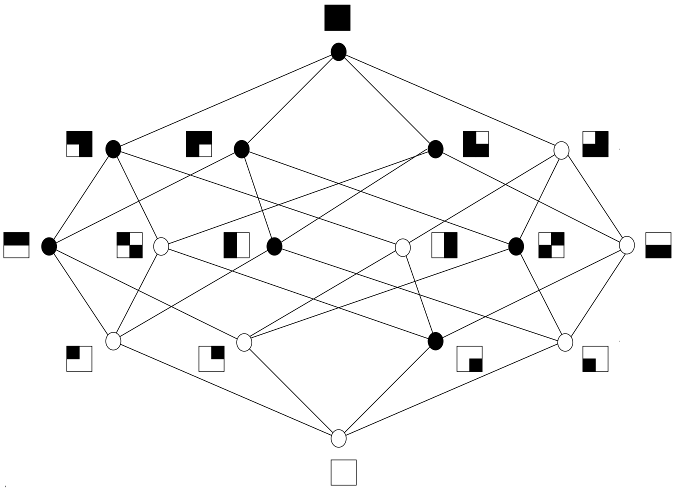

Figure 3-3: Lattice representation of the upper-left cell output for the billiard ball cellular automaton[113].

3.5 Physical Locus of Logical Completeness in the Billiard Ball Model

We close this section by showing how the partial order method can be used to identify the particular state transitions of the billiard ball cellular automaton that make it computation-universal. Figure 3-3 shows the partial order for the output of the upper-left cell of Margolus's cellular automaton version[113, 111] of Fredkin's billiard ball model[54] (see Fig. 6-3 for the complete transition matrix $\mathcal{S}_{\mathrm{BBM}}$), when the output of the upper-left cell is considered a function of the input values of all four cells.

Since it is computation-universal, the properties of nonlinearity and nonmonotonicity must occur somewhere in the mapping $\mathcal{S}_{\mathrm{BBM}}$ from inputs to outputs. In the lattice representation, the nonmonotonicity is apparent in the transitions ▗ $\rightarrow$ ▚, ▗ $\rightarrow$ ▐, ▗ $\rightarrow$ ▄ and ▞ $\rightarrow$ ▟. The nonlinearity is evident in any transition from the first (i.e., next to lowest) level to the second level. In particular, the input ▐ has two predecessors on the 1st level, ▝, which has output 0, and ▗, which has output 1. Therefore, if it were linear, the output of ▐ would be equal to the parity of the outputs of its predecessors, or 1. Since the output is 0, the upper-left cell of the 2×2 neighborhood, regarded as a logical function of the input states of the four cells, is nonlinear as well nonmonotonic. Therefore, under the assumptions of the Post-Glushkov theorem, it is logically complete, and so can provide the basis for constructing any binary function.

Chapter 4

Inverse Quantum Scattering

In the previous chapter we have seen how the partial order graph on input states enables us to determine the logical properties a few-body automaton acquires from the interaction it is based on. However, there are circumstances, particularly in the search for practical applications, where we might want to do the converse—that is, we might want to specify the logical operation, the input/output relation of a given few-body automaton, and find the form of some interaction that will produce that operation. By relating primitive logical operations to physical scattering events, few-body automata make it possible in principle to achieve this by applying a recently developed tool for determining interactions from scattering data—*inverse* scattering theory. Although current inverse scattering techniques are not quite general enough to solve the cases that are most interesting from a computational point of view, Cheney[32] and others[122, 123, 124] are actively extending the techniques. In the expectation that these techniques will be of increasing utility in the very near future, in this chapter, we briefly describe how we might apply the inverse scattering technique, along with a simple derivation of the central result of inverse scattering theory, namely the Marchenko equation.

In the familiar theory of (direct) quantum scattering, we learn to calculate the scattering amplitude $\widehat{S}$ given the interaction potential $V(\vec{r})$. The goal of inverse scattering theory is just the opposite: for a specified $\widehat{S}$, find the interaction $V(\vec{r})$ that would produce $\widehat{S}$.

Intuitively, the inverse problem may seem, except in special cases, to be intrinsically underdetermined—like asking if we can hear the shape of a drum[80]. But a moment's reflection reveals that the inverse problem is the problem physicists really wanted to solve all along. We only find the direct problem so familiar because it is so often the case that the inverse problem cannot be solved, and our only recourse is to guess, solve the direct problem, compare to experiment, and guess again until we are confident we have found the correct potential.

The quantum inverse scattering problem is indeed solvable in some cases, notably for local, rotationally symmetric potentials in three dimensions and local potentials in one dimension. However, there are some caveats. In particular, if it is known that the unknown potential has bound states (as all attractive potentials in one dimension have), then the method of solving the inverse problem requires us to provide some additional (and difficult to obtain) data about the bound state wavefunctions. However, we will see that in the application to few-body automata, these problems would not arise because the input data, $\widehat{S}$ would not come from scattering experiments.

Instead, we would start with a computation-universal few-body cellular automaton. Any collection of scattering amplitudes consistent with its input/output relation would serve as a "synthetic" $\widehat{S}_{\text{comp}}$ for the inversion procedure. By solving the inverse problem, we would obtain an analytic expression for an interaction $V(\vec{r})$. If any real systems existed that were governed by this interaction, its dynamics would be isomorphic to that of the few-body automaton described by $\widehat{S}_{\text{comp}}$.

Unfortunately, it is not quite possible yet to carry out this program because the development of inverse scattering theory has been much more laborious than that of the more familiar (direct) scattering theory. The history of inverse scattering theory really begins around 1940 when Heisenberg[69] and Wheeler[149] first defined the scattering matrix $\widehat{S}$, and Heisenberg conjectured that it contained all "physical" information. If this were the case, it should be possible to determine all physical observables, in particular the scattering potential $V(\vec{r})$, from $\widehat{S}$. Soon thereafter, Hylleraas published a procedure for finding a central potential $V(r)$ from the scattering phase shifts $\delta_l(E)$, which are related to the matrix elements $s_l(E)$ of $\widehat{S}$ in the angular momentum sector l by $s_l(E) = e^{i2\delta_l(E)}$. Unfortunately, in the following year Bargmann[8, 7] exhibited two distinct central potentials that give identical phase shifts, thus showing that Hylleraas' procedure was flawed. Levinson's theorem[102] reestablished the uniqueness of Hylleraas procedure when the potential is assumed to have no bound states. But Levinson's theorem simulataneously dealt a fatal blow to Heisenberg's idea that the $\widehat{S}$ matrix contained all physical information, since the theorem showed that the *only* connection between the bound states and phase shifts is that $\delta_l(E \to 0) - \delta_l(E \to \infty) = N_l$, where N_l is the number of bound states of angular momentum l. Still missing was a solution to the inverse problem that worked when the unknown potential was known to have bound states. In the narrowest mathematical sense, I. M. Gel'fand and B. M. Levitan[58] gave the first such solution in 1951. But their solution requires knowledge of physically peculiar scattering data, such as the slope of the normalized bound state wavefunctions at the origin[1].

The first solution of the quantum inverse scattering problem that starts from the kind of data few-body automata can provide—the scattering matrix—was presented by Marchenko[109] in 1955, almost thirty years after the corresponding direct scattering problem was first solved. Since the Marchenko equation is the simplest member of the class of inverse methods that are applicable to few-body automata, we devote the next two sections to reviewing its structure. In a third section, we examine the prospects for using more sophisticated methods in the Marchenko class for the purpose of deriving the form of computationally interesting interactions from the form of few-body automata.

The Marchenko equation is simplest in the case where it is known beforehand that the potential we are seeking has no bound states. In that case, the Marchenko method requires that, given the scattering amplitude $s_l(E)$ at some particular angular

[1]There are some cases where the so-called "norming constants" can either be deduced from (non-scattering) experiments, or dispensed with altogether. Thaker *et al.*[141] have reconstructed infinitely deep quark potentials this way by observing that for symmetric potentials $V(r) = -V(-r)$, knowledge of the bound state energies alone is sufficient.

momentum l for all energy E, we first form the kernel

$$Q(r,r) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[1 - s_l(E)]\chi_l^{0+}(E,r)\chi_l^{0+}(E,r')dE \tag{4.1}$$

where $\chi_l^{0\pm}(E,r)$ are the *free Jost solutions*[2], solutions of the free Schrödinger equation that behave asymptotically as $\chi_l^{0\pm}(E,r) \to e^{\pm ik(E)r}$.

Next, the method prescribes that we solve the linear integral equation

$$K(r,r') + Q(r,r') + \int_r^{\infty} K(r,r'')Q(r'',r')dr'' = 0, \tag{4.2}$$

known as Marchenko's equation, for the generalized shift operator $K(r,r')$. Finally, the desired potential is

$$V(r) = \frac{\hbar^2}{m}\frac{d}{dr}K(r,r). \tag{4.3}$$

The only thing immediately clear about this procedure is that the scattering data goes in one end and the potential comes out the other. In the next two sections, I try to expose the simple quantum mechanical ideas behind Marchenko's seemingly opaque inversion procedure.

4.1 Inverse Schrödinger Equation: Discrete Case

The physical ideas that underly solution of the inverse scattering problem are most easily grasped when we solve the problem on a discrete lattice[29, 30, 154, 153]. We consider the problem of finding a three-dimensional, rotationally-invariant potential for the non-relativistic scattering of a spinless particle from a finite-range potential given the scattering amplitude $s_l(E) = e^{2i\delta_l(E)}$ for all energies E at fixed angular momentum l. Except for the property of rotational-invariance, each of these conditions can be relaxed without greatly increasing the complexity[31].

Since the system is rotationally symmetric, we will be solving the radial Schrödinger

[2] For scattering in three dimensions from an unknown potential that is assumed rotationally symmetric, $\chi_l^{0\pm}(E,r)$ are the Ricatti-Hankel functions[139]; in one dimension, they are plane waves.

equation. Letting $\Psi(r,\theta,\phi) = r\psi_l(E,r)Y_l^m(\theta,\phi)$, after separation of variables the radial equation becomes

$$-\frac{d}{dr^2}\psi_l(E,r) + \frac{l(l+1)}{r^2}\psi_l(E,r) = k^2\psi_l(E,r). \tag{4.4}$$

Defining $k = \sqrt{2mE}/\hbar$, two independent solutions of 4.4 are the Ricatti-Bessel functions

$$j_l(kr) = \sqrt{\frac{\pi kr}{2}}J_{l+1/2}(kr), \tag{4.5}$$

and Ricatti-Neumann functions

$$n_l(kr) = (-1)^l\sqrt{\frac{\pi kr}{2}}J_{-l-1/2}(kr). \tag{4.6}$$

These functions behave asymptotically (i.e. as $r \to \infty$) as $j_l(kr) \to sin(kr - l\pi/2)$ and $n_l(kr) \to cos(kr - l\pi/2)$.

It is more useful to form combinations of these functions that behave asymptotically like incoming and outgoing plane waves, and these are called the Ricatti-Hankel functions

$$h_l^{\pm}(kr) = n_l(kr) \pm ij_l(kr). \tag{4.7}$$

Since the potential $V(r)$ that produced the given $s_l(k)$ is assumed to have finite range (i.e. $V(r \geq a) = 0$), the exact solution $\psi_l(k,r)$ of (4.4) satisfies

$$\psi_l(k, r \geq a) = \frac{i}{2}[h_l^-(kr) - s_l(k)h_l^+(kr)]. \tag{4.8}$$

For $r < a$, the exact solution can be expressed in the same form, but with the so-called *Jost solutions* $\chi_l^{\pm}(k,r)$ replacing the Riccati-Hankel functions

$$\psi_l(k, r \geq a) = \frac{i}{2}[\chi_l^-(kr) - s_l(k)\chi_l^+(kr)]. \tag{4.9}$$

Without loss of generality[3], we consider the case that we are given the $l = 0$

[3]We will see later that it is no trouble at all to include the centrifugal potential for $l \neq 0$, or any

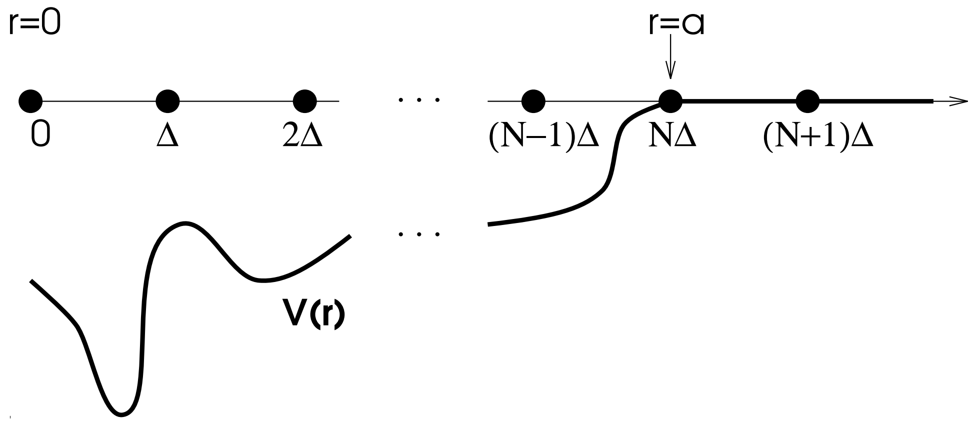

Figure 4-1: Discrete radial Schrödinger equation. The potential vanishes for $r \geq a = N\Delta$.

scattering amplitude $s(E) \equiv s_0(E) = e^{2i\delta_0(E)}$ so that the centrifugal potential term $l(l+1)/r^2$ vanishes. Then the discrete radial Schrödinger equation (see Fig. 4-1) becomes

$$\frac{-\hbar^2}{2m}\frac{\psi(E,n+1) - 2\psi(E,n) + \psi(E,n-1)}{\Delta^2} + V(n)\psi(E,n) = E\psi(E,n). \tag{4.10}$$

Where $V(n)$ is completely unknown for $n < N$ and is identically zero for $n \geq N$; that is $a = N\Delta$.

Since $s(E)$ determines $\psi(E,n)$ for $n \geq N$, we would like to solve the set of discrete coupled equations (4.10) by performing the recursion

$$\psi(E,n-1) = \left(2 + \frac{2m\Delta^2}{\hbar^2}[V(n) - E]\right)\psi(E,n) - \psi(E,n+1) \tag{4.11}$$

inward from the asymptotic region $n \geq N$.

Set $l = 0$ in (4.8) and substitute the asymptotic forms of $h_l^{\pm}(kr)$. At points N and $N+1$ the free asymptotic behavior is still valid and we can determine the

in fact any fixed local potential at all. The only requirement is that we should be able to solve for the resulting energy eigenstates, because we will need to use them in place of the free eigenstates. The inversion procedure then requires as input the phase shifts (or scattering amplitude) relative to the "free" case that includes the fixed potential, and it yields a potential difference relative to the fixed potential.

wavefunction from the exact free solutions and the given scattering data $s(E)$,

$$\psi(E, N+1) = e^{i\delta_0(E)} sin[k(N+1)\Delta + \delta_0(E)] \tag{4.12}$$

and, since $V(N+1) = 0$,

$$\psi(E, N) = \left(2 - \frac{2m\Delta^2}{\hbar^2}E\right)\psi(E, N+1) - \psi(E, N+2). \tag{4.13}$$

At $n = N-1$, the inward recursion begins without difficulty. From the wavefunction recursion relation (4.11) and $V(N) = 0$ we find,

$$\psi(E, N-1) = \left(2 - \frac{2m\Delta^2}{\hbar^2}E\right)\psi(E, N) - \psi(E, N+1). \tag{4.14}$$

However, the next inward step of the recursion runs into trouble. The recursion relation (4.11) yields

$$\psi(E, N-2) = \left(2 + \frac{2m\Delta^2}{\hbar^2}[V(N-1) - E]\right)\psi(E, N-1) - \psi(E, N). \tag{4.15}$$

The problem with (4.15) is that it contains *two* unknowns, $\psi(E, N-2)$ and $V(N-1)$. Furthermore, no help is to be found in applying the recursion relation to $n = N-3$ since that introduces two new unknowns, $\psi(E, N-3)$ and $V(N-2)$.

This is probably the impasse our intuition had in mind when it warned us that solving the inverse problem is just impossible in principle.

Fortunately, our intuition is incorrect. Another relation can be brought to bear: the completeness relation of the energy eigenfunctions,

$$\frac{1}{2\pi}\int \psi(E, n)\psi^*(E, n)dE = \delta_{mn}/\Delta. \tag{4.16}$$

Multiplying (4.15) by $\psi(E, N-1)$ and integrating over all energies yields an expression

$$V(N-1) = \frac{\Delta}{2m\pi}\int E\psi^*(E, N-1)\psi(E, N)dE - \frac{\hbar^2}{m\Delta^2}, \tag{4.17}$$

for the unknown potential. $\psi(E,N)$ and $\psi(E,N-1)$ are already known from the previous iteration of the wavefunction recursion relation (4.11).

Notice that our ability to obtain expression (4.17) hinged on two easily overlooked but crucial properties. First, by using the completeness relation, we have commited ourselves to obtaining $\psi(E,N)$ and $\psi(E,N-1)$ not just for positive energy scattering states, but also for the bound states, if any. In particular, we must be given the energies $\{E_i\}$ of these bound states and the normalization c_i of their wavefunctions $\psi(E_i,r) = c_i\chi_l^{0\pm}(iE_i,r)$ when expressed as free scattering states at imaginary energy Second, in order to obtain (4.17) we used the fact that the potential entered only at the point $n = N-1$; that is, we have assumed that the unknown potential $V(r)$ is local.

Once $V(N-1)$ is known, we can go back to the wavefunction recursion relation (4.15) and continue the inward recursion by solving for $\psi(N-2)$. It is straightforward to see that we can alternate application of the wavefunction recursion relation (4.11) and the potential recursion relation,

$$V(n-1) - \frac{\Delta}{2m\pi}\int E\psi^*(E,n-1)\psi(E,n)dE - \frac{\hbar^2}{m\Delta^2}, \tag{4.18}$$

as indicated schematically in Fig. 4-2.

The important thing to note is that the dependencies denoted by the arrows are all directed downward or horizontally, and therefore the coupled recursion relations can descend toward $n = 0$ until both $\psi(E,n)$ and $V(n)$ are found for all $n \geq 0$, and the inverse scattering problem is solved.

4.2 Derivation of the Marchenko Integral Equation

The wavefunction and potential recurrence relations (4.11) and (4.18) show that the physical ideas behind Marchenko's solution (4.2) of the inverse problem are indeed simple—completeness of energy eigenfunctions and inward recursion from the asymp-

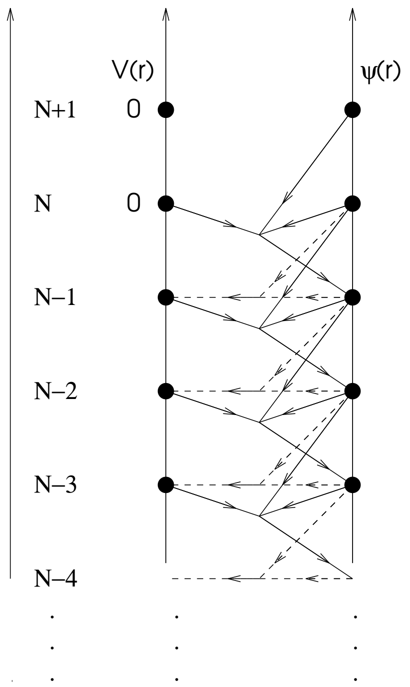

Figure 4-2: Schematic representation of how the two recursion relations alternate in solution of the Marchenko-like discrete inverse scattering problem.

totic wavefunctions determined by $s(E)$. They also provide a convenient starting point for numerically solving the inverse problem. But we still haven't connected the inward recursion described in the last section to the Marchenko equations (4.1–4.3), which were presented *ex cathedra.* In this subsection we follow Zakhar'ev, Mel'nikov et al.[153, 154] in deriving the Marchenko equation from the discrete solution of the inverse problem.

We start by introducing a change of the energy variable E that will simplify the algebra. Rewriting the free Schrödinger equation as

$$\psi(E, n+1) + \psi(E, n-1) = \left(-\frac{2mE}{\hbar^2} + \frac{2}{\Delta^2}\right) \psi(E, n), \tag{4.19}$$

suggests that we might be able to simplify the recrusively generated expression for the free Jost solutions $\chi^{0\pm}(E, n)$ by defining

$$z + \frac{1}{z} \equiv \left(-\frac{2mE}{\hbar^2} + \frac{2}{\Delta^2}\right). \tag{4.20}$$

Then

$$\psi(z, n+1) + \psi(z, n-1) = (z + \frac{1}{z})\psi(z, n), \tag{4.21}$$

which has solutions

$$\chi^{0\pm}(z, n) = z^{\pm n}. \tag{4.22}$$

Since for $l = 0$ (and also in one dimension), the free Jost solutions satisfy

$$\chi^{0\pm}(E, n) = e^{\pm ikn\Delta}, \tag{4.23}$$

so we recognize that

$$z = e^{ik(E)\Delta}. \tag{4.24}$$

Because $V(n \geq N) = 0$, the full Jost solutions satisfy

$$\chi^{\pm}(z, n) = \chi^{0\pm}(z, n), \quad n \geq N. \tag{4.25}$$

So we can start the inward recursion

$$\chi^{\pm}(z, n-1) = \left(z + \frac{1}{z} + \frac{2m}{\hbar^2}V(n)\right) - \chi^{\pm}(z, n+1) \tag{4.26}$$

for the full Jost solutions at $n = N$ using

$$\chi^{\pm}(z, N+1) = \chi^{0\pm}(z, N+1) = z^{\pm(N+1)}, \tag{4.27}$$

and

$$\chi^{\pm}(z, N) = \chi^{0\pm}(z, N) = z^{\pm N}. \tag{4.28}$$

From the form of the recursion (4.26) it is clear that for $n < N$ the $\chi^{\pm}(z, n)$ are polynomials in z. As the recursion proceeds toward the origin, the least power of z present in the polynomial $\chi^{\pm}(z, n)$ decreases by one and the largest power of z present increases by one.

For $N - 3 < n \leq N - 1$, exact cancellations of the highest power (attributable to residual effects of the free asymptotic form $z^{\pm n}$) cause a slight deviation from this

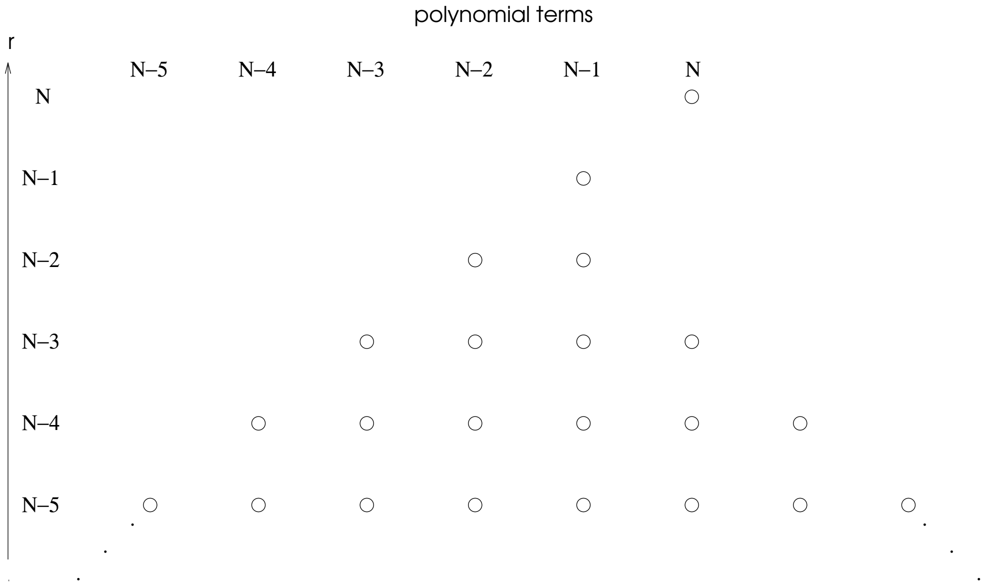

Figure 4-3: Schematic representation of the polynomials $\chi^{+}(z,N)$.

rule. The powers of z actually present in the polynomial $\chi^{+}(z,N)$ are shown in Fig. 4-3.

Expanding the full Jost functions $\chi^{\pm}(z,N)$ in terms of the free Jost functions $\chi^{0\pm}(z,N) = z^{\pm n}$,

$$\chi^{\pm}(z,N) = \chi^{0\pm}(z,N) + \sum_{m=n+1 \leq 2N-(n+3)}^{2N-(n+3)} K(n,m)\chi^{0\pm}(z,N)\Delta. \tag{4.29}$$

By definition, the radial wavefunction can be written as

$$\psi(z,n) = \frac{i}{2}[\chi^{-}(z,N) - s(z)\chi^{+}(z,N)], \tag{4.30}$$

in terms of the scattering amplitude $s(z)$ and Jost functions. Using (4.29) to represent the Jost functions, the radial equation can be written

$$\psi(z,N) = \psi^{0}(z,N) + \sum_{m=n+1 \leq 2N-(n+3)}^{2N-(n+3)} K(n,m)\psi^{0}(z,N)\Delta, \tag{4.31}$$

where the auxilliary functions

$$\psi^0(z,n) = \frac{i}{2}[\chi^{0-}(z,n) - s(z)\chi^{0+}(z,n)] \tag{4.32}$$

are defined to be that combination of solutions $\chi^{0\pm}(z,n)$ to the free Schrödinger equation that have exactly the asymptotic behavior prescribed by the scattering amplitude $s(z)$.

From the orthogonality of the energy eigenfunctions

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \psi(E,n)\psi^*(E,m)dE = \frac{1}{2\pi i}\int_{\mathcal{C}} \psi(z,n)\psi^*(z,m)\frac{dz}{z\Delta}, \tag{4.33}$$

and the expression (4.31) for the radial wavefunction in terms of the auxilliary functions $\psi^0(z,n)$, it follows that $\psi(z,n)$ is orthogonal to all the polynomials $\psi^0(z,n)$ with $m > n$[4].

Substituting (4.31) into the orthogonality relation (4.33) yields the Marchenko equation (4.2)

$$K(n,m) + Q(n,m) + \sum_{m=n+1 \leq 2N-(n+3)}^{2N-(n+3)} K(n,p)Q(p,m) = 0, \tag{4.34}$$

where the kernel is

$$Q(n,m) = \frac{1}{2\pi i}\int_{\mathcal{C}} [1 - s(z)]z^{n+m}\frac{dz}{z\Delta} - \frac{\delta_{mn}}{\Delta}. \tag{4.35}$$

Substituting equation (4.29) for the full Jost functions $\chi^{\pm}(z,N)$ into the discrete Schrödinger equation and using the fact that the free Jost functions $\chi^{0\pm}(z,N)$ satisfy the free Schrödinger equation, we find

$$V(n) = -\frac{\hbar^2}{m}\frac{K(n,n+1) - K(n-1,n)}{2\Delta}. \tag{4.36}$$

[4]This property of $K(m,n)$ is analogous to that possessed by a triangular matrix, and hence is called *triangularity*. It is crucial to the development of a solution to the inverse scattering problem, and the difficulty of finding a similar property in $d > 1$ has retarded the solution of higher dimensional inverse problems.

In the continuum limit (recalling that $z^m = h_l^+(km\Delta) \to h_l^+(kr)$, etc.) the discrete equations (4.34), (4.35) and (4.36) become the equations of the Marchenko method (4.2), (4.1) and (4.3).

4.3 Discussion

We have seen that given any scattering amplitude $s_l(E)$ for a single angular momentum l, we can use a Marchenko inverse scattering method to find a local, rotationally symmetric potential $V(r)$ that produces it. The inverse scattering problem we would really like to solve for the sake of finding potentials that produce the few-body automaton of our choice is somewhat more challenging.

First, the preferred dimension is $d = 2$, not $d = 1$, which is less attractive because there are few really interesting one-dimensional CA, or $d = 3$, which makes fabrication enormously more difficult, and would therefore preempt one the ultimate goal of making a connection with experiment. Second, the restriction that $V(\vec{r})$ be rotationally-symmetric restricts our ability to choose computationally interesting few-body as the generators of $\widehat{S}$. Finally, we would really like to solve the two-dimensional, non rotation-invariant inverse scattering problem for *multichannel* scattering; that is, for the case where the scattered entities could change their internal state, as well as their momentum vector, when scattered. Permitting the scattered particles to change internal state opens up the widest variety of computationally interesting possibilities.

Somewhat surprisingly, the development of the inverse scattering methods we need may be near. In just the last decade, R. G. Newton has published a series of papers[122, 123, 124, 125] laying the foundation for solving the single-channel but not necessarily rotation-invariant inverse scattering problem in $d > 1$. And in the last four years, Cheney[32] has addressed the problem of logarithmic singularities peculiar to the general inverse scattering problem in $d = 2$. Once these results are generalized to the multichannel case, it should be possible to derive explicit interaction potentials for specific few-body automata in the manner outlined here. Then we will be able to see if such potentials can be realized experimentally.

In the meantime, there is at least one possible application of inverse scattering methods that may already be amenable to experimental realization. Undoubtedly, the case that has seen the most applicability is not the three-dimensional rotationally-invariant case described in the last two sections, but the strictly one-dimensional case on the entire real line $-\infty < x < \infty$. It has been used to diagnose irregularities in transmission lines from their measured reflection coefficient $r(k)$, and to determine the ionization density of the ionosphere from the time-profile of a pulse reflected back to its earth-bound transmitter.

It may be feasible to apply similar methods to systems like quantum wires, whose novel behaviors have recently excited the interest of physicists and may someday lead to computational applications. Briefly, one might first employ the Landauer formula

$$G = \frac{e^2}{\pi\hbar}\frac{T}{R} = \frac{e^2}{\pi\hbar}\frac{1-|r|^2}{|r|^2} \tag{4.37}$$

to relate the measured conductance G of a one-dimensional conductor to the modulus of the reflection coefficient r. If the phase of r can be measured or deduced by methods analogous to those used in three-dimensional scattering[121], then it should be possible in principle to apply such techniques in order to obtain the profile of the potential produced by the scattering impurities.

The Landauer formula (4.37) applies to a single spin-1/2 electron channel. That is, it describes a two-probe measurement in which a single transverse mode propagates between the two leads. to a single transverse mode but both spin projections. If such an inversion procedure hoped to make contact with experiment, it would have to deal directly with multiple channels[5]. Inverse scattering methods for one-dimensional, multichannel systems have been developed in the last few years[154], but subtleties in the correct definition of the multichannel $\widehat{S}$ matrix[23, 137] probably push such investigations to the edge of what is currently feasible. I do not know if it is practical to make the conductivity measurements needed to determine $r(k)$, or if the potential

[5]For actual experimentally accessible systems, the number of transverse modes present is apparently of order 30[40].

$V(x)$ obtained from the inverse scattering procedure is of sufficient scientific interest to justify the effort.

Chapter 5

A Quantum Cellular Automaton that Occurs in Nature

Finding quantum-mechanical models of computation is not just an important part of Landauer's philosophical program for a self-consistent theory of physical computation. It also bears indirectly on the practical goal of exploiting quantum effects for practical applications. Device physicists have already produced structures in which the quantum-mechanical phase coherence of the electron is maintained over thousands of unit cells[12, 39]. So it is not too soon to begin considering the compatibility of quantum mechanics and computation.

One strategy for finding ways to apply quantum devices to computation is to first find a touchstone model, a model simple enough to be solvable, but complex enough that it exhibits the interplay of quantum effects and computation. Understanding this interplay is a prerequisite if we are to systematically identify the obstacles and opportunities for practical applications.

This chapter describes how a cellular automaton of 2-body type can be used to construct a model of parallel "quantum computation" that is both physically realistic and exactly solvable. The model consists of a time-independent Hamiltonian that provides all of the physical mechanisms—the synchronization network—required by a generic, asynchronous 2-body CA. Only a small physical system, the internal structure of each cell, remains to be specified. In Chapter 2, we have shown that the internal structure needs no more than six bits (six spin-1/2 particles) to attain computation universality.

The central result, from which the thesis takes its title, is the formulation of the first exactly solvable model of "quantum computation". More specifically, I show that not only is the 2-body class of cellular automata computationally powerful enough to simulate any possible computer, it is also physically simple enough to be identified with an exactly solvable quantum spin model: the Lieb-Shultz-Mattis model[104, 117], sometimes known as the one-dimensional XY model.

The exact solution lets us obtain a rigorous picture of how computational properties, like the computation rate, are related to physical properties, such as the excitation energy.

Unlike previous models of quantum computation, which have been roundly criticized by Landauer[96] and others for their Rube-Goldberg-like Hamiltonia, the model proposed here has a form familiar to physicists. At very low temperatures, the Hamiltonian of the $1D$ XY model, to which our model is very closely related, may even describe experimentally accessible systems. About twenty years ago, it was shown experimentally that the $1D$ XY model accurately describes heat capacity and transverse electric susceptibility of praseodymium trichloride[65] and praseodymium ethyl sulphate[52] at temperatures near 1K.

5.0.1 What is 'Quantum Computation'?

Since the methods and key problems of quantum computation are not widely known, §5.0.1 is devoted to a brief review and §5.0.2 gives a short history, restricted to those previous contributions on which our own depends.

How is a binary variable represented?

Over the past decade, there has been an increasing interest in models of "quantum computation"[13, 14, 15, 44, 156]—that is, models in which the information-bearing degrees of freedom obey quantum-mechanical rather than classical-mechanical equations of motion. These models are constructed from ordinary models of binary computers—models like the Turing machine–by making the straightforward association

$$\{0,1\} \rightarrow \{|\downarrow\rangle, |\uparrow\rangle\} \tag{5.1}$$

between binary values and the spin-projection states of spin-$\frac{1}{2}$ particles.

Once this identification is made, any logical function of n binary variables $\{p_i\}$ can be written as an operator $\widehat{F}$ consisting of a sum of products of the $2n$ raising and lowering operators,

$$\sigma^{+}_{p_i}|m_z\rangle = \delta_{m_z,\downarrow}|\uparrow\rangle_i \tag{5.2}$$

and

$$\sigma^{-}_{p_i}|m_z\rangle = \delta_{m_z,\uparrow}|\downarrow\rangle_i, \tag{5.3}$$

on the Hilbert space $\mathcal{H}^{(n)}_{1/2}$ of n spin-$\frac{1}{2}$ particles.

How is a logic gate represented?

For a given logic gate, the corresponding operator $\widehat{F}$ is easiest to write down if we first define two kinds of auxiliary operators that perform simple logical operations on the basis states $|\phi\rangle \in \{|m_z^{(0)}, m_z^{(1)}, ..., m_z^{(n-1)}\rangle\}$, $m_z^{(i)} \in \{|\downarrow\rangle, |\uparrow\rangle\}$. The first type is a "checking" operator $\widehat{C}_\phi$. On a general state $|\psi\rangle \in \mathcal{H}^{(n)}_{1/2}$, $\widehat{C}_\phi$ checks for the presence of $|\phi\rangle$ and returns the uniform spin-down state with the same amplitude: For any basis state $|\phi\rangle$,

$$\widehat{C}_\phi|\psi\rangle \equiv \langle\phi|\psi\rangle|\downarrow,\downarrow,...\downarrow\rangle. \tag{5.4}$$

The second set of operators, "lookup" operators, map the uniform spin-down state to each of the basis states:

$$\widehat{L}_{\phi'}|\downarrow,\downarrow,...\downarrow\rangle \equiv |\phi'\rangle \tag{5.5}$$

Together, the checking and lookup operators can be composed to construct an operator representation for any lookup-table—that is, for any logic gate with n binary inputs and outputs.

As a simple example, consider the NAND gate on two binary variables p and q.

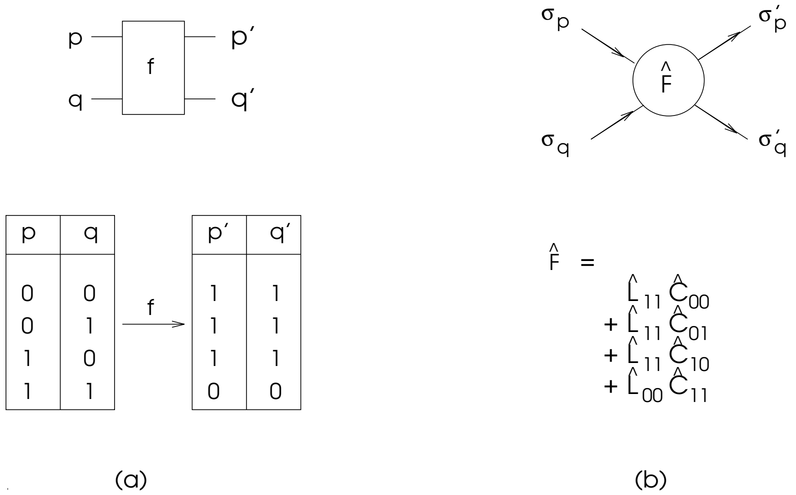

Figure 5-1: (a) Binary representation of the logical operation f = NAND, (b) Representation of f = NAND as an operator $\widehat{F}$ on two spin-$\frac{1}{2}$ particles, σ_p and σ_q.

Figure 5-1a shows the the schematic (top) and binary lookup-table for an 2-input/2-output function $f(p,q)$, both of whose outputs are the logical NAND of its inputs.

In Figure 5-1b, two $s = \frac{1}{2}$ quantum spins, σ_p and σ_q, represent p and q quantum mechanically. The operator $\widehat{F}$ that represents $f(p,q)$ is composed of a sum of products of checking and lookup operators on the two-spin system. The checking and lookup operators, in turn, are easily constructed in terms of the familiar raising and lowering operators. For example,

$$\widehat{C}_{00} = \sigma_p^- \sigma_p^+ \sigma_q^- \sigma_q^+, \tag{5.6}$$

where we have implicitly used the association (5.1) in denoting $\widehat{C}_{\downarrow\downarrow}$ as $\widehat{C}_{00}$. Lookup operators such as $\widehat{L}_{11}$ have an equally simple form,

$$\widehat{L}_{11} = \sigma_p^+ \sigma_q^+. \tag{5.7}$$

In the bottom of Fig. 5-1b, each term is a mere transliteration, using the raising and lowering operators $\sigma_p^\pm$ and $\sigma_q^\pm$, of one row of the truth-table for f in Fig. 5-1a. Since the input states in the lookup table are an exhaustive set of mutually exclusive possibilities and the basis states form a complete orthonormal basis on $\mathcal{H}_{1/2}^{(2)}$, we can think of the operator $\widehat{F}$ as describing the one-step evolution of a "quantum NAND gate".

How is a sequence of logical operations represented?

Note, however, that because the NAND function is not invertible, the operator $\widehat{F}$ is not invertible and hence not unitary. Therefore, if we hope to use this technique of associating spin operators with binary computations, we must restrict ourselves to logically reversible operations. Recall from §1.1 that Fredkin and Margolus have established that reversibility does not preclude us from attaining computation universality. If we confine ourselves to logically reversible operations, the corresponding operators will be unitary. Since any computation on n binary variables can be accomplished by some sequence of logically reversible operations, any computation can be represented as the action of some unitary "time evolution" operator $\widehat{U} = \prod_m \widehat{F}_m$ on a lattice of n spin-$\frac{1}{2}$ particles. The operator $\widehat{U}$ describes the evolution of a quantum computer.

The quantum computer is "programmed" by initializing it in the basis state corresponding to the binary inputs string. The result of the computation can in principle be read by performing appropriate measurements on the spin variables; possible techniques for doing this are described in more detail in refs. [51, 131]. Since we are interested here in finding a quantum-mechanically realistic description of the computation process, rather than the measurement process, we will not discuss initialization and read-out further.

The operator $\widehat{U}$ was constructed solely to have a specified computational behavior. At no point did we attempt to insure that it was physically realistic. In order to see more clearly what kind of physical system the evolution operator $\widehat{U}$ corresponds to, we need to find the Hamiltonian $\widehat{H}$ that has $\widehat{U}$ as its Schrödinger evolution operator. The structure of $\widehat{H}$ will tell us how what kinds of interactions $\widehat{U}$ requires, and we will be able to compare these interactions with realistic ones.

Integrating the time-dependent Schödinger equation, the hermitian operator $\widehat{H}$

that satisfies

$$\widehat{U}(t) = 1 - \frac{i}{\hbar} \int_0^t \widehat{H}(t')\widehat{U}(t')dt', \tag{5.8}$$

can be identified as the Hamiltonian of the *quantum computer*.

So, at least formally, it is not difficult to recast models of computation in the language of quantum mechanics. However, the burden remains on us to show that the resulting models are not merely empty transliterations from the theory of computation, producing "cleverly chosen Hamiltonians[99]" which have little hope of being physically realized.

Physically realistic models of quantum computation are intrinsically hard to construct because the form of the Hamiltonian $\widehat{H}$ depends directly on the form of the evolution operator $\widehat{U}$. The form of $\widehat{U}$, in turn, is necessarily dictated by the form of the computer being emulated, and not by considerations of physical realism or simplicity.

A physically credible model of quantum computation should have virtues comparable to those possessed by Fredkin's billiard ball model of classical-mechanical computation. The model should have simple, physically realistic interactions. It should attain its computational power from the careful choice of initial conditions, boundary conditions, and from an apt computational interpretation of its information-bearing degrees of freedom—not from an unnatural, concocted complexity residing in its interactions.

Models of quantum computation can be made more physically realistic by endowing their Hamiltonians with three particular properties: time-independence, locality, and simple few-body interactions. In this chapter we exhibit and solve a model of quantum computation with all three properties. But first, we review the progress made by three previous investigators: Benioff, Feynman and Margolus, on whose efforts our own contribution is built.

5.0.2 Brief History of Quantum Computation

Prehistory

Even before the first real model of quantum computation, Benioff's model, many people working on models of reversible computation realized that a brute-force model of quantum computation was possible. Any deterministic classical computer C is completely defined by the set of all sequences of states—*computational orbits*—that it traverses on all possible input programs. If all of these orbits are known ahead of time, a quantum mechanical Hamiltonian $\widehat{H}$ describing the computer can be found as follows. Suppose the computer has a state set $\{s_n\}(n = 1, ..., N)$. By associating a basis state $|\phi_n\rangle$ with each state s_n of the computer, it is trivial to construct a unitary time evolution operator $\widehat{U}$ that simulates C on the Hilbert space spanned by $\{|\phi_n\rangle\}$. Since U is unitary, and hence normal (i.e., commutes with its adjoint), there is some unitary unitary operator $\widehat{V}$ that diagonalizes it,

$$\widehat{U}(\tau) = \widehat{V}^\dagger \begin{pmatrix} e^{i\lambda_1} & 0 & ... & 0 \\ 0 & e^{i\lambda_2} & ... & 0 \\ \vdots & \vdots & ... & \vdots \\ 0 & 0 & ... & e^{i\lambda_N} \end{pmatrix} \widehat{V}. \tag{5.9}$$

If we take

$$\widehat{H}\tau/\hbar \equiv -\widehat{V} \begin{pmatrix} \frac{i}{\tau}\lambda_1 & 0 & ... & 0 \\ 0 & \frac{i}{\tau}\lambda_1 & ... & 0 \\ \vdots & \vdots & ... & \vdots \\ 0 & 0 & ... & \frac{i}{\tau}\lambda_N \end{pmatrix} \widehat{V}^\dagger, \tag{5.10}$$

then $\widehat{H}$ satisfies $\widehat{U}(\tau) = e^{-i\widehat{H}\tau/\hbar}$.

However, even if the computation described by $\widehat{U}$ is simple, the resulting $\widehat{H}$ is in general highly *nonlocal*, that is, it involves couplings between distant spins. Furthermore, as Benioff[131], Bennett [95] and Margolus[113] have all pointed out, since we need to perform all possible computations on all possible initial states before we can even write down the hamiltonian (5.10), this construction is a look-up table rather than a computer.

Benioff

Benioff[13, 14] was the first person to propose a model of quantum computation that did not have to precompute all possible computational orbits. He accomplished this by defining a quantum spin system whose evolution simulated that of a Turing machine.

A Turing machine (Fig. 5-2) is a mathematical abstraction drawn (albeit implicitly) from computers whose internal mechanisms are accurately described by classical mechanics. The abstract model consists of two parts: an infinite tape T and a head M. The tape is divided into discrete cells, each of which is either blank or contains exactly one symbol from a finite set Σ. The head can perform four operations: read the symbol from the cell that is directly underneath it, write a symbol in that cell, move left or right by one cell, and change its state q to one of a finite number of values. The time evolution of a Turing machine is specified by a table that exhaustively describes what action the head takes depending upon the symbol it is currently scanning and its own internal state. Perhaps the most significant result of the theory of computation is that for any computation, there is some Turing machine that can simulate it. Because it combines this powerful "universal" property with structural simplicity, Benioff selected it as his target for constructing a quantum computer.

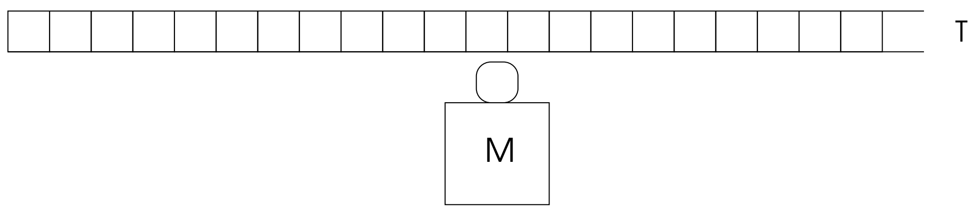

Figure 5-2: Turing machine model[73, 118].

He proceeded by systematically translating the binary operations of a Turing machine into a unitary operator on a system of quantum spins. His model[13, 14] has the virtue of requiring only short-range interactions but, unfortunately, the corresponding Hamiltonian has an unnatural, concocted appearance. It has a time-dependence that results from alternately turning on and off three distinct Hamiltonians

$$H_1 = \sum_{l \in Q} \sum_{v} \sum_{k=-N}^{N} \sum_{k'=1}^{N} P_l \otimes P_k \otimes P_{vk} \otimes P_{k'} \otimes U_{bl}^{Mk'} \otimes U_{bv}^{Ck'} \otimes U_{bk}^{Pk'}, \tag{5.11}$$

$$H_2 = \sum_{q \in Q} \sum_{k=-N}^{N} \sum_{k'=1}^{N} U_q^M \otimes U_{qk}^{H_C T} \otimes P_{k'} \otimes P_{qk'}^M \otimes P_{qk'}^C \otimes P_{kk'}^P, \tag{5.12}$$

and

$$H_3 = 1 \otimes 1 \otimes 1 \otimes B \otimes 1 \otimes 1 \otimes 1 \otimes, \tag{5.13}$$

where (in the notation of Benioff[13]) the operators P, U and B are various projection and permutation operators that act on the spins shown in Fig. 5-3.

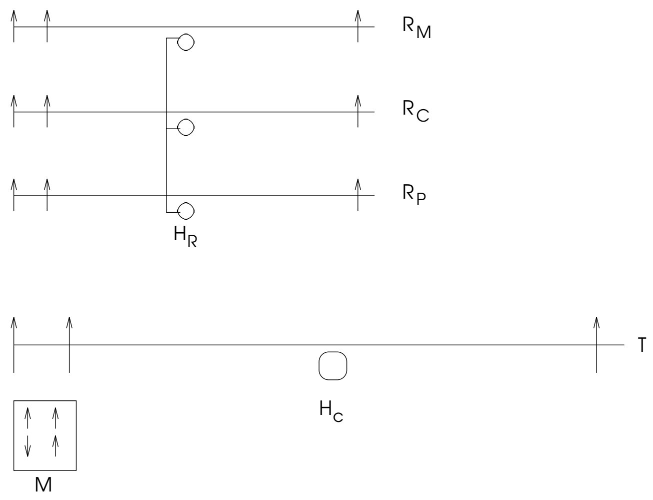

Figure 5-3: Benioff's quantum-spin Turing machine[13]

In the figure, M is a small spin system whose function is to record the state of the head of the Turing machine. H_c is a single spin that marks the position of the head on the tape T, which is implemented as a linear array of spins. The state of each spin corresponds to a symbol on the tape of the Turing machine. The auxiliary tapes R_M, R_C and R_P are used to keep a record of the states of M, T and the position of the head H_C on the tape T.

The Hamiltonian H_1 measures the state of the head M, the position of the head H_C, and the tape symbol at that position and records the results of these measurements on the auxiliary tapes R_M, R_P and R_C, respectively. The Hamiltonian H_2 carries out the appropriate transformation of the the state of the head M, changes the tape symbol on T, and moves the head H_C. Finally, H_3 shifts the record head H_R to the next position.

Suffice it to say, Hamiltonians with the form (5.11)–(5.13) are hard to come by in the real world.

Feynman

Feynman's principal goal was to construct a quantum-spin computer that replaced the awkward, three-phase, time-dependent Hamiltonian of Benioff with a time-independent one. Unlike Benioff, who took as his evolution operator a unitary operator $\widehat{F}$ that described one step of the forward evolution he wanted, Feynman[51] made a very different use of the unitary operator $\widehat{F}$ that describes a single forward step of his computation. He formed the hermitian operator $\widehat{H}_{\text{comp}} = \widehat{F} + \widehat{F}^{\dagger}$ and took that as the Hamiltonian of his quantum computer. Since $\widehat{F}$, which moves the computation forward, and $\widehat{F}^{\dagger}$, which moves it backward, now occur symmetrically in $\widehat{H}$ (and hence in the evolution operator), it is no longer obvious that the computation can proceed forward any more often than backward. Superficially, at least, this looks like a model of diffusive computation.

The problem of diffusive computation is a very serious one. At a clock rate of 1 GHz, a computation that would take 12 days on a computer making one forward step on each cycle would take an expected 100×10^9 years to complete on a diffusive computer working at the same rate. Consequently, it is reasonable to question whether diffusive computation should be considered "effectively" universal computation or not.

Feynman's solution was to couple the part of his system that performed the computation, the part described by $\widehat{H}_{\text{comp}}$, to another subsystem, an infinite line of "clock spins" (see Fig. 5-4) that synchronizes the computation. The interactions among clock spins can be described by a simple one-dimensional Heisenberg Hamiltonian $\widehat{H}_{\text{sync}}$. By initializing the clock spins in a magnon state with positive momentum, he was able to drag the computation forward at a uniform, rather than diffusive, rate.

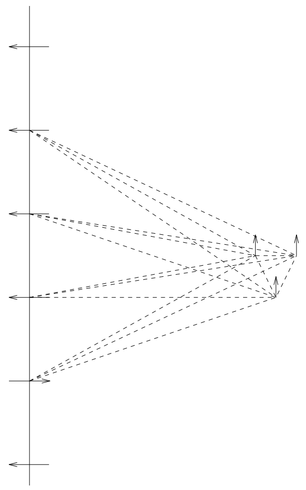

Figure 5-4: In the Feynman model[51], an upward traveling spin wave on the line of "clock spins" (left) drags the three spins of the computational subsystem (right) through the desired sequence of computations at a uniform rate.

Margolus

Although there are no infinite-range interactions in the subsystem performing the computation (the group of three spins in Fig. 5-4), the range of the interactions between the clock spins and the spins in the computational subsystem (shown as dashed lines) becomes infinite as the length of the computation increases.

In order to make a Hamiltonian model of quantum computation that retained the property of time-independence but had no infinite-range interactions, Margolus[112] applied Feynman's clock spin idea to an intrinsically local model of computation, cellular automata. His two-dimensional cellular automaton model[112] succeeded in providing a model in which the Hamiltonian is both time-independent and short-ranged; however, the rate at which useful computation occurs is again diffusive. The problem of diffusive computation is more serious in Margolus's cellular automaton model than in Feynman's model of serial computation since there is no apparent way to modify Feynman's spin-wave solution so that it can be applied here.

Some progress was made on the diffusion problem for a one-dimensional version of this quantum cellular-automaton. Margolus and I were able to show[115] that the computation-rate operator defined in [112] commutes with the Hamiltonian. This suggested that there might be some hope of finding of an initial state analogous to the spin-wave solution in Feynman's construction, a state that would insure that the computation progressed at a uniform rate.

However, we were not able to calculate any of the eigenvalues or eigenstates of the computation-rate operator[115] or even show them to be nonzero. In the absence of at least one positive eigenvalue, one cannot assert that the problem of diffusion is overcome. Furthermore, one is prevented from addressing such questions as how the maximum computation-rate $\gamma_{\max}(M)$ scales with system size M, that is, how much additional processing one obtains by adding as additional cells are added to the cellular automaton.

In this chapter, we show how to do better. We present the first model of quantum computation that satisfies Landauer's criteria, stated below, for the realizability of a model.

5.0.3 Plan of this Chapter

> *Most physicists would very likely be satisfied by a Hamiltonian which has a sufficiently familiar and simple structure, e.g., a Hamiltonian for a reversible cellular automaton which resembles an Ising lattice Hamiltonian. But it really is a very drastic assumption about the real world to assume that even simple Hermitean operators are* physically realizable *Hamiltonians.* Landauer[99]

There are two parts to Landauer's criteria: reasonableness and actual existence. The first criterion, then, is that the Hamiltonian should be recognizable as some idealized theoretical model—preferably a model originally designed to capture some aspect of a real system.

In this chapter we present a model of quantum computation that satisfies Landauer's first criterion—all of the long-range properties of the system are described by a Hamiltonian $\widehat{H}_{\mathrm{sync}}$ identical to that of the one-dimensional XY model of Lieb, Shultz and Mattis[104].

The model of parallel quantum computation described by $\widehat{H}_{\mathrm{sync}}$, unlike the cellular automaton models of refs. [112] and [115], is exactly solvable in the sense that closed-form expressions can be found for the eigenvalues and eigenvectors of both the Hamiltonian and computation-rate operators for systems of arbitrary size. The exact solution permits us to give the first analytical treatment of the interplay of quantum-mechanical evolution on computation properties, such as computation rate.

We find that the maximum achievable rate of parallel computation $\gamma_{\max}(M)$ scales linearly with system size M. However, the linear coefficient is $2/\pi$, rather than the ideal limit of 1. We give a simple physical argument based on a variational principle to show that for any quantum computer the ideal limit must be unachievable.

The simplicity of the anisotropic antiferromagnetic Hamiltonian, $\widehat{H}_{\mathrm{sync}}$, thus provides a model of quantum computation that possesses all three of the physical properties desired: it is spatially local, time-independent, and simple enough to be considered physically realistic. As such, it represents another step toward the goal of making our models more physical.

By sheer good fortune, it turns out that the quantum cellular automaton presented here is not only simple and familiar; to a large extent, it also meets the second of Landauer's criteria, physical realizability. There appear to be real experimental systems which, at sufficiently low T, are accurately described by our Hamiltonian. The experimentally observed heat capacity C_V and transverse electric susceptibility $\chi_\perp$ of both praseodymium trichloride ($PrCl_3$) and praseodymium ethyl sulfate ($Pr(C_2H_5SO_4)_3 \cdot 9H_2O$,) near $T \approx 1$K agree with the theoretical values predicted by $\widehat{H}_{\mathrm{sync}}$ to within about 1%.

5.1 The Model

5.1.1 Hamiltonian of the Quantum Cellular Automaton

As we have seen, quantum computers are defined by translating some classical model of computation, such as a Turing machine or cellular automaton, into a Hamiltonian $\widehat{H}$ acting on a Hilbert space $\mathcal{H}$. In addition to making our Hamiltonian local, we want its interactions to be as simple as possible. One strategy for achieving this is to make the interactions involve as few particles as possible. To this end, we base our model of quantum computation on a 2-body automaton. Recall from Chapter 2 that in this form of cellular automaton, cells interact at any given time with either their left or right neighbors but never with both simultaneously. The space-time diagram for the evolution of a 2-body cellular automaton was shown in Fig. 2-4 (reproduced here as Fig. 5-5).

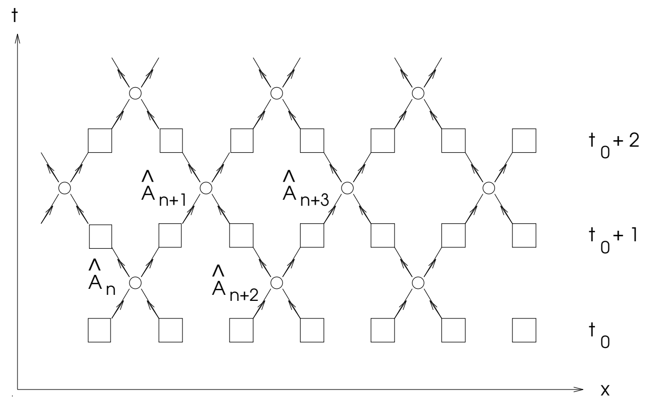

Figure 5-5: Time evolution of a 2-body CA in one dimension is identical to that of a one-dimensional lattice gas.

The pattern of time evolution of a 2-body cellular automaton (Fig. 5-5) can easily be translated into that of a one-dimensional lattice of spins. Suppose each cell n of the automaton can take on one of $|Q|$ states. Then, for any J such that $2^J \geq |Q|$, the state of a cell can be represented by the state of a group of J spin-$\frac{1}{2}$ particles, $\widehat{\tau}_{n,j}$ with $0 \leq j \leq J-1$.

In the same manner as we constructed an operator that represented the NAND gate, any 2-body cellular-automaton rule f_n having the pair of sites $(n, n+1)$ as its inputs and outputs can represented as a sum $\widehat{F}_n$ of products of raising and lowering operators on the $2J$ quantum spins $\widehat{\tau}_{n,j}$, Because cellular automata are spatially uniform, the function f is independent of n.

5.1.2 Asynchrony

The goal of keeping the Hamiltonian completely local compels us to consider asynchronous computations. By asynchronous we mean that at different locations in the cellular automaton different amounts of progress will have been made at any given time.

In order to construct a time-independent Hamiltonian, we employ Feynman's method, taking our Hamiltonian to be $\widehat{H} = \widehat{F} + \widehat{F}^{\dagger}$. Since we want $\widehat{H}$ to be local, we must take the forward-step operator $\widehat{F}$ as a sum of local operators $\widehat{F}_n$, each involving raising and lowering operators for nearby spins. That is, we take $\widehat{F} = \sum_n \widehat{F}_n$ rather than $\widehat{F} = \prod_n \widehat{F}_n$. The $\widehat{F}_n$ carry out the cellular automaton rule on a single neighborhood of the 2-body rule; that is, on the spins at sites n and $n+1$. So the requirements of time-independence and locality lead us to an evolution operator

$$\widehat{U} = e^{i(\sum_n \widehat{F}_n + \sum_n \widehat{F}_n^{\dagger})/\hbar}. \tag{5.14}$$

Expanding the exponential, we obtain a sum of products of the forward ($\widehat{F}_n$) and backward ($\widehat{F}_n^{\dagger})/\hbar$) evolutions at various locations n. So $\widehat{U}$ inevitably generates nonvanishing amplitudes for configurations in which some regions of the cellular automaton have experienced more updates than others—that is, configurations of an asynchronous computation.

Many, but not all, such configurations will spoil the computation irremediably. For example, in Fig. 5-6(a), the configuration in which a single pair has been updated once, and no other cells have been updated at all, is not disastrous, since neighboring cells still have a chance to catch-up before this pair of cells changes state again. However, if the same pair updates twice before its left and right neighbors have updated once, as in Fig. 5-6, the computation is beyond repair since the states of each member of the pair might well have been different if their left and right neighbor pairs had updated first, as they would have in the synchronous computation. In short, the pattern of interactions shown in Figs. 5-5 and 5-6, the pattern that defines a 2-body automaton, is compatible with asynchronous computation so long as neighboring cells differ in the net number of forward updates they have experienced by no more than one. In Fig. 5-6 this has a simple pictorial interpretation. If the local times (solid cells) are thought of as monomers of a polymer extending along the x-axis, then an asynchronous configuration is allowable if the polymer is unbroken (as in Fig. 5-6a) and unallowable if (as in Fig. 5-6b) it is broken.

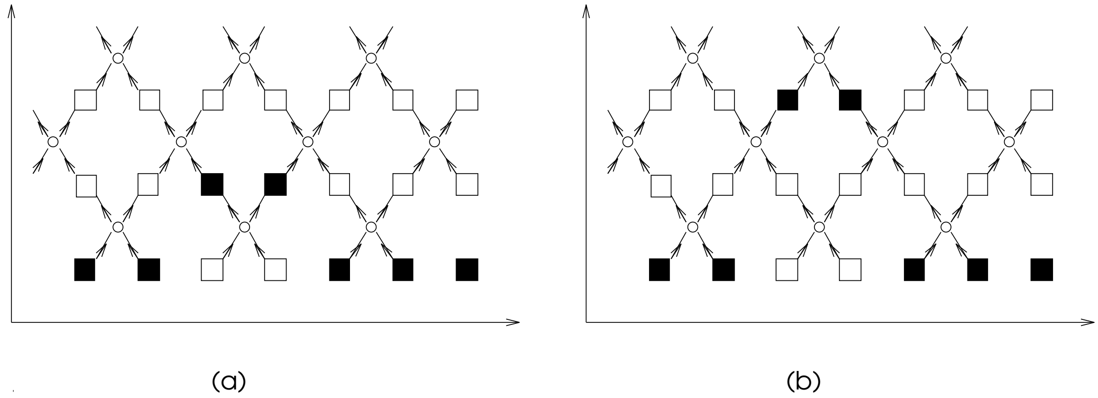

Figure 5-6: (a) An allowable asynchronous configuration—that is, one which preserves the integrity of the computation, and (b) one which must be prohibited.

Fortunately, it is possible to add a constraint which will insure that $\widehat{U}$ gives only the allowable asynchronous configurations a non-zero amplitude. Toffoli[143] has given a general technique for preserving the integrity of a computation in the face of asynchrony by adding a few degrees of freedom to the system. In our case a single bit per site suffices. This additional *synchronization bit* is represented by a spin-$\frac{1}{2}$ operator $\widehat{\sigma}_n$ at each site. The synchronization bits are coupled to one another in such a way that the places that get ahead wait for their neighbors to catch up (see Fig. 5-8). They insure that no pair of cells are updated unless they contain data corresponding the same moment of the synchronous computation. In operator language, they insure that $\widehat{F}_n$ and $\widehat{F}_n^\dagger$ annihilate any configuration unless the neighboring pair of cells they operate on have experienced exactly the same number of net forward steps.

One can synchronize the asynchronous computation in many different ways[143]. Since all choices simulate the same synchronous computation, the choice has no effect on what is ultimately computed. That is, the detailed choice of how we preserve the integrity of the computation may be roughly likened to the choice of a gauge in a system with gauge invariance, where the choice has no effect on the resulting behavior of physical observables. And just as an apt choice of gauge can enable us to solve an otherwise intractable problem, we will find that an apt choice of synchronization "gauge" leads us to a Hamiltonian different from that considered in [115], but computationally equivalent and exactly solvable.

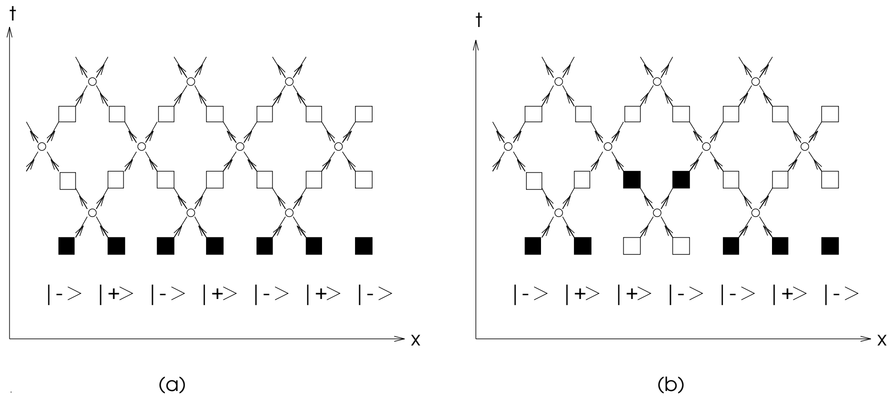

Figure 5-7: The synchronization bit $|\sigma\rangle \in \{|-\rangle, |+\rangle\}$ enforces a constraint on the asynchronous configurations so that $\widehat{U}$ does not corrupt the overlying computation.

The "gauge" we choose for the the synchronization bit σ_n is shown in Fig. 5-7. Figure 5-7a shows a "flat" configuration, in which each cell has been updated an equal number of times (horizontal row of solid cells). The corresponding configuration of the synchronization spins σ_n is an alternating product of states with $m_z = \pm 1$. The synchronization constraint is enforced by requiring the synchronization spins at sites $(n, n+1)$ to be in the state $|-\rangle_n|+\rangle_{n+1}$ before the operator $\widehat{F}_n$ is permitted to update sites $(n, n+1)$ in the 2-body automaton.

When the update is allowed, the configuration of the synchronization spins changes according to

$$|-\rangle_n|+\rangle_{n+1} \to |+\rangle_n|-\rangle_{n+1}. \tag{5.15}$$

Similarly, the backward step operator $\widehat{F}_n^\dagger$ is only permitted to operate when the corresponding synchronization spins are in the state $|+\rangle_n|-\rangle_{n+1}$. Whenever $\widehat{F}_n^\dagger$ acts, the configuration of the synchronization spins changes according to

$$|+\rangle_n|-\rangle_{n+1} \to |-\rangle_n|+\rangle_{n+1}. \tag{5.16}$$

It is straightforward to verify that these rules preserve the integrity of the computation despite the fact that $\widehat{U}$ generates nonvanishing amplitudes on configurations in which some regions of the cellular automaton have been updated more often than others.

Algebraically, the synchronization spins are incorporated into the dynamics by taking the forward computation operator to be

$$\widehat{F} = \sum_{n=0}^{\frac{(M-1)}{2}-1} \sigma_{2n}^+\sigma_{2n+1}^-\widehat{A}_{2n} + \sigma_{2n+1}^+\sigma_{2n+2}^-\widehat{A}_{2n+1} = \sum_{n=0}^{M-1} \sigma_n^+\sigma_{n+1}^-\widehat{A}_n \tag{5.17}$$

where the σ_n^-, σ_n^+ are the usual lowering and raising operators for the synchronization spin at site n. The full Hamiltonian for the quantum cellular automaton then becomes

$$\widehat{H} = \sum_{n=0}^{M-1} \sigma_n^+\sigma_{n+1}^-\widehat{A}_n + \sigma_n^-\sigma_{n+1}^+\widehat{A}_n^\dagger \tag{5.18}$$

5.1.3 Effective Long-range Hamiltonian

The Hamiltonian (5.18) acts on a Hilbert space $\mathcal{H} = \mathcal{H}_{\text{sync}} \otimes \mathcal{H}_{\text{comp}}$ composed of the synchronization and computation parts. Since it determines all the long-range properties (such the computation-rate), we will mostly be concerned with the synchronization part.

The evolution on $\mathcal{H}_{\text{comp}}$ is given by the operator $\widehat{F}_n$ that describes the cellular automaton rule. The details of its construction and evolution are a matter of computational, but not physical, interest. In ref. [22] (see Chapter 2) we have shown that a one-dimensional cellular automaton can perform universal computation with as few as six bits per site[22].

The long-range properties we are interested in, like the rate of computation and its dependence on the total energy, depend only on the evolution of the synchronization system, and not on the details of the computation riding on top of it. The evolution of the synchronization subsystem can be considered separately because its dynamics is *data blind.* This can be seen from the fact that the Hamiltonian (5.18) has a special form. Since operator products $\sigma_n^+\sigma_{n+1}^-$ and $\sigma_n^-\sigma_{n+1}^+$ in (5.18) each multiply the operators $\widehat{A}_n$ and $\widehat{A}_n^\dagger$ which describe, respectively, the *entire* forward and backward computations, the transitions made by the synchronization spins are completely independent of the transitions—the terms in the sum $\widehat{A}_n$—experienced by the computational degrees of freedom. Consequently, we henceforth consider only the effective Hamiltonian on the synchronization spins. We will see that it captures all of the effects of quantum mechanics on the computation-rate of our cellular automaton model.

The effective Hamiltonian for the evolution on $\mathcal{H}_{\rm sync}$ is

$$\widehat{H}_{\rm sync} = \sum_{n=0}^{M-1} \sigma_n^+\sigma_{n+1}^- + \sigma_n^-\sigma_{n+1}^+. \tag{5.19}$$

We recognize the effective Hamiltonian $\widehat{H}_{\rm sync}$ as an extremely anisotropic Heisenberg antiferromagnet

$$\widehat{H} = J_\perp \sum_{n=0}^{M-1}(\sigma_n^x\sigma_{n+1}^x + \sigma_n^y\sigma_{n+1}^y) + J_z \sum_{n=0}^{M-1} \sigma_n^z\sigma_{n+1}^z \tag{5.20}$$

with exchange couplings $J_\perp = 1$ and $J_z = 0$. This model is sometimes called the one-dimensional XY model[104]. From the identification $\widehat{H} \equiv \widehat{H}_{\rm sync} = \widehat{H}_{1{\rm D}XY}$ we know the energy eigenstates and eigenvalues [117]; however, in this chapter we are principally interested in the spectrum of a different operator, an operator that characterizes the computation rate of the quantum cellular automaton.

5.1.4 Computation Rate Operator $\widehat{\Gamma}(M)$

Definition

Because we wanted a time-independent Hamiltonian, we used Feynman's method, defining $\widehat{H} = \widehat{F} + \widehat{F}^{\dagger}$. The addition of the $\widehat{F}^{\dagger}$ term insures that $\widehat{H}$ is hermitian, but it also means that the time evolution operator $\widehat{U} = e^{i\widehat{H}/\hbar}$ will undo parts of the computation whenever the operator $\widehat{F}^{\dagger}$ acts. Because of the symmetric occurrence of $\widehat{F}$ and $\widehat{F}^{\dagger}$ in $\widehat{U}$, it is not obvious that the computation can occur at other than a diffusive rate. Recall that Feynman solved this problem by adding an infinite line of clock spins to his quantum spin computer. Each clock spin was coupled to *every* spin of the quantum computer in such a way that as the clock spin moved forward, the quantum computer took one step forward in its computation; when the it moved backward, the quantum computer took one step backward in its computation. By initializing the line of clock spins with a spin wave of positive momentum, it was possible to insure that the quantum spin computer made progress at a uniform rate. This established that, despite the presence of $\widehat{F}^{\dagger}$, diffusive computation was not inevitable.

Unfortunately, in order to remove the infinite-range couplings, we are compelled to remove the infinite line of clock spins that supported the positive momentum spin wave. In place of the line of spins, we have only a single spin at each site, and so Feynman's solution is no longer available to us. We must find another way of establishing the possibility of uniform-rate computation in the quantum cellular automaton with Hamiltonian (5.18).

To address this issue, we must explicitly define an operator $\widehat{\Gamma}$ that captures some reasonable notion of the net rate at which the computation is moving forward. Then we need to show that the computation-rate operator $\widehat{\Gamma}$ has an eigenstate $|\gamma\rangle$ with a positive eigenvalue. Finally, we need to show that as the system evolves the rate of computational progress does not decrease to zero. To do this, it suffices to show that $|\gamma\rangle$ is also an eigenvector of $\widehat{H}$, since the eigenvalue $\gamma > 0$ is then a constant of the motion.

To help motivate the definition of $\widehat{\Gamma}$, we define an auxiliary operator $\widehat{N_c}$. On the basis states—states in which each cell has been updated a definite number of times—$\widehat{N_c}$ returns the sum over all cells of the number of forward minus the number of backward steps that have taken place relative to some arbitrarily fixed state. We can think of the computational-rate operator as the time-derivative of this operator

$$\widehat{\Gamma} \equiv \frac{d}{dt}\widehat{N_c} = \frac{1}{i\hbar}[\widehat{H}, \widehat{N_c}]. \tag{5.21}$$

So $\widehat{\Gamma}$ characterizes the net rate at which the computation runs in the forward direction.

Evaluating the commutator, we obtain

$$\widehat{\Gamma}(M) = \frac{1}{i}\sum_{n=0}^{M-1} \sigma_n^+\sigma_{n+1}^- - \sigma_n^-\sigma_{n+1}^+. \tag{5.22}$$

The computation-rate operator characterizes two distinct aspects of how parallel quantum computations scale: temporal and spatial scalability.

The Problem of Diffusion: Temporal Scalability

We will often contrast the undesirability of diffusive-rate computation with the desirability of "uniform-rate" computation. The natural diffusive behavior can be visualized as the diffusion of a closed polymer loop (representing the net computational progress at each site of the periodic array of spins) along a tube (Fig. 5-8). By uniform-rate computation, we mean that after a time mT, the quantum computer has made αm times as much forward progress as it had made after at time T. In terms of the computation-rate operator $\widehat{\Gamma}(M)$, uniform-rate computation will be possible if and only if $\widehat{\Gamma}(M)$ has at least one positive eigenvalue. Uniform rate computation is more than just a convenience. Without it, many simple computations are beyond practical reach. As we have seen in §5.0.2, a computer that makes only diffusive progress is so unhelpful that it scarcely deserves to be considered a computer at all. Consequently, although the quantum cellular automaton model with Hamiltonian (5.19) and computation-rate operator (5.22) is physically-realistic, until we show that

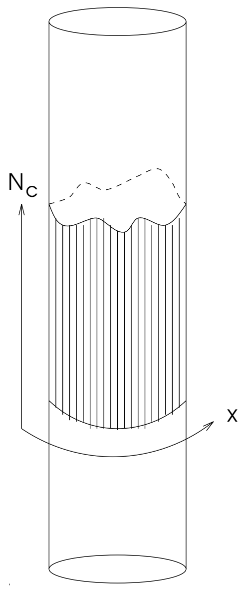

Figure 5-8: Diffusive evolution of the one-dimensional quantum cellular automaton with periodic boundary conditions can be visualized as diffusion of a closed loop on along a tube, where x denotes the spatial extension of the automaton and N_c the amount of computational progress. $\widehat{\Gamma}$ measures the rate of change of the striped area.

$\widehat{\Gamma}$ has a positive eigenvalue, we cannot be certain it should be considered a model of *computation*.

Parallel Computation: Spatial Scalability

The second kind of scalability, spatial scalability, will bring us into contact with some notions that currently interest architects of parallel computers. Here we would like to know how much additional useful computation we obtain as we increase the size of our system.

If we regard each site of the quantum CA as a processor in a parallel computer, the analog of the diffusion problem is the problem of *scalability of parallel computation*. For example, if we are given a parallel computer that is M times as large as another, we might reasonably hope to perform our old computations M times as fast; alternatively we might expect to perform a computation M times as large in the same time T_0. If we discover that we can only perform a computation $\sqrt{M}$ times as large in time T_0, then we find ourselves in a situation analogous to that of diffusive computation.

There is an extensive theory of parallel computation[101, 92], but the focus of much of this theory has been on models which implicitly incorporate classical-mechanical assumptions about the nature of interactions between processors. One way to see the point of the model of quantum computation proposed here is to contrast it with one such model, the parallel-random-access-memory (PRAM) model[59, 101]. In the PRAM model, any processor can read the state of any part of memory without delay, but no two processors can simultaneously change the state of the same part of memory. The PRAM model thus abstracts out nearly all notion of the cost of communication. This makes it a particularly useful tool for exploring the inherent parallelism of an algorithm, because it cleanly separates retarding effects caused by non-parallelizable segments of the algorithm from retarding effects due to communication delays in a particular architecture.

Here our interest is just the complement. Once we have satisfied ourselves that computation-universal algorithms of 2-body type exist (see Chapter 2), we can focus on how the quantum couplings—the communication between cells—affect the computation rate, by ignoring the particular algorithm running on the quantum cellular automaton.

Keeping this complementarity of interest in mind, we can adopt some useful concepts from the theory of parallel computation.

Nussbaum and Agarwal[127] have recently given a rigorous definition of the scalability ψ which reflects the cost of interprocessor communication in a given architecture. For an algorithm α and problem of size n, they define the scalability of architecture R to be

$$\psi_\alpha^R(n) = \max_p \left(\frac{T_\alpha^{\mathrm{PRAM}}(n)}{T_\alpha^R(n,p)} \right), \tag{5.23}$$

where p denotes the number of processors, $T_\alpha^{\mathrm{PRAM}}(n)$ is the execution time on a PRAM, and $T_\alpha^R(n,p)$ is the execution time on a p-processor machine with architecture R. We consider the scaled-speedup[62], in which the size n of the problem is assumed to be the same as the machine size M.

To adapt this definition to our interests, we consider the best-case scenario in which α is ideally parallelizable, thereby focusing on the cost of communication rather than the particular algorithm α. This has two consequences. Since α is assumed ideally parallelizable, the PRAM achieves the ideal speedup $T_\alpha^{\mathrm{PRAM}}(M) = 1/M$. Second, the maximum over p in (5.23) is equivalent to the asymptotic limit as $p \to \infty$, so that, setting $p = M$,

$$\lim_{M\to\infty} \psi_\alpha^R(M) = \lim_{M\to\infty} \left(\frac{1}{MT_\alpha^R(M)}\right) = \lim_{M\to\infty} \frac{S_\alpha^R(M)}{M}. \tag{5.24}$$

The quantity

$$\eta \equiv \lim_{M\to\infty} S_\alpha(M)/M, \tag{5.25}$$

the speedup achieved by M processors relative to the ideal speedup, is called the *asymptotic parallel efficiency*.

For the quantum cellular automaton, the measure of parallel efficiency that corresponds to these classical measures is

$$\lim_{M\to\infty} \gamma_{\max}(M)/M, \tag{5.26}$$

where $\gamma_{\max}(M)$ is the largest positive eigenvalue of $\widehat{\Gamma}$ for a system of size M. The spatial analog of diffusive computation corresponds to

$$\gamma_{\max}(M) \sim M^\beta \tag{5.27}$$

with $\beta \leq 1/2$.

From a computational point of view, of course, the most desirable scaling exponent is $\beta = 1$, which defines uniform-rate computation. In that case we will find

$$\gamma_{\max}(M) = \eta M \tag{5.28}$$

for some coefficient η, equal to the parallel efficiency, $0 < \eta \leq 1$. In the case $\eta = 1$, we say that *ideal scalability* is achieved.

Until now, no model of quantum computation was solvable, so it was not possible to address any of these issues. The exact solution given below will enable us to compute the eigenvalues and eigenstates of $\widehat{\Gamma}(M)$ and the scaling of $\gamma_{\max}(M)$ with system size M.

5.2 Exact Solution

Our goal then is to find the exact eigenstates and eigenvalues of $\widehat{\Gamma}(M)$ in systems of arbitrary size M. In particular, we are interested in the limits of parallelism, so we will want to find the largest eigenvalue $\gamma_{\max}(M)$ of $\widehat{\Gamma}(M)$. We want to know what state to prepare the system in to obtain the fastest possible computation-rate, so we will want to know the eigenvector $|\gamma_{\max}(M)\rangle$. that corresponds to the largest eigenvalue. Finally, we are interested in how much all this parallel computation will cost us in terms of energy—we want to understand the correspondence between the energy eigenvalues of the one-dimensional XY Hamiltonian (5.19) and the parallel computation-rate eigenvalues.

The exact solution we obtain in this section will let us answer these questions, but the path to that solution is a little more tedious than one might expect from the simple form of the operators (5.22) and (5.19). Fortunately, the physical reasons for this are easy to see[117].For the computation-rate operator (5.22), clearly the state $|0\rangle = \prod_{n=0}^{M} |-\rangle_n$ is an eigenstate with eigenvalue $\gamma = 0$. Since the system is translation invariant, single-particle states with translation eigenvalue k (such that $e^{ikMa} = 1$) may be constructed as

$$|k\rangle = \sum_{n=0}^{M} e^{ikna} \sigma_n^+ |-\rangle. \tag{5.29}$$

If we try to continue by constructing two-particle states

$$|k, k'\rangle = |k\rangle \; |k'\rangle, \tag{5.30}$$

we realize that the product contains terms like $\sigma_n^+|-\rangle \times \sigma_n^+|-\rangle$, where both particles (flipped spins) reside on the same site, in violation of the exclusion principle. Ordinarily, for indistinguishable fermions we would be able to apply an antisymmetrizer $\mathcal{A}$ to $|k,k'\rangle$ to obtain the correct statistics. But in our case, this goes too far; the lattice index n makes the operators on different sites effectively distinguishable. The antisymmetrizer solves the problem of multiple occupancy of an individual site (which we want) but it also antisymmetrizes between sites (which we don't want).

One solution is to apply the antisymmetrizer, and then undo the antisymmetry between sites "by hand". This can be done by taking appropriate boundary conditions in place of the periodic condition $e^{ikMa}=1$. If we antisymmetrize, we obtain

$$\sum_{m=0}^{M}\sum_{n=0}^{M}(e^{ikma+ik'na}-e^{ikna+ik'ma})\sigma_m^+\sigma_n^+|0\rangle \tag{5.31}$$

or, equivalently

$$\sum_{m=0}^{M}\sum_{m<n<N+m}^{M}(e^{ikma+ik'na}-e^{ikna+ik'ma})\sigma_m^+\sigma_n^+|0\rangle, \tag{5.32}$$

where we take $\sigma_{N+j}^+\equiv\sigma_j^+$. To undo the unwanted part of the antisymmetrization, note that for two sites (m,n), where $m<n<M$, the transposition of sites occurs in this sum as $(n+M,m)$ rather than as (n,m). So the transpose contains the factor

$$e^{ikma+ik'na}e^{ikMa}-e^{ikMa}e^{ikna+ik'ma}. \tag{5.33}$$

From (5.33) we see that the unwanted anticommutation between sites can be undone by taking the antiperiodic boundary condition

$$e^{ikMa}=-1 \tag{5.34}$$

in place of the periodic boundary condition.

In general, for the p-particle states, when the particle with the largest lattice index passes through the origin to become the particle with the smallest index, it is equivalent to transposing it with the $p-1$ intervening particles. Since the antisymmetrizer treated these particles as fermions an unwanted sign $(-1)^{p-1}$ was acquired.

To remove it, we need to take the antiperiodic boundary condition (5.34) when p is even and the periodic boundary condition when p is odd.

In principle, we could perform this hand symmetrization throughout, but in practice it becomes too cumbersome for computing the spectra of the operators we are interested in. We need a more systematic way to handle the mixture of commutation and anticommutation relations that occur in our system. This is essentially what the Jordan-Wigner fermion operators[77, 79, 91] accomplish. We will be engaged in manipulating these operators throughout §5.2.1–5.2.4. As a result, we will obtain the exact eigenstates and eigenvalues displayed in Table 5.1 at the end of §5.2.4.

5.2.1 Fermion Interpretation

Introducing Jordan-Wigner operators

$$\begin{aligned} c_n &\equiv \left(\prod_{i=0}^{n-1} e^{i\pi\sigma_i^+\sigma_i^-}\right)\sigma_n^- = \left(\prod_{i=0}^{n-1} -\sigma_i^z\right)\sigma_n^-, \\ c_n^\dagger &\equiv \sigma_n^+\prod_{i=0}^{n-1} e^{i\pi\sigma_i^+\sigma_i^-} = \sigma_n^+\prod_{i=0}^{n-1} -\sigma_i^z, \end{aligned} \tag{5.35}$$

we see that the string of σ^z operators that "tie" each $\sigma^\pm$ to the origin have the effect (Fig. 5-9) of generating the required off-site anticommutation relation, since the anticommutation of any two c_i, c_j with $i < j$ will always entail the anticommutator $\{\sigma_i^-, \sigma_i^z\} = 0$, so that $\{c_i, c_j\} = 0$. Similarly, one can verify that $\{c_i, c_j^\dagger\} = \delta_{mn}$.

Using the relations

$$\begin{aligned} \sigma^\pm\sigma^z &= \mp\sigma^\pm, \\ \sigma^z\sigma^\pm &= \pm\sigma^\pm, \end{aligned} \tag{5.36}$$

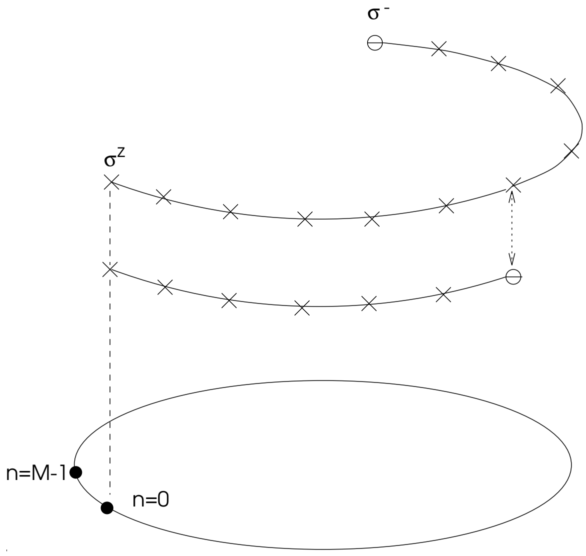

Figure 5-9: Schematic depiction of the Jordan-Wigner operators. Each $\times$ represents a factor of σ^z and each $\ominus$ a factor of σ^-. The anticommutation of the operators indicated by the dotted arrows gives the required off-site anticommutation relation for fermions.

we find that for the non-boundary cases, $n = 0, \ldots M-2$,

$$
\begin{aligned}
\sigma_n^+ \sigma_{n+1}^+ &= -c_n^\dagger c_{n+1}^\dagger, \\
\sigma_n^- \sigma_{n+1}^- &= c_n c_{n+1}, \\
\sigma_n^+ \sigma_{n+1}^- &= -c_n^\dagger c_{n+1}, \\
\sigma_n^- \sigma_{n+1}^+ &= c_n c_{n+1}^\dagger.
\end{aligned} \tag{5.37}
$$

5.2.2 Periodic Boundary Conditions

As described above, unexpected subtleties arise from the conjunction of fermion exchange symmetry and periodic boundary conditions.

If we were trying to understand the thermodynamic properties of this system then, in the thermodynamic limit, the particular boundary conditions chosen would typically affect the result by a term of order $\sim 1/M$, where M is the system size. For such purposes a convenient and harmless inconsistency can then be indulged in: expansions are done in a basis of eigenstates of the periodic boundary conditions, while the tedious operator relations entailed by periodic boundary conditions are avoided by (inconsistently) employing the simpler relations appropriate to a chain with free ends. Unfortunately, the computation-rate operator and Hamiltonian cannot be diagonalized for the open-ended line; they can be only be simultaneously diagonalized if periodic boundary conditions are imposed. This unexpected distinction between situations that are usually physically indistinguishable delayed our observation that we could have $[\widehat{H},\widehat{\Gamma}]=0$.

The case $n=M-1$ must therefore be handled with some care. We can derive the following operator identities in the case of periodic boundary conditions

$$\begin{aligned} c^{\dagger}_{M-1}c_0 &= \sigma^{+}_{M-1}(\prod_{j=0}^{M-2}-\sigma^z_j)\sigma^{-}_0 \\ &= \sigma^{+}_{M-1}(\prod_{j=0}^{M-1}-\sigma^z_j)\sigma^{-}_0 . \end{aligned} \tag{5.38}$$

Since $\sigma^z_0\sigma^-_0=-\sigma^-_0=-\sigma^-_0\sigma^z_0$, it follows that

$$c^{\dagger}_{M-1}c_0=-\sigma^{+}_{M-1}\sigma^{-}_0\prod_{j=0}^{M-1}-\sigma^z_j, \tag{5.39}$$

so that the product of operators straddling the origin is given by

$$\sigma^{+}_{M-1}\sigma^{-}_0=-c^{\dagger}_{M-1}c_0 e^{i\pi\sum_{j=0}^{M-1}c^{\dagger}_jc_j}=-c^{\dagger}_{M-1}c_0(-1)^{\widehat{N}_F} \tag{5.40}$$

where $\widehat{N}_F$ is the fermion number operator. And similarly,

$$\sigma^{-}_{M-1}\sigma^{+}_0=-e^{i\pi\sum_{j=0}^{M-1}c^{\dagger}_jc_j}c^{\dagger}_0c_{M-1}=-(-1)^{\widehat{N}_F}c^{\dagger}_0c_{M-1}. \tag{5.41}$$

Armed with these identities, we can now express the computation-rate and Hamiltonian operators in terms of the true fermion operators:

$$\widehat{\Gamma} = \frac{1}{i} \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n + (-1)^{\widehat{N}_F} c_0^\dagger c_{M-1} - (-1)^{\widehat{N}_F} c_{M-1}^\dagger c_0 \tag{5.42}$$

and

$$\widehat{H} = \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} + c_{n+1}^\dagger c_n - (-1)^{\widehat{N}_F} c_0^\dagger c_{M-1} - (-1)^{\widehat{N}_F} c_{M-1}^\dagger c_0 \,. \tag{5.43}$$

Note that the number of fermions present is manifestly conserved by the Hamiltonian. For initial states capable of synchronizing the quantum cellular automaton properly, the number of fermions is equal to $M/2$.

5.2.3 Parity of Fermion Occupation

In the fermion basis, we can diagonalize the computation-rate operator

$$\widehat{\Gamma} = \frac{1}{i} \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n + (-1)^{\widehat{N}_F} c_0^\dagger c_{M-1} - (-1)^{\widehat{N}_F} c_{M-1}^\dagger c_0 \,. \tag{5.44}$$

The appearance of the number operator complicates the diagonalization of the computation-rate operator $\widehat{\Gamma}$. It also occurs in the Hamiltonian,

$$\widehat{H} = \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} + c_{n+1}^\dagger c_n - (-1)^{\widehat{N}_F} c_0^\dagger c_{M-1} - (-1)^{\widehat{N}_F} c_{M-1}^\dagger c_0 \,. \tag{5.45}$$

First, define projection operators[82, 83]

$$P_\pm = \frac{1}{2}(1 \pm (-1)^{\widehat{N}_F}) \tag{5.46}$$

onto the subspaces of even and odd N_F. Then, since

$$P_\pm (-1)^{\widehat{N}_F} = \pm P_\pm, \tag{5.47}$$

we can split the computation-rate operator

$$\widehat{\Gamma} = P_+ \widehat{\Gamma} + P_- \widehat{\Gamma} = P_+ \widehat{\Gamma}_+ + P_- \widehat{\Gamma}_- \tag{5.48}$$

into even- and odd-fermion parts $\widehat{\Gamma}_\pm$, where the even-fermion part is

$$\widehat{\Gamma}_+ = \frac{1}{i}\sum_{n=0}^{M-2} c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n + c_0^\dagger c_{M-1} - c_{M-1}^\dagger c_0 \tag{5.49}$$

and the odd part is

$$\widehat{\Gamma}_- = \frac{1}{i}\sum_{n=0}^{M-1} c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n \ . \tag{5.50}$$

Similarly, the Hamiltonian can be written as

$$\widehat{H} = P_+\widehat{H} + P_-\widehat{H} = P_+\widehat{H}_+ + P_-\widehat{H}_- \ , \tag{5.51}$$

where the even-fermion part is

$$\widehat{H}_+ = \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} + c_{n+1}^\dagger c_n - c_0^\dagger c_{M-1} - c_{M-1}^\dagger c_0 \tag{5.52}$$

and the odd part

$$\widehat{H}_- = \sum_{n=0}^{M-1} c_n^\dagger c_{n+1} + c_{n+1}^\dagger c_n . \tag{5.53}$$

The operators (5.49)–(5.53) are now free of reference to $\widehat{N}_F$ and can be diagonalized independently. The correct spectra of $\widehat{\Gamma}$ and $\widehat{H}$ must then be reconstructed by selecting as eigenstates and eigenvalues of $\widehat{\Gamma}$ (and $\widehat{H}$) only those eigenstates and eigenvectors of $\widehat{\Gamma}_\pm$ (resp. $\widehat{H}_\pm$) that survive the corresponding projections in (5.48) and (5.51).

5.2.4 Eigenstates and Eigenvalues

Our Hamiltonian is now seen to involve only the free propagation of fermions, and furthermore, the number of fermions is manifestly conserved. For the initial condition we contemplate when using this system to synchronize a quantum computation, every other site contains a fermion. The evolution of the synchronization bits then corresponds to the free propagation, with correct exchange symmetry, of these fermions along a one-dimensional ring.

Letting

$$a_k = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} e^{ikn} c_n,$$

$$a_k^\dagger = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} e^{-ikn} c_n^\dagger, \tag{5.54}$$

we find

$$\sum_n c_n c_{n+1}^\dagger = \sum_n \frac{1}{M} \sum_k e^{-ikn} a_k \sum_{k'} e^{i(k'+1)n} a_k'^\dagger \; . \tag{5.55}$$

Consider the even-fermion part of the computation-rate operator $\widehat{\Gamma}_+$,

$$\widehat{\Gamma}_+ = \frac{1}{i} \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n + c_0^\dagger c_{M-1} - c_{M-1}^\dagger c_0 \; . \tag{5.56}$$

If we define the sets $K_{\text{odd}} = \{\frac{\pi}{Ma}(2q+1)|q = 0, \pm 1, ...\}$ and $K_{\text{even}} = \{\frac{\pi}{Ma} 2q | q = 0, \pm 1, ...\}$, then

$$\begin{aligned} \widehat{\Gamma}_+ \;=\; & \tfrac{1}{i} \sum_{n=0}^{M-2} \left(\tfrac{1}{M} \textstyle\sum_{k \in K_{\text{odd}}} \tfrac{1}{M} \sum_{k' \in K_{\text{odd}}} \right. \\ & \quad \left. e^{ikna} a_k^\dagger e^{-ik'(n+1)a} a_{k'} - e^{ik(n+1)a} a_k^\dagger e^{-ik'na} a_{k'} \right) \\ & + \tfrac{1}{M} \textstyle\sum_{k \in K_{\text{odd}}} \tfrac{1}{M} \sum_{k' \in K_{\text{odd}}} \\ & \quad - e^{ik(M-1)a} a_k^\dagger a_{k'} + a_k^\dagger e^{ik'(M-1)a} a_{k'} \; . \end{aligned} \tag{5.57}$$

Since for even fermion occupation, $e^{ikMa} = e^{ik'Ma} = -1$, the two boundary terms can be combined with the sum from 0 to $M-2$,

$$\widehat{\Gamma}_+ = \sum_{k \in K_{\text{odd}}} -2 \sin ka \; a_k^\dagger a_k \; . \tag{5.58}$$

For the odd-fermion part $\widehat{\Gamma}_-$ of the computation-rate operator,

$$\widehat{\Gamma}_- = \frac{1}{i} \sum_{n=0}^{M-1} c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n \; . \tag{5.59}$$

Note that the limit in the sum is now $M-1$, since the boundary terms are merged into the sum in the odd-fermion case,

$$\widehat{\Gamma}_- = \tfrac{1}{i}\sum_{n=0}^{M-1} \Big(\tfrac{1}{M}\textstyle\sum_{k\in K_{\rm even}} \tfrac{1}{M}\sum_{k'\in K_{\rm even}} \\ e^{ikna} a_k^\dagger e^{-ik'(n+1)a} a_{k'} - e^{ik(n+1)a} a_k^\dagger e^{-ik'na} a_{k'} \,.\Big) \tag{5.60}$$

Simplifying,

$$\widehat{\Gamma}_- = \sum_{k\in K_{\rm even}} -2\sin ka\ a_k^\dagger a_k\ . \tag{5.61}$$

The even and odd-fermion parts of the Hamiltonian are handled in the same way:

$$\widehat{H}_+ = \sum_{n=0}^{M-2} c_n^\dagger c_{n+1} + c_{n+1}^\dagger c_n - c_0^\dagger c_{M-1} - c_{M-1}^\dagger c_0. \tag{5.62}$$

Taking the Fourier transform,

$$\begin{aligned} \widehat{H}_+ = \sum_{n=0}^{M-2} \Big(&\tfrac{1}{M}\textstyle\sum_{k\in K_{\rm odd}} \tfrac{1}{M}\sum_{k'\in K_{\rm odd}} \\ &e^{ikna} a_k^\dagger e^{-ik'(n+1)a} a_{k'} + e^{ik(n+1)a} a_k^\dagger e^{-ik'na} a_{k'}\Big) \\ &-\tfrac{1}{M}\textstyle\sum_{k\in K_{\rm odd}} \tfrac{1}{M}\sum_{k'\in K_{\rm odd}} \\ &-e^{ik(M-1)a} a_k^\dagger a_{k'} + a_k^\dagger e^{ik'(M-1)a} a_{k'}\ . \end{aligned} \tag{5.63}$$

Again, for even fermion occupation, $e^{ikMa} = e^{ik'Ma} = -1$, and the two boundary terms can be combined with the sum from 0 to $M-2$, yielding

$$\widehat{H}_+ = \sum_{k\in K_{\rm odd}} 2\cos ka\ a_k^\dagger a_k\ . \tag{5.64}$$

For the odd-fermion part $\widehat{H}_-$,

$$\widehat{H}_- = \sum_{n=0}^{M-1} c_n^\dagger c_{n+1} + c_{n+1}^\dagger c_n\ , \tag{5.65}$$

$$\widehat{H}_- = \sum_{n=0}^{M-1} \Big(\tfrac{1}{M}\textstyle\sum_{k\in K_{\rm even}} \tfrac{1}{M}\sum_{k'\in K_{\rm even}} \\ e^{ikna} a_k^\dagger e^{-ik'(n+1)a} a_{k'} + e^{ik(n+1)a} a_k^\dagger e^{-ik'na} a_{k'} \,.\Big) \tag{5.66}$$

Operator	Allowed Wavenumbers	Eigenvalues	Eigenvectors
$\widehat{\Gamma}_+$	$\{\ (2q+1)\frac{\pi}{Ma}\ \vert\ q=0,\pm1,...\}$	$2\sin(2q+1)\frac{\pi}{M}$	$a^\dagger_{2q+1}\vert 0\rangle$
$\widehat{\Gamma}_-$	$\{\ 2q\frac{\pi}{Ma}\ \vert\ q=0,\pm1,...\}$	$2\sin 2q\frac{\pi}{M}$	$a^\dagger_{2q}\vert 0\rangle$
$\widehat{H}_+$	$\{\ (2q+1)\frac{\pi}{Ma}\ \vert\ q=0,\pm1,...\}$	$2\cos(2q+1)\frac{\pi}{M}$	$a^\dagger_{2q+1}\vert 0\rangle$
$\widehat{H}_-$	$\{\ 2q\frac{\pi}{Ma}\ \vert\ q=0,\pm1,...\}$	$2\cos 2q\frac{\pi}{M}$	$a^\dagger_{2q}\vert 0\rangle$

Table 5.1: Eigenvectors and eigenvalues.

The odd-fermion case proceeds in complete analogy to that for the computation-rate operator, $\widehat{\Gamma}_-$,

$$\widehat{H}_- = \sum_{k\in\{\frac{\pi}{Ma}2q|q=0,\pm1,...\}} 2\cos ka\ a^\dagger_k a_k\ . \tag{5.67}$$

So we see that the eigenvalues for both $\widehat{\Gamma}_\pm$ are sums over occupied momentum states, $\sum_{k \text{ occupied}} 2\sin ka$, where $k \in \{\frac{\pi}{Ma}(2q+1)|q = 0,\pm1,...\}$ for $\widehat{\Gamma}_+$ and $k \in \{\frac{\pi}{Ma}2q|q=0,\pm1,...\}$ for $\widehat{\Gamma}_-$. In each case, the corresponding eigenvector is given by

$$|\psi\rangle = \prod_{\{k\}\ \text{occupied}} a^\dagger_k|0\rangle, \tag{5.68}$$

where $|0\rangle$ is the momentum space vacuum, and the k's are either odd or even multiples of π/Ma for the even or odd fermion occupancy cases, respectively. Similarly for the even and odd fermion parts of the Hamiltonian, $\widehat{H}_\pm$, except that $2\cos k$ replaces $2\sin k$ throughout.

5.3 Parallel Computation Rate

From the single-particle eigenstates and eigenvalues found in the previous section, we can construct eigenstates of $\widehat{\Gamma}(M)$ and determine both the maximum rate of parallel computation and the energetic cost of computing at a specified rate.

However, we must be careful about what we call the maximum computational rate; it need not correspond to the largest eigenvalue of $\widehat{\Gamma}(M)$. Recall that the quantum state corresponding to the initial state of the classical computation has alternate spins up and down. Since the number of flipped spins, which is also the number of fermions, is manifestly conserved by the Hamiltonian, all configurations will satisfy

$\widehat{N}_F = M/2$. Therefore, a configuration of the synchronization bits is of no interest, regardless of how big γ_c in that state, if it is not also contained in the eigenspace of $\widehat{N}_F$ with eigenvalue $M/2$. We will refer this as the *physical subspace.*

The maximum computational velocity is therefore properly understood to mean the largest eigenvalue $\gamma_{\max}(M)$ of $\widehat{\Gamma}(M)$ such that the corresponding eigenspace $L_{\gamma_{\max}}$ has a nontrivial intersection with the physical subspace. We will need to keep this constraint in mind in the following section where we construct exact eigenvectors of $\widehat{\Gamma}$ and in Section 5.3.2 where we use it to eliminate spurious numerical solutions.

5.3.1 Maximum Computation Rate

First, we briefly review the construction of the ground state[117]. For $M \to \infty$, the spectra of both $\widehat{H}$ and $\widehat{\Gamma}$ in Table 5.1 can be represented as a continuous plot with respect to k as in Fig. 5-10. In actuality, only the cases where M is even produce a 2-body cellular automaton with well-defined neighborhoods. In addition, the figure must be used with the understanding that for finite M, if $M = 0 \pmod 4$ then $k = (2q+1)\pi/(Ma)$, while for $M \neq 0 \pmod 4$, $k = (2q)\pi/(Ma)$. At $T = 0$, the ground state is obtained by filling the $M/2$ lowest energy single-particle states. This procedure creates a Fermi sea in which all single-particle states with $k \in (-\pi, -\pi/2)$ or $k \in (\pi/2, \pi)$ are occupied. The total energy in the ground state is then

$$E_{\text{g.s.}}(M) = \int_{-\pi}^{-\pi/2} 2\cos ka\ d(Mak/2\pi) + \int_{-\pi}^{-\pi/2} 2\cos ka\ d(Mak/2\pi) = -2M/\pi\ . \tag{5.69}$$

The maximum computation-rate $\gamma_{\max}(M)$ can be found in a similar manner. Because we always divide by the number of sites M, rather than the number of fermions, in order to determine the parallel efficiency, we cannot maximize the parallel efficiency $\gamma(M)/M = \sum \gamma_k$ by including only a few single-particle states with very large eigenvalues γ_k. Instead, starting from the largest positive single-particle eigenvalue $\gamma_{\pi/2}$, we should include progressively smaller γ_k until we have either included $M/2$ single-particle states or we have exhausted all positive eigenvalues.

However, because the spectrum of $\widehat{\Gamma}$ is symmetric about the k-axis, exactly half of

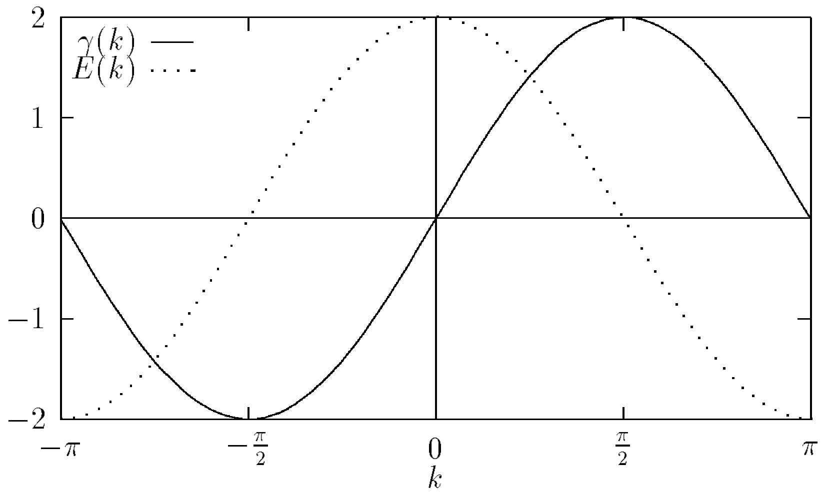

Figure 5-10: Single-particle spectra of $\widehat{H}(M)$ and $\widehat{\Gamma}(M)$ in the limit $M \to \infty$.

the single particle states have $\gamma_k > 0$. Therefore, the physical case of half-filling (i.e., with $M/2$ single-particle states) coincides with the case that maximizes the parallel utilization Γ/M. We see that the largest (and smallest) eigenvalues of $\widehat{\Gamma}$ always belong to the physical subspace.

Because a translation/reflection symmetry connects the energy and computation-rate spectra, there is a simple relation between the ground-state energy and the maximum computation rate. If we reflect the spectrum of $\widehat{H}$ through the k-axis and translate it by $-\pi/2$, the states of lowest energy are brought into congruence with the states that perform parallel computation at the highest rate. Because of this symmetry, we see that the maximum parallel computation rate is identically equal to the negative of the ground-state energy:

$$\gamma_{\max}(M) = \lim_{M\to\infty} -E^{\rm AF}_{\rm g.s}(M) = \frac{2}{\pi}M \ . \tag{5.70}$$

Therefore, the asymptotic efficiency of parallel computation is predicted to be

$$\lim_{M\to\infty} \frac{\gamma_{\max}(M)}{M} = \frac{2}{\pi}. \tag{5.71}$$

The convergence to this asymptotic limit can be examined for small M either by constructing the exact solution explicitly or by obtaining the eigenvalues of $\widehat{\Gamma}(M)$ numerically.

5.3.2 Numerical Results

The Hilbert space $\mathcal{H}(M)$ for the M site space is $\mathcal{H}(M) = \bigotimes_{n=0}^{M-1} \mathcal{H}_n$. In the basis consisting of the 2^M tensor products $\{\bigotimes_{n=0}^{M-1} |m_z\rangle_n \;\; | \; m_z = \pm 1\}$, ordered from $\bigotimes_{n=0}^{M-1} |-\rangle_n$ to $\bigotimes_{n=0}^{M-1} |+\rangle_n$, the $\widehat{\Gamma}(M)$ are sparse Hermitian matrices which have a self-similar structure along the diagonal. For $M = 4$ and $M = 6$, they are shown in Figs. 5-11 and 5-12.

$$
\begin{array}{cccccccccccccc}
 & - & & & & & & + & & & & & & \\
+ & & & - & & & & & & & & & & \\
 & & & & - & & & & & + & & & & \\
 & + & & & & & & - & & & & & & \\
 & & + & & & - & & & - & & & + & & \\
 & & & & + & & & & & - & & & & \\
 & & & & & & & & & & - & & & + \\
- & & & + & & & & & & & & & & \\
 & & & & + & & & & & - & & & & \\
 & & - & & & + & & & + & & & - & & \\
 & & & & & & + & & & & & & - & \\
 & & & & - & & & & & + & & & & \\
 & & & & & & & & & & + & & & - \\
 & & & & & & - & & & & & & + &
\end{array}
$$

Figure 5-11: Form of $\widehat{\Gamma}(4)$. Nonzero elements are all $\pm i$.

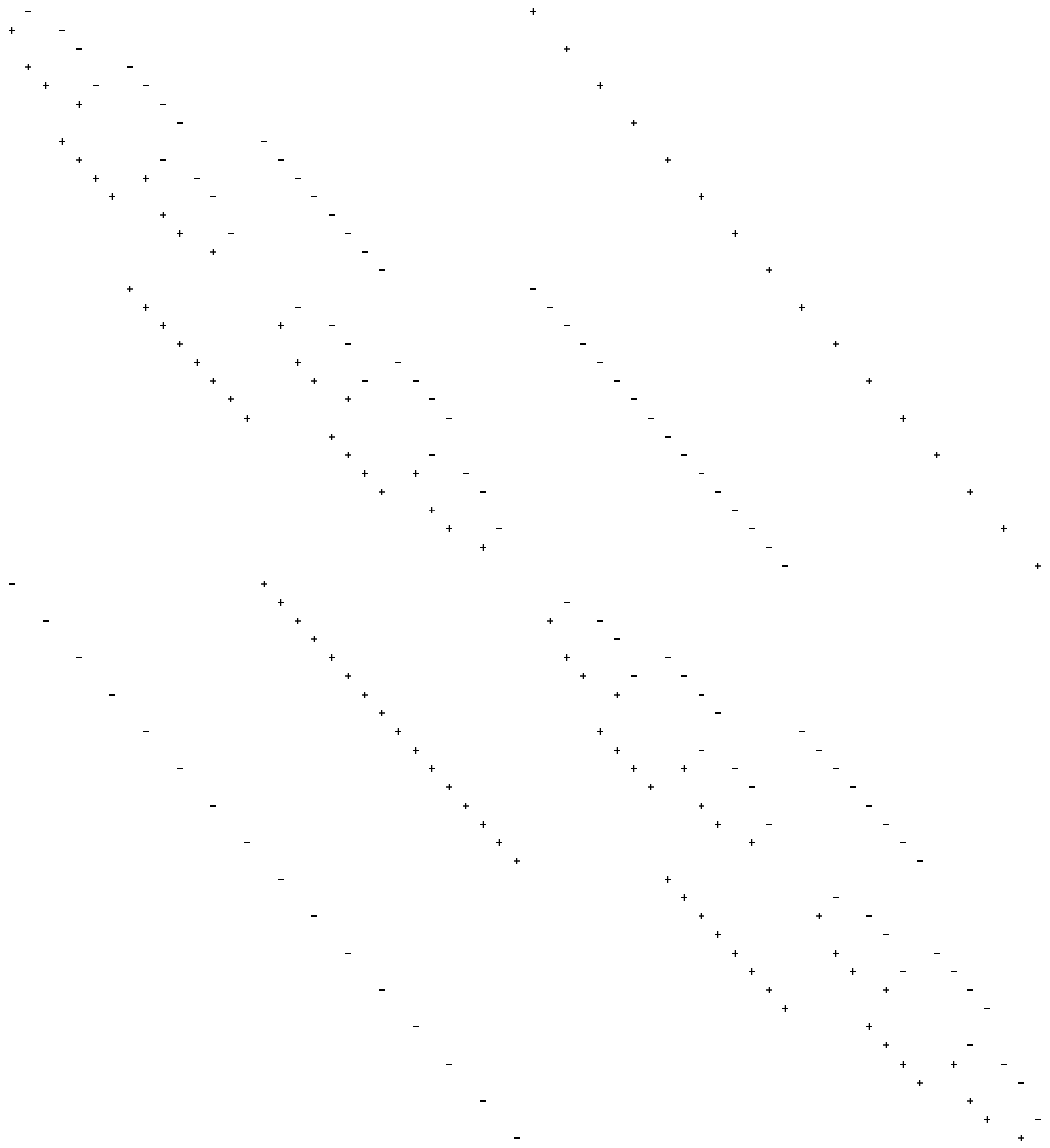

Figure 5-12: Form of $\widehat{\Gamma}(6)$.

The eigenvalues and eigenvectors were found by iterating the Jacobi transformation[133]. Figures (5-13)–(5-15) show the spectral distribution obtained by diagonalizing $\widehat{\Gamma}(M)$ for $M = 4, 6$, and 8 numerically.

In each case, the single-particle spectrum is concentrated around $\gamma = 0$, with the greatest spectral weight at exactly $\gamma = 0$. Consequently, it is far from obvious that one can construct any $M/2$-particle state whose computation rate satisfies the spatial

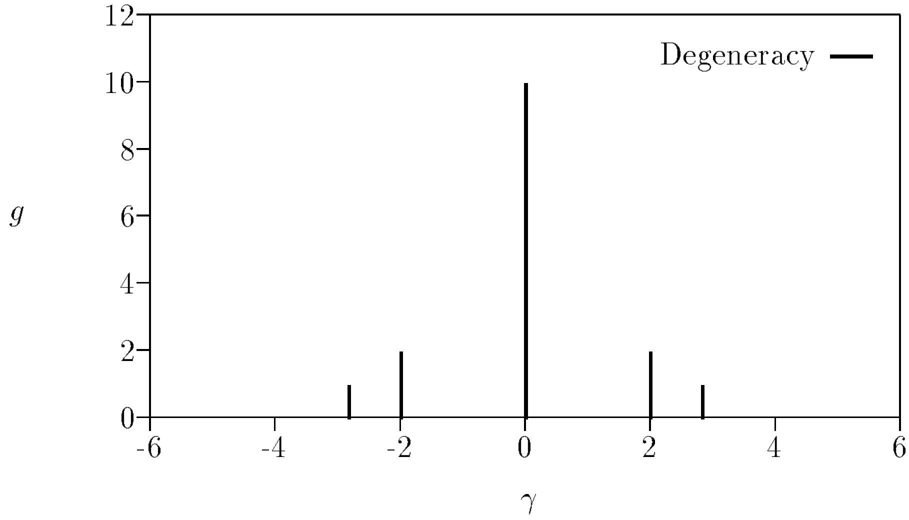

Figure 5-13: Spectrum of $\widehat{\Gamma}(M)$ for $M = 4$. Degeneracy g versus eigenvalue γ.

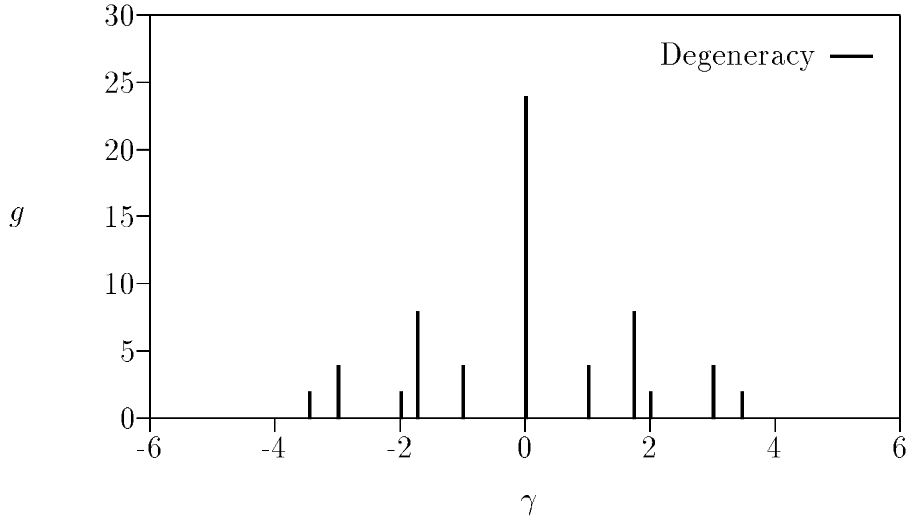

Figure 5-14: Spectrum of $\widehat{\Gamma}(M)$ for $M = 6$.

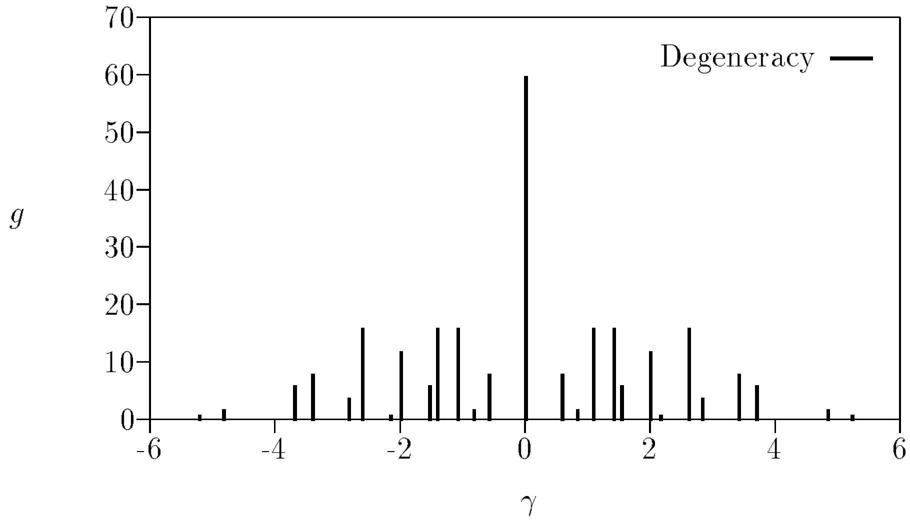

Figure 5-15: Spectrum of $\widehat{\Gamma}(M)$ for $M = 8$.

scaling relation $\gamma_{\max}(M) > M^{\beta}$ with an exponent β larger than the diffusion-like $\beta = 1/2$.

Numerical and exact analytic solutions will show that in fact $\gamma_{\max}(M) \sim M^{\beta}$ with $\beta = 1$. That is, the quantum automaton attains linear spatial scalability. Consider the construction of the maximum computation-rate eigenvalue for the simplest nontrivial case, $M = 4$. From Table 5.1, the allowed values of ka are $\pm\pi/4$ and $\pm 3\pi/4$. The corresponding single-particle eigenvalues are $\gamma_k = 2\sin(ka)$. That is $\gamma_{\pm\pi/4} = \pm\sqrt{2}$ and $\gamma_{\pm 3\pi/4} = \pm\sqrt{2}$. We need to choose the two largest, so we obtain

$$\gamma_{\max}(4) = 2\sqrt{2}. \tag{5.72}$$

Numerically we find $\gamma_{\max}(4) = 2.82843$.

The eigenstate corresponding to $\gamma_{\max}(4) = 2\sqrt{2}$ can be obtained from Table 5.1 by calculating the coefficients of the product basis when the two single-particle states

$ka = \pm\pi/4$ and $\pm 3\pi/4$ are included. We find

$$\begin{aligned} |2\sqrt{2}\rangle &= -\tfrac{i}{4}\sqrt{2}|--++\rangle + \tfrac{1}{2}|-+-+\rangle + \tfrac{i}{4}\sqrt{2}|+--+\rangle \\ &\quad +\tfrac{i}{4}\sqrt{2}|-++-\rangle - \tfrac{1}{2}|+-+-\rangle - \tfrac{i}{4}\sqrt{2}|++--\rangle \,. \end{aligned} \tag{5.73}$$

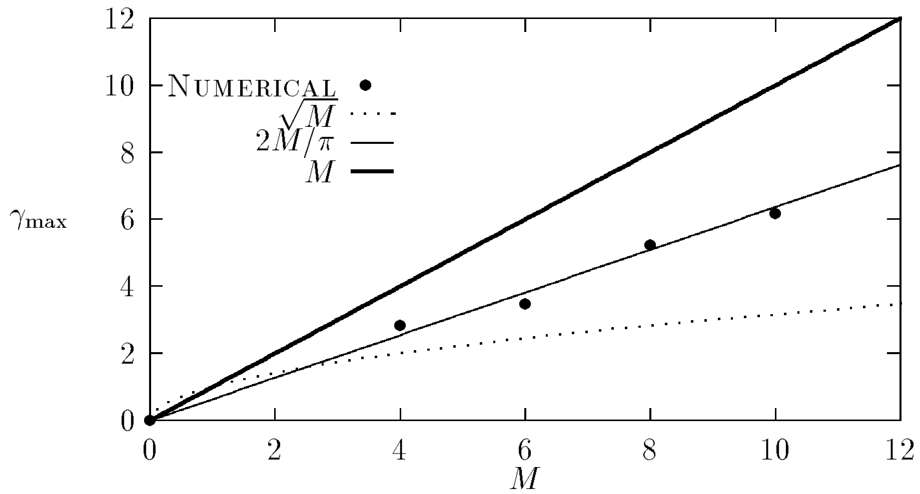

Figure 5-16: Comparison of numerical results for maximum parallel computation velocity and exact value of asymptotic $M \to \infty$ behavior of $-E^{\mathrm{AF}}_{\mathrm{g.s}}$. Linear and square root scaling is shown for comparison.

Both numerical and analytic solutions for $M = 4, 6$, and 8 indicate that the asymptotic efficiency of parallel computation $2M/\pi$ is attained even for the smallest possible values of M (see Fig. 5-16). From the spectra of $\widehat{\Gamma}$ and $\widehat{H}$ (Fig. 5-10), we see that, in order to attain the maximum computation rate, an excitation energy $\Delta E = 2M/\pi$ must be supplied. Since this equals half the maximum possible energy of the system, this is a relatively large excitation. Now we would like to explore the question of how fast the parallel quantum cellular automaton can compute with smaller excitation energies.

5.3.3 Relation Between Energy and Computation Rate

When the system is in its ground state, the computation rate γ vanishes because the energy spectrum is symmetric and the computation-rate spectrum antisymmetric about $k = 0$. In fact, we can see from Fig. 5-17 that the computation rate γ vanishes whenever the system is in thermal equilibrium at any finite T. This follows from the fact that the computation-rate spectrum (thin solid line) is symmetric about the Fermi surface at $k = \pm\pi/2$, so that the finite-temperature Fermi distribution function (bold line) cannot change $\gamma(M)$ from its $T = 0$ value, which is $\gamma(M) = 0$.

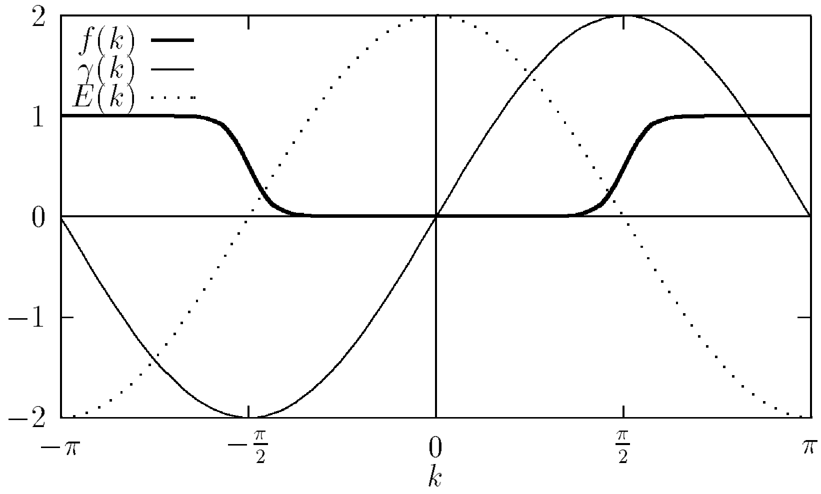

Figure 5-17: Temperature dependence of the computation rate. The Fermi distribution function $f(k)$ (solid bold line) at the finite temperature $\beta = 10$ illustrates why $\langle\gamma\rangle = 0$ at all finite temperatures.

Although it is not possible to compute faster than diffusively when the system is in thermal equilibrium[17], the quantum cellular automaton can attain a substantial fraction of its maximum possible computation rate with surprisingly little excitation energy.

To show this, we must first identify the state with the largest computation rate among those of a given energy. Alternatively, we can achieve the same end by identifying the state with minimum energy among those with a given computation rate.

States of many different total energies can have the same parallel computation-rate. For a given $\gamma_0 \leq \gamma_{\max} = 2M/\pi$, we would like to find the minimum energy at which γ_0 can be achieved. From the spectra of $\widehat{H}$ and $\widehat{\Gamma}$ (Fig. 5-10), we see that to obtain a given γ_0 at a minimum energy cost we should shift the ground state configuration of the Fermi sea to the left until γ has increased from its ground state value of zero to γ_0. This follows from the fact that the spectra of $\widehat{H}$ and $\widehat{\Gamma}$ are shifted by $\pi/2$ along the k-axis, and therefore the left shift adds the k states with the largest available eigenvalues of $\widehat{\Gamma}$ and smallest positive energy (near $k = \pi/2$), while simultaneously vacating the states of least negative energy and most negative computation-rate (near $k = -\pi/2$).

For example, assume $M \pmod 4 \neq 0$, so that $e^{ikMa} = 1$, and therefore $k_n a = 2\pi n/M$. From Table 5.1, the single-particle eigenvalues in this case are

$$\begin{aligned} &2\sin 2\tfrac{\pi}{M}n \quad (\text{for } \widehat{\Gamma}), \\ &2\cos 2\tfrac{\pi}{M}n. \quad (\text{for } \widehat{H}) . \end{aligned} \tag{5.74}$$

When M is large, the full $M/2$-particle eigenvalues of $\widehat{\Gamma}$ and $\widehat{H}$ can be approximated by integrals over occupied single-particle states. If $\delta ka = \epsilon$ denotes the amount by which the Fermi sea has shifted toward the left, then

$$\gamma(\epsilon) = 2\int_0^{\epsilon} 2\cos ka \frac{M}{2\pi} d(ka) = \frac{2M}{\pi}\sin\epsilon \tag{5.75}$$

and

$$E(\epsilon) = 2\int_0^{\epsilon} 2\sin ka \frac{M}{2\pi} d(ka) = \frac{4M}{\pi}\sin^2\frac{\epsilon}{2}. \tag{5.76}$$

From these, we can express the maximum computation rate obtainable at energy E,

$$\gamma(E) = \sqrt{E(\frac{4M}{\pi} - E)}. \tag{5.77}$$

Figure 5-18 shows the maximum computation rate per site that can be attained for a given energy density (i.e., energy per site). The parallel computation rate is normalized to the maximum attainable at any energy. The maximum parallel computation rate $\gamma_{\max}(M)$ occurs when the energy density is equal to half the maximum energy density the system can possess. Note that near the ground-state, a small excitation energy can produce a relatively large increase in the computation rate. An excitation energy $\Delta E = 0.1 \times E_{\max}$ one can yield a parallel computation rate $\gamma = 0.6 \times \gamma_{\max}$. In contrast, as the maximum computation rate is approached, relatively large increments to the energy of the system are unable to produce much additional computation rate. Beyond an excitation energy equal to half the maximum energy content of the system, additional excitation energy actually reduces the computation rate.

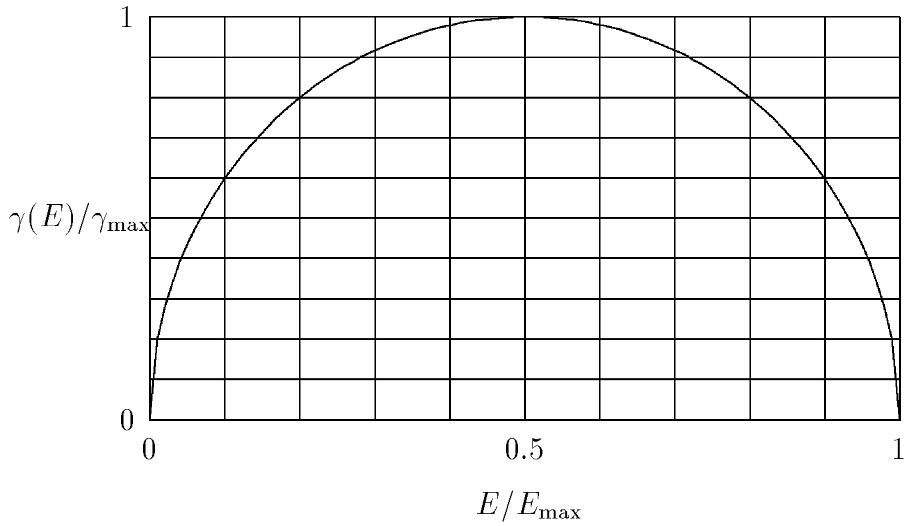

Figure 5-18: Maximum computation rate per site as a function of excitation energy per site.

5.4 A Quantum Limit on Parallel Computation

What are the ultimate limits imposed on the computational process, as a result of the fact that computation is inevitably done with physical degrees of freedom, subject to the laws of physics. Landauer[99]

In equation (5.71) of the previous section, we have seen that the maximum parallel computation rate $\gamma_{\max}(M)$ scales linearly with the size M of the system. And, at least for the one-dimensional 2-body automaton, the coefficient of proportionality was $\eta = 2/\pi$.

In this section, we want to explore the physical origin of the linear scaling coefficient. We show that the result $\eta = 2/\pi$ is a particular example of a more general limit, $\eta < 1$. That is, the efficiency $\eta = \gamma_{\max}(M)/M$ of parallel computation is strictly less than 1 for any nontrivial computational system whose computing elements are governed by quantum mechanics. The more general result can be obtained using an argument akin that use by P. W. Anderson[3] to set bounds on the ground state energy of antiferromagnets.

Recall the variational technique for obtaining upper bounds on the actual ground state energy E_0 of a quantum-mechanical Hamiltonian $\widehat{H}$ that acts on wavevectors in a Hilbert space $\mathcal{H}$. By restricting attention to a parametrized family $\{|\psi(\vec{\alpha})\rangle\} \subset \mathcal{H}$ of test wavevectors, we evaluate $E(\vec{\alpha}) = \langle\psi(\vec{\alpha})|\widehat{H}|\psi(\vec{\alpha})\rangle$, vary $\vec{\alpha}$ to obtain the minimum $E(\vec{\alpha}_{\min})$ of $E(\vec{\alpha})$. Implicit in this procedure is the observation that $E(\vec{\alpha}_{\min}) \geq E_0$ because the parametrized set $\{|\psi(\vec{\alpha})\rangle\}$ always constrains, rather than enlarges, the possible wavevectors in $\mathcal{H}$.

The argument that will be used here depends crucially on this notion of constraint, so we refer to it as the "constraint argument". However, what we want to obtain here is not an upper bound on the ground state energy, but an upper bound on the maximum eigenvalue of $\widehat{\Gamma}$. This is analogous to obtaining a lower bound to the ground state energy E_0.

Our argument will be seen to apply whenever the operator whose eigenvalues we want to bound has two properties: it should be of N-body form; that is, it should consist of a sum over clusters of terms that involve no more than N degrees of freedom. Secondly, the operator should be a normal operator[1].

The twist is that, instead of applying the constraint argument to the actual ground state on one hand and a relatively constrained parametrized wavefunction on the other, the role of the parametrized wavefunction is played by the actual state $|\gamma_{\max}\rangle$ with maximum computation-rate, while the role of the actual ground state is now

[1]That is, if the commutator $[\widehat{\Gamma}_n, \widehat{\Gamma}_n]$ vanishes.

played by a tensor product of states, each of which maximizes the computation rate of an individual non-interacting cluster.

In analogy to the observation that $|\psi(\vec{\alpha})\rangle$ is constrained by the parametrization, we observe that $|\gamma_{\max}\rangle$ is constrained, relative to the case of non-interacting clusters, by the requirement that it be maximum for the entire system of interacting clusters. From this, it follows that the sum over non-interacting clusters of their individually attainable maximum computation rates is an upper bound to the computation rate attainable by the interacting system. From the computational point of view, this is simply the statement that when processors need to communicate with one another, the computation rate decreases. Physically, the communication between processors corresponds to the interactions between clusters.

As an example, suppose the exact solution were not available and we wanted to obtain an upper bound on the largest eigenvalue of the operator $\widehat{\Gamma}$ for the quantum cellular automaton model. Note that

$$\widehat{\Gamma} = \sum_{n=0}^{M-1} \frac{1}{i}[c_n^\dagger c_{n+1} - c_n c_{n+1}^\dagger] = \sum_{n=0}^{M-1} \widehat{\Gamma}_n. \tag{5.78}$$

can be written as a sum of local two-body operators $\widehat{\Gamma}_n$ involving creation and annihilation only on sites n and $n+1$. Strictly speaking, of course, $\widehat{\Gamma}_n$ is an operator on the full M-site Hilbert space $\mathcal{H}$, but it can be straightforwardly identified with its restriction $\widehat{\Gamma}'_n$ to the four-dimensional space $\mathcal{H}_n \otimes \mathcal{H}_{n+1}$. In the occupation number basis on sites n and $n+1$ we find

$$\widehat{\Gamma}'_n = \frac{1}{i}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.79}$$

which has eigenvalues, $\gamma_j^{(n)} \in \{-1, 0, 0, 1\}$.

Letting $|\gamma_{\max}(M)\rangle$ denote an eigenvector corresponding to the maximum eigenvalue of $\widehat{\Gamma}$ for a system of M sites, we can use (5.78) to write

$$\begin{aligned}\gamma_{\max}(M) &= \langle\gamma_{\max}(M)|\widehat{\Gamma}|\gamma_{\max}(M)\rangle \\ &= \sum_{n=0}^{M-1}\langle\gamma_{\max}(M)|\widehat{\Gamma}_n|\gamma_{\max}(M)\rangle \ .\end{aligned} \tag{5.80}$$

We will obtain the desired bound on $\gamma_{\max}(M)$ by applying the constraint argument in order to obtain a bound on each $\langle\gamma_{\max}(M)|\widehat{\Gamma}_n|\gamma_{\max}(M)\rangle$ for $n = 0, ..., M-1$.

The full Hilbert space can be decomposed in two distinct ways corresponding to the two clusterings of the 2-body automaton: either as

$$\mathcal{H} = (\mathcal{H}_0 \otimes \mathcal{H}_1) \otimes (\mathcal{H}_2 \otimes \mathcal{H}_3) \otimes \cdots \otimes (\mathcal{H}_{M-1} \otimes \mathcal{H}_{M-1}) \tag{5.81}$$

or as

$$\mathcal{H} = (\mathcal{H}_{M-1} \otimes \mathcal{H}_0) \otimes (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes \cdots \otimes (\otimes\mathcal{H}_{M-3} \otimes \mathcal{H}_{M-2}). \tag{5.82}$$

If we choose to decompose $\mathcal{H}$ according to (5.81) then, since the $\widehat{\Gamma}_n$ are Hermitian (and hence normal), we can form a complete basis for $\mathcal{H}$ by taking tensor products of the eigenstates $\{|\gamma_j^{(n)}\rangle|\ 1 \leq j \leq 4\}$ of the $\widehat{\Gamma}_n$ for n even. Similarly, if we chose to decompose $\mathcal{H}$ according to (5.82) we can form a complete basis for $\mathcal{H}$ by taking tensor products of $\{|\gamma_j^{(n)}\rangle|\ 1 \leq j \leq 4\}$ for n odd.

In order to obtain a bound on $\langle\gamma_{\max}(M)|\widehat{\Gamma}_n|\gamma_{\max}(M)\rangle$ for n even, we chose the decomposition (5.81) and write

$$|\gamma_{\max}(M)\rangle = \sum_{n'\ \text{even}}\sum_{j=1}^{4} c_j^{(n')}|\gamma_j^{(n')}\rangle \ . \tag{5.83}$$

The constraint argument applied to the operator

$$\widehat{\mathbf{1}} \otimes \widehat{\mathbf{1}} \otimes \cdots \otimes \widehat{\Gamma}_n \otimes \cdots \otimes \widehat{\mathbf{1}} \otimes \widehat{\mathbf{1}} \tag{5.84}$$

in the state $|\gamma_{\max}(M)\rangle$ then immediately gives us

$$\langle\gamma_{\max}(M)|\widehat{\Gamma}_n|\gamma_{\max}(M)\rangle = \sum_{j=1}^{4} |c_j^{(n)}|^2 \gamma_j^{(n)} \leq \max_j\{\gamma_j^{(n)}\} = 1. \tag{5.85}$$

Upon substitution into (5.80), (5.85) gives the desired bound on η,

$$\eta \equiv \frac{\gamma_{\max}(M)}{M} \leq 1 \ . \tag{5.86}$$

The equality in (5.86) only holds if $|\gamma_{\max}(M)\rangle$ is in fact also an eigenstate of (the extension of) each local operator $\widehat{\Gamma}_n$. In other words, only if the actual state $|\gamma_{\max}(M)\rangle$ is in fact just a simple tensor product of the "local states" $|\gamma_{\max}^{(n)}(M)\rangle$. This can be the case only if the global system is merely a collection of non-interacting local subsystems.

Since any non-trivial computation requires communication between sites and since communication is inevitably realized by physical interactions, the equality in (5.86) never holds:

$$\gamma_{\max}(M) < M. \tag{5.87}$$

The generality of this result is somewhat greater than may appear. For instance, it does not depend on the dimensionality of the space. It depends only on the ability to decompose the appropriate computation-rate operator into a sum of local cluster operators $\widehat{\Gamma}_n$ as in (5.78). In order to apply the constraint argument, we must be able to compose an orthonormal basis for the entire Hilbert space by taking products of the eigenstates of the $\widehat{\Gamma}_n$. If $\widehat{\Gamma}_n$ is normal, then its eigenvectors form an orthonormal basis for the space on which $\widehat{\Gamma}_n$ is defined and we can construct the required orthonormal basis for the entire Hilbert space. The class of normal operators is fairly broad; it includes many commonly encountered types: unitary, hermitian, anti-unitary, anti-hermitian. It is therefore reasonable to expect that, in the absence of severe pathology, $\widehat{\Gamma}_n$ will often be normal.

5.5 Experimental Realizations

In Section 5.1.3 we have shown that the long-range synchronization of the one-dimensional quantum cellular automaton is governed by the Hamiltonian

$$\widehat{H}_{\mathrm{sync}} = \sum_{n=0}^{M-1} \sigma_n^+ \sigma_{n+1}^- + \sigma_n^- \sigma_{n+1}^+, \tag{5.88}$$

which we identified as a type of one-dimensional XY model[104]. Elsewhere[22] we have shown that this form of cellular automaton requires no more than six bits per site to achieve universal computation. Together, these suffice to show that quantum computation need not involve unphysical, highly-contrived Hamiltonians. Instead, the arbitrarily defined part of the Hamiltonian can be restricted to act on a local computational subsystem $\mathcal{H}_{\mathrm{comp}}$ involving no more than six spins[22]. The long-range synchronization of these subsystems is mediated by the simple, physically-realistic XY Hamiltonian. However, the ultimate standard of realism is not simplicity alone, but what Nature has seen fit to include in its repertoire. In this section, we review the experimental search for real systems governed by the one-dimensional XY Hamiltonian H_{sync}.

The search for experimental attention has focused on certain rare earth compounds because both theoretical arguments based on crystal symmetry[1] and EPR (electron paramagnetic resonance) studies[150] indicated that they were likely candidates. The two systems that have been investigated most extensively are praseodymium trichloride ($PrCl_3$) and praseodymium ethyl sulfate ($Pr(C_2H_5SO_4)_3 \cdot 9H_2O$, or PrES). In both crystals, the Pr^{3+} ions have the trigonal point group C_{3h} as shown in Fig. 5-19 for PrES. Paramagnetic resonance experiments[150] have shown that for isolated Pr^{3+} ions in yttrium ethyl sulphate the resonant absorption transition is primarily due to electric rather than magnetic dipoles associated with the ions. But a simple parity argument shows that the presence of the σ_h symmetry operation (reflection through the plane perpendicular to the z-axis) forbids the existence of any electric dipole moment along the z-axis[1, p.661]. Since the chains of Pr^{3+} ions along the z-axis are known to order antiferromagnetically, the net interaction between a chain of spins and its neighboring chains may be small. So we might reasonably expect the electric dipoles associated with the Pr^{3+} ions to interact via a Hamiltonian of the form

$$\hat{H} = \sum_{m=0}^{M} J_{\perp}(\sigma_m^x \sigma_{m+1}^x + \sigma_m^y \sigma_{m+1}^y) + J_z \sigma_m^z \sigma_{m+1}^z \tag{5.89}$$

with $J_{\perp} \gg J_z$.

In the extreme anisotropic limit $J_z = 0$ this Hamiltonian has precisely the form of H_{sync}, the Hamiltonian that synchronizes the quantum cellular automaton.

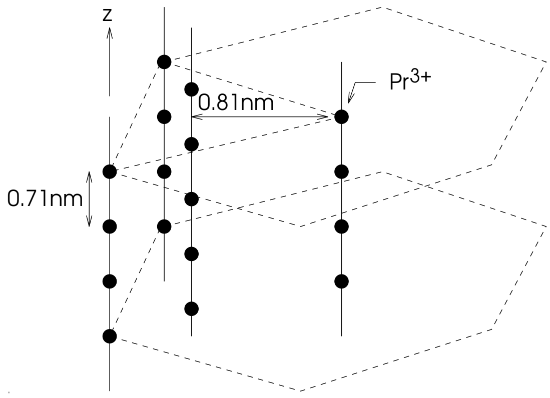

Figure 5-19: Location of Pr^3+ ions (solid dots) in praseodymium ethyl sulphate ($Pr(C_2H_5SO_4)_3 \cdot 9H_2O$). Paramagnetic resonance experiments[150] strongly suggest that along the chains nearest neighbors are coupled by electric dipole interactions with $J_z \ll J_x = J_y$, while interactions between chains are negligible.

The effective values of $J_{\perp}$ and J_z between pairs of Pr^{3+} ions can be determined experimentally via ESR (electron spin resonance). To determine $J_{\perp}$ and J_z in $PrCl_3$, a crystal of $LaCl_3$ is doped with enough Pr^{3+} to produce a significant number of nearest neighbor pairs along the z-axis. Since the spin-spin interaction between the Pr^{3+} ions shifts the location of the resonance peaks in the ESR experiment, $J_{\perp}$ and J_z can be determined by comparing the location of the peaks with their location in a similar experiment on a crystalline sample containing only isolated Pr^{3+} ions. Similarly, the value of $J_{\perp}$ and J_z in praseodymium ethyl sulphate ($Pr(C_2H_5SO_4)_3 \cdot 9H_2O$, abbreviated PrES) can be determined from ESR experiments on Pr^{3+} pairs in LaES. In a series of such experiments, Culvahouse *et al.*[37, 36, 38, 52] have measured the

	$PrCl_3$	PrES
$J_\perp/k_B$	2.85± 0.14K	0.6720± 0.0010K
J_z/k_B	0.11± 0.14K	0.0105± 0.0004K
$J_\perp/J_z$	25.9	64.0

Table 5.2: Experimentally measured coupling constants between Pr^{3+} ions in praseodymium trichloride and praseodymium ethyl sulphate[37, 36, 38, 52] determined from electron spin resonance (ESR) absorption.

values shown in Table 5.2.

Since the transverse coupling constant $J_\perp$ is greater than the longitudinal coupling J_z by one to two orders of magnitude in both PrES and $PrCl_3$, these measurements indicate that the effective Hamiltonians of these crystals is of nearly the same $1D$ XY form as the Hamiltonian that governs the quantum cellular automaton.

This suggests that other experimentally measurable quantities may agree with predictions derived from the $1D$ XY Hamiltonian. Unfortunately, the occurrence of the exponentiated fermion-number operator $\widehat{N_F}$ throughout the theory has made it difficult for theorists to use the exact solution of the 1D XY Hamiltonian (5.43) to predict the values of most experimentally accessible observables. However, theoretical expressions for two observables, the specific heat C_V and the transverse electric susceptibility χ_{xx} have been derived by Katsura[82] and Capel[27], respectively.

For the $PrCl_3$ system, Harrison *et al.*[65] measured both the specific heat and transverse susceptibility at temperatures of 1K − 10K. Folinsbee *et al.*[52] measured the same two quantities in the PrES system. All four measurements show remarkable agreement between theory and experiment. Data from Refs. [52] and [65] are reproduced below.

The experimental evidence clearly suggests that the effective Hamiltonian governing the dynamics of Pr^{3+} ions in crystalline $PrCl_3$ and PrES is the same as the Hamiltonian (5.43) that governs the synchronization of the one-dimensional quantum cellular automaton.

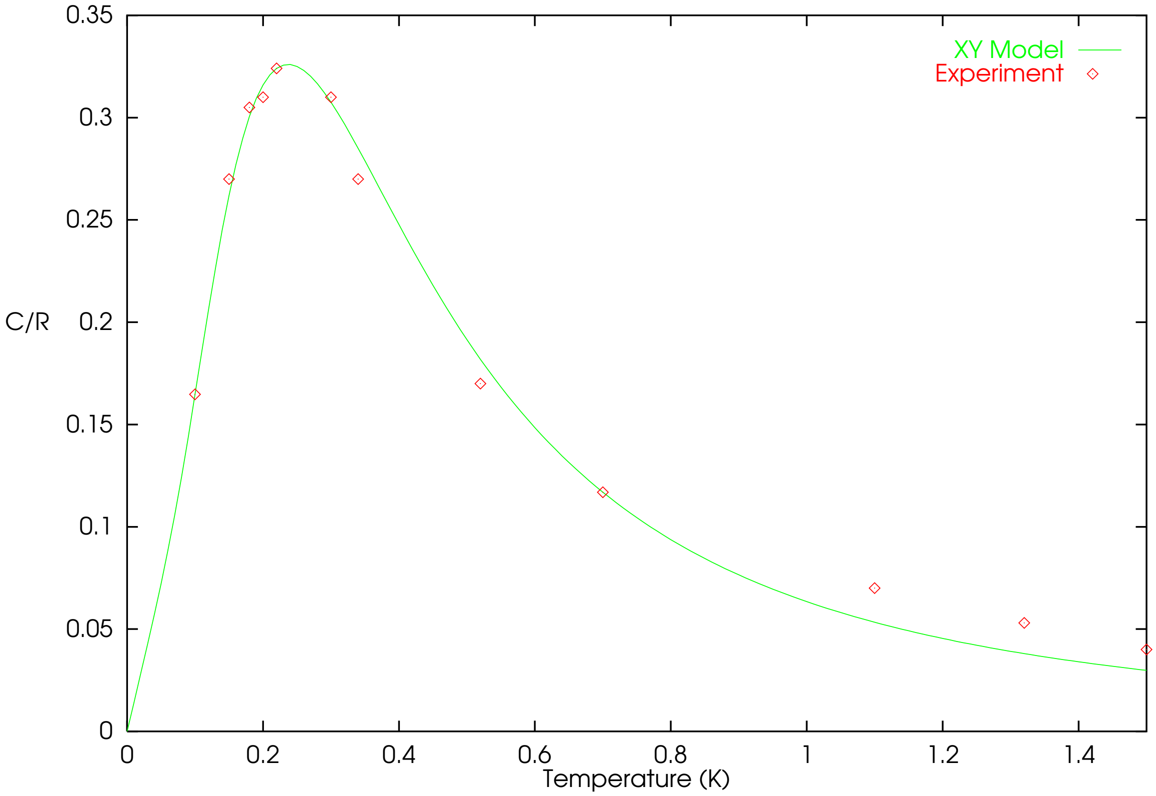

Figure 5-20: Specific heat of PrES. Theoretical prediction of the 1D XY model (dashed line) and experimentally measured values (solid and open circles) are shown.

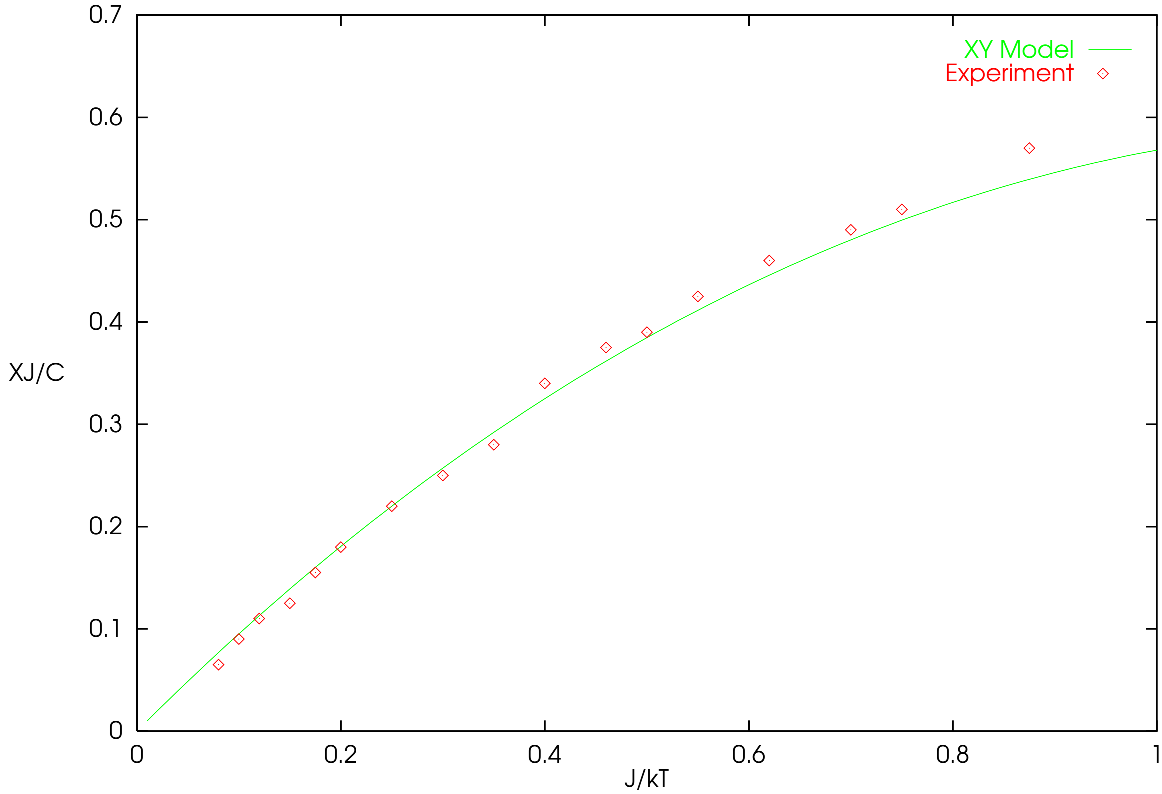

Figure 5-21: Electric susceptibility χ_{xx} of PrES.

5.6 Discussion

In this chapter we have attempted to address some of the well-placed criticism[99] that has been lodged against Hamiltonian models of quantum computation, principally the criticism that the Hamiltonians are so physically artificial that the physical relevance of the models themselves is brought into question. Here we have presented a quantum cellular automaton all of whose important long-range properties are described by a simple one-dimensional anisotropic antiferromagnet—the one-dimensional XY model. But the XY model is more than just a simple, realistic-looking Hamiltonian. It predicts many low-temperature properties of the rare earth compounds praseodymium trichloride and praseodymium ethyl sulphate that have been experimentally verified.

Some artificiality remains in $\widehat{H}_{\mathrm{comp}}$, which describes the internal short-range dynamics of each cell of the cellular automaton. But even if we require the quantum cellular automaton to be computation-universal, the artificiality is confined to a uniformly repeated unit cell of no more than six spins[22]. It is amusing to note that the C_{3h} symmetry of the praseodymium compounds gives each Pr^{3+} ion exactly this many nearest neighbors. If the dynamics of the small but arbitrary unit cell could ever be realized, something completely unexpected will have come to pass. The *Gedanken* computers of Benioff and Feynman will have become real physical systems.

But even if (as seems likely) this never occurs, the quantum cellular automaton presented here can serve as an analytically solvable yet physically realistic model of quantum computation. It may also help open an experimental window to understanding how two flourishing theoretical enterprises rooted in the early twentieth century —quantum theory and the implicitly-classical theory of computation—will ultimately be reconciled.

Chapter 6

Ballistic Computation in an Array of Quantum Dots

Few-body automata can also guide us in finding ways that—at least in principle—let us extract universal computational behavior from realistic models of nanostructures. This chapter is adapted from a paper that described the first computation-universal cellular automaton that used realistic effects in a quantum dot array as the basis of its computational dynamics. In particular, it is based on the quantum dot structure proposed by Obermayer, Teich and Mahler[130, 129, 140, 128, 108], and the interactions between nearby dots that they have predicted.

Growing concern about the future of device miniaturization [43, 152, 42, 134, 66] has recently led a number of device physicists [11, 10, 28, 50, 68, 53] to take a fresh look at one of the oldest[1] paradigms for computation, cellular automata. Unfortunately, as we mentioned briefly in Sec. 2.6, conventional cellular automata (CA), such as the well-known "Game of Life"[20], are not readily adaptable to serve as an architectural paradigm for arrays of submicron devices. Our purpose in this chapter is two-fold. First, we explain why the conventional form of CA is not suitable for the purposes device physicists contemplate. Secondly, we show how few-body cellular automata overcome the limitations of conventional CA and, in some circumstances, can provide

[1] Cellular automata were introduced around 1950 by Ulam[146], von Neumann[147] and Zuse[158].

a physically realistic model for submicron device arrays. But before we approach these issues, we should understand the trends that are leading some device physicists to revisit the venerable concept of cellular automata.

Two kinds of limits to the continued down-scaling of device dimensions are apparent: limits to scaling individual MOS devices and limits imposed by the presence of long interconnections.

The scaling of individual MOS devices is limited by the range of achievable materials properties. In order to maintain reliability in the face of thermal fluctuations and manufacturing variations, operating voltages must be kept above a certain minimum. Unfortunately, since the range of achievable dielectric constants is limited, as one continues to operate progressively smaller devices at this minimum voltage, one inevitably produces larger electric fields within the device. As these fields increase, the physical assumptions on which the logical operation of MOS-type devices is based will eventually break down, and the scaled device will become computationally useless.

Because improved materials (and the ingenuity of designers) play some role, it is impossible to give a precise limit to the scaling of MOS devices, but a number of careful analyses[42, 134, 66, 11] suggest that channel lengths below about 0.1μ would require techniques so extraordinary as to be uneconomical. In an effort to continue miniaturization past this point, many researchers are investigating novel, nanometer-scale devices based on qualitatively different physical principles[135, 90].

The second barrier to continued downscaling, the interconnect problem, is not fundamentally a physical limit, but an architectural one. Combinatorial analyses[45] have shown that current architectures produce a distribution of wire lengths containing so many long wires that their clock speed would ultimately be limited by τ_{RC}, the approximate time scale required to charge or discharge long wires when transmitting a signal. However, the length of the longest wires is not a reliable metric. If an architecture requires only a few long wires, special techniques can be employed to alleviate their delays. A better indicator of the limiting clock speed of a circuit is based on the average length $\langle l \rangle$ of its wires.

The average length can be related to a simple architectural property, the so-called

Rent exponent, which characterizes how the number P of input/output ports required by a circuit scales with the number of gates G. For a wide variety of circuits, the function $P(G)$ has been found to follow "Rent's rule":

$$P = kG^{\rho} \tag{6.1}$$

where k is a constant and the Rent exponent ρ is typically between 1/2 and 1. Consider the architectural assumptions that lead to those two extremal cases. If every gate had its own connection to the outside world, then clearly we would be in the $\rho = 1$ limit. At the other extreme, in a two-dimensional array of gates with area $A \propto G$, only nearest neighbors are connected, so input/output ports only occur along the perimeter, and therefore the two-dimensional array has a Rent exponent $\rho = 1/2$. Current architectures for gate arrays exhibit Rent exponents of $\rho \sim 0.6$-0.7 [138, 100, 34]. Because ρ is an exponent, this excess over the ideal limit $\rho = 1/2$ becomes significant for large enough arrays.

The Rent exponent can be related to the average wire length, which in turn limits the maximum speed of the system. Donath[45, 48] has shown that the average wire length $\langle l \rangle$ depends on the gate count G and the Rent exponent ρ as[2]

$$\langle l \rangle \sim \begin{cases} G^{\rho-\frac{1}{2}} & \text{if } \rho > 1/2 \\ c_0 & \text{if } \rho \leq 1/2 \end{cases} \tag{6.2}$$

where c_0 is a constant. Modeling a wire as a transmission line with resistance R and capacitance C per unit length yields a simple diffusion equation with solutions characterized by a time delay $\tau_{\text{RC}} \sim \langle l \rangle^2$. This time scale then sets the scale for the maximum clock speed.

Letting $\delta \equiv \rho - 1/2$, we see from (6.2) that if $\delta > 0$, then $\tau_{\text{RC}} \sim G^{2\delta}$ eventually diverges as the circuit complexity G grows. As the size G of the circuit increases, the maximum system speed decreases. However, we could prevent interconnection delays

[2]Actually, for ρ exactly equal to 1/2, Donath finds $\langle l \rangle \sim \log G$, but the distinction is insignificant for our purposes.

from limiting overall system speed if we were able to achieve $p \leq 1/2$. By eq. (6.2), this corresponds to the case of fixed interconnect length. This line of reasoning has led some device physicists to consider modes of computing that require only short local interconnections; that is, cellular automata.

Several groups[11, 10, 28, 50, 68, 53, 9, 49, 148] have noted the formal similarities between regular arrays of interacting nanometer-scale devices and two-dimensional CA . In existing cellular automata machine architectures[126, 144, 145, 116], the CA evolution is determined by an external lookup table, and requires additional circuitry and interconnections beyond that needed to store the state of the cells. If instead of a lookup table the existing physical interactions between neighboring devices could somehow be used directly to produce the dynamics of a computation universal cellular automaton, then the logical operation of each cell would require only local interconnections.

Over the past decade, device physicists have expended considerable effort in exploring the variety of interactions present in nanostructures (see[135, 90, 49]). But CA researchers have thus far not made a reciprocal effort aimed at exploring the question, "What form must a cellular automaton take when its dynamics is produced not from an arbitrarily definable lookup table, but directly from known physical interactions?"

To answer this question, we must first enlarge the abstract notion of CA to include a new kind of *physical* CA ; that is, cellular automata whose structure reflects important properties of physical interactions, such as their range of interaction.

If we try to take the mathematical definition of conventional von Neumann (or Moore) neighborhood CA as a literal prescription for a physical implementation, we are immediately faced with serious problems. If we simply arrange nanometer-scale devices in a dense rectangular array, the device cells may interact with each other, but we have given up all control over which cells interact with which neighbors, and when they interact. In general, the state of a cell at any given time will depend on the states of a multitude of nearest, next-nearest, and further neighbors at a multitude of times, depending on the effective range of the interaction. The resulting dynamics may be of interest in its own right, but it has little more than its spatial discreteness in common with cellular automata. Its range of computational utility will likely be limited to the simulation of a few systems that coincidentally possess a dynamics similar to its own.

If there is to be any hope of performing more general-purpose computation with these arrays, we need a less literal interpretation of what it means to impose the structure of a cellular automaton on a physical system composed of interacting nanometer-scale subsystems. We need an interpretation that recognizes certain common properties of physical interactions—in particular, that their strength varies smoothly, often with some characteristic screening length Λ. Furthermore, this interpretation should provide enough structure to give us some control over both (a) which devices interact as neighbors and (b) when these interactions occur.

In this chapter we present an alternative line of reasoning that leads to few-body automata. In the viewpoint introduced here, we proceed by finding a simple way of endowing the class of partitioning cellular automata[126, 110] (a generalization of lattice-gas automata[110, 46, 56, 64, 63]) with the two desired properties properties described above. Because this transformation entails replicating each cell, we call this special case of few-body cellular automata *replica* CA and the transformation itself is called the *replication* transformation; the PCA from which the replica CA is generated will be called its *progenitor*. We illustrate the utility of replica CA by using a (4-body) replica CA to solve a previously posed problem: that of finding a computation-universal cellular automata architecture that uses an idealized form of the single-electron quantum-dot device proposed by Obermayer *et al.*[130, 129, 140, 128, 108]. For a number of practical reasons not relevant to the present discussion, the resulting construction is not easily amenable to error-correction [3] (or even to fabrication), but it will serve us well enough as a simple example of how replica CA might be applied to more realistic nanoscale devices.

[3] Zurek[157] has shown that the effective phase space of the classical billiard ball model doubles at every collision.

6.1 Partitioning Cellular Automata

Partitioning automata (PCA) are characterized by the fact that, unlike conventional von Neumann automata[146], their neighborhoods can be regarded as time-dependent clusterings of cells, each cluster being completely isolated from every other.

In a von Neumann automaton, the next state of a cell is determined by its own state and by the current state of a time-independent set of neighbors (a typical neighborhood in two dimensions consisting of N, S, E and W neighbors). In contrast, a two-dimensional partitioning automaton begins by partitioning the entire space into a large number of non-overlapping *clusters* of cells; the cells of each cluster then evolve to a new state that depends only on the current state of cells in that cluster. That is, each cell belongs to one and only one neighborhood at any given time. At the next discrete time, a different partition, or *clustering*, may be used. Cells that were neighbors (i.e., in the same cluster) at the previous time-step are typically no longer neighbors; cells that were not neighbors previously may become neighbors in the new partition.

This seemingly inelegant clustering mechanism has had an important benefit to lattice-gas theorists: the global conservation laws that produce hydrodynamics at large length scales can be incorporated into the dynamics of a simulation simply by ensuring that within each cluster the corresponding local conservation law is obeyed. Fig. 6-1 shows the form of a two-dimensional PCA where the clusters are 2×2 squares of cells, and where two distinct clusterings (partitions) are used.

6.1.1 Tilings, Partitions, and Clusters

In order to describe the construction and operation of replica CA we will first need to characterize ordinary, unreplicated PCA a little more formally. Since our principal application is to planar arrays, we can define two-dimensional PCA in the notation of tiling theory[61]; this notation will also give us a concise way to describe some of the less familiar lattices generated by the replication transformation. The relation between tilings and PCA is intuitive: we associate individual tiles with the cells of

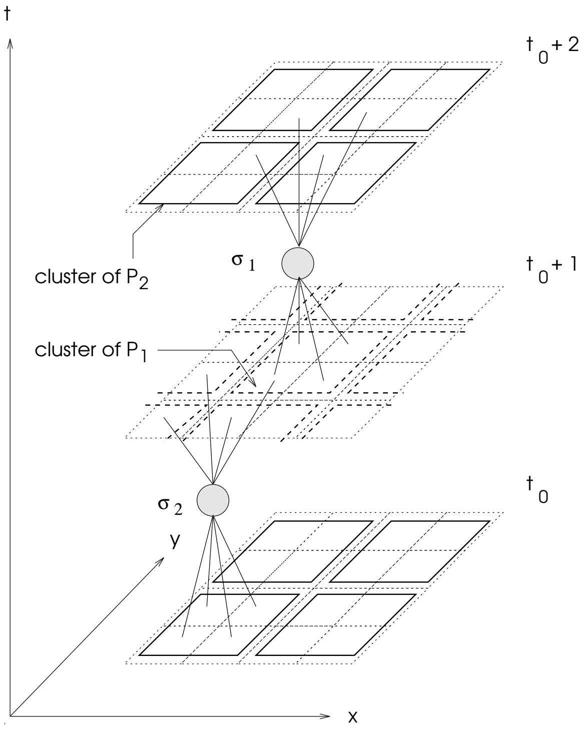

Figure 6-1: Partitioning CA with 2×2 cluster neighborhoods.

the CA , and each partition of a PCA will be regarded as a coarser tiling of the plane by the clusters.

A *tiling* is defined as a countable family of closed sets with pairwise-disjoint interiors, called *tiles*, whose union is the entire plane. Although both PCA and the replication transformation can be extended to general tilings, for simplicity we consider only *regular*[4] tilings, where each tile is a regular n-gon. Three such tilings of the plane exist: by equilateral triangles, by squares and by regular hexagons; these will suffice to illustrate the replication concept. Generalization to both higher and lower dimensions, and to irregular tilings is straightforward, and will not be discussed further.

Definition. A *partition*, $\mathcal{P}(\tau)$, of a tiling τ is itself a tiling of the plane such that each tile of $\mathcal{P}(\tau)$, known as a *cluster*, contains an integral number $n \geq 2$ of tiles of τ.

---

[4]This use of the term "regular" is, of course, totally unrelated to that in Sec. 2.2, where it referred to the behavior of interactions at $r = 0$ and $r \to \infty$.

In partitioning automata, clusters play the role of neighborhoods. To obtain a nontrivial dynamical system, more than one partition must be employed. When the cells of the PCA are clustered according to the m^{th} partition, it is said to be in its m^{th} phase.

Definition. A n *m-phase k-state partitioning* cellular automaton is an $(m+1)$-tuple, $m \geq 2$,

$$[\tau, (\mathcal{P}_1(\tau), \sigma_1), ..., (\mathcal{P}_m(\tau), \sigma_m)],$$

where τ is a tiling of the plane, each $\mathcal{P}_i(\tau)$ is a distinct partition of τ, and the functions $\sigma_i : Z_k^{n_i} \rightarrow Z_k^{n_i}$ are the i^{th}-phase evolution operators; n_i is the number of tiles contained in each cluster of $\mathcal{P}_i(\tau)$, and Z_k denotes the integers modulo k.

The operation of a PCA corresponding to the square tiling of the plane is shown in Fig. 6-1. The clusters of the two partitions, $\mathcal{P}_1$ and $\mathcal{P}_2$, are indicated by the heavy dashed and heavy solid lines, respectively; the tiles of the underlying cellular array are drawn with light dotted lines. The two evolution operators, σ_1 and σ_2, are indicated by circles; both are binary-valued functions $\sigma_i : Z_2^4 \rightarrow Z_2^4$. The lowest level of the figure depicts an initial clustering $\mathcal{P}_2$ of cells into 2×2 clusters. Each cluster then evolves according to σ_2, the new states (not shown in the figure) replacing the old as time advances to $t_0 + 1$. At $t_0 + 1$, the cells are again grouped in 2×2 clusters, but with clusters defined by the partition $\mathcal{P}_1$. The clusters of $\mathcal{P}_1$ evolve according to the operator σ_1. At time $t_0 + 2$ the $\mathcal{P}_2$ clustering is again in force. This two-phase cycle repeats indefinitely. The replica CA architecture we will describe for the charge-transfer quantum dot device has as its progenitor a PCA with the form shown in Fig. 6-1.

6.1.2 Ballistic Computation

In order to show that the quantum dot system of Obermayer *et al.* is capable of universal computation in two-dimensions, we must focus on some particular computation-universal cellular automaton rule. One of the simplest schemes for performing universal computation with a physically-inspired (though highly idealized) interaction is

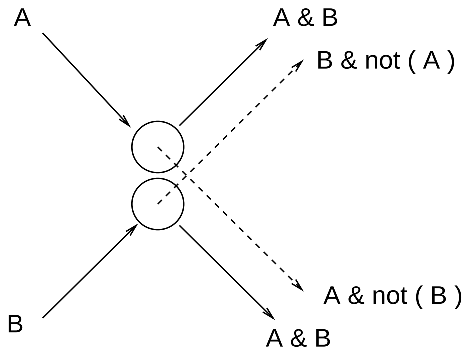

Figure 6-2: Computation universality of Fredkin's billiard ball model. A ball will be present in each output stream if and only if the indicated conditions on the input streams, A and B hold.

Fredkin's billiard ball model (or BBM) [55].

The billiard ball model is a classical-mechanical model of digital computation obeying the reversible equations of motion for a gas of finite diameter, infinitely hard billiard balls confined within a hard-walled container. The computational interpretation of its dynamics is straightforward. Streams, consisting of balls and gaps, play the role of wires carrying 1s and 0s in a digital circuit. The locus of a possible collision between two balls is interpreted as a two-input/four-output logic gate (Fig. 6-2) that computes the nonlinear, nonmonotonic Boolean functions $f_1(A, B) = A \wedge \bar{B}$ (meaning, A AND (NOT(B))) and $f_2(A, B) = B \wedge \bar{A}$. The rigid walls redirect the ball streams so that the output of one logic gate can become the input to another. According to a classical theorem[120] of switching theory, if one is given access to constant sources of 1s and 0s along with the ability to form arbitrary interconnections, then any nonmonotonic nonlinear element suffices to synthesize all computable Boolean functions. Therefore, by using these components to construct digital circuits, the evolution of a meticulously arranged initial state of the gas of billiard balls can be interpreted as the operation of a computer (see Fig. 6-2).

From this computation-universal but spatially continuous model, it is possible to define a computation-universal CA . Because Fredkin's BBM is both reversible and and particle-conserving, efforts to implement it directly as a conventional von Neu-

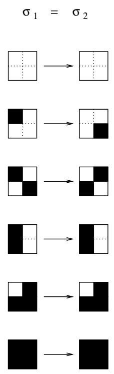

Figure 6-3: The transition table for the billiard ball CA .

mann CA only produced unwieldy constructions that required large neighborhoods and many states per cell[110]. However, by introducing PCA , Margolus[110, 114] was able to define a simple reversible and computation-universal cellular automaton that is based on Fredkin's BBM. In the PCA version, a single finite-diameter billiard ball is represented by a train of two tokens separated by at least one empty cell; the length of the train reflects the finite diameter of the ball.

The transition rule, which is the same for both phases (i.e., $\sigma_1 = \sigma_2$), is shown in Fig. 6-3, where the black squares represent a token and white squares empty space. The rule is invariant under 90° rotations of the 2×2 cluster, so it is completely defined by its action on the six rotationally inequivalent cluster configurations shown. (For example, the second row of Fig. 6-3, implies three additional symmetry-related cases: an isolated token in any of the four cells moves diagonally.)

It is easy to verify that this CA faithfully reproduces the logic of Fredkin's model, and thus inherits both its reversibility and its computation universality.

6.2 Spatial Replication Method

The ease with which PCA can incorporate physically conserved quantities makes them useful for simulating physical systems with conservation laws. But, like von Neumann CA , a direct translation of their mathematical definition still does not provide a usable *geometry* for implementing parallel nanoscale computation for at least two distinct reasons:

1. The cluster neighborhoods of PCA are even more physically unrealistic than the nearest-neighbor-only neighborhoods of von Neumann CA . Physical interactions (e.g. the Coulomb interaction) are generally strongest between nearest neighbors, weaker between next-nearest neighbors, and so on. In contrast, at any given time a cell of a PCA interacts *only* with others in the same cluster, even if these are not its nearest neighbors. Simultaneously, the cell ignores *all* cells in other clusters, even though some of them are in fact its nearest neighbors,

2. Because they vary arbitrarily from one time-step to the next (in general, $\sigma_i \neq \sigma_{i+1}$), the PCA evolution operators appear to make unrealistic assumptions about the physical interactions between device cells. Few interactions available at the nanometer scale can be made to exhibit arbitrary time-dependent behavior.

Replica CA overcome these problems by incorporating the formal notion of what we have termed a 'cluster' as an actual, spatially localized structure. Once localized, the cells in each cluster can in principle be screened from the influence of cells in other clusters. If such screening is possible in practice, we will see that the clustering structure of replica CA enables us to control which cells interact, and when they interact. In addition, since the replicas of a cell will be localized in distinct regions, we will be able to tailor their immediate environs differently, thereby making it possible to achieve different local evolution operators (i.e., $\sigma_i \neq \sigma_{i+1}$) without resorting to unrealistic time-dependent interactions.

6.2.1 Replication Transformation

The essential idea of spatial replication is this: each cell of an m-partition PCA is replaced by m distinct *replica* cells. The replica cells are linked into a ring[5] by auxilliary cells whose sole function is to shuttle data from one replica to the next. Any computation that would have occurred during the m^{th} phase at cell (i, j) of the progenitor PCA will instead occur in the replica CA at the locus of the m^{th} replica of site (i, j).

Like conventional CA, PCA possess a notion of contiguity, but not distance between cells; in particular, PCA lack a notion of the distance between the clusters of its partitions. Replica CA are specifically constructed to possess a well-defined intercluster distance, so that we can simultaneously obtain strong *intra*cluster interactions and weak *inter*cluster interactions. A replica CA is produced from a progenitor PCA by the three-step replication process $\mathcal{R}$: replication, dilation, and relinking, depicted in Fig. 6-4. The case shown, replication of a square lattice employing the two partitions of Fig. 6-1, is the one we will use for the array of charge-transfer quantum dots.

In Fig. 6-4, the first arrow indicates the combined *replication* and *dilation* steps of the transformation. Each cell is replicated a number of times equal to the number of partitions (in the case shown, twice). The resulting doubled array is then dilated by moving the replicated clusters apart in space. The magnitude of dilation required will depend on the range of the particular physical interactions between cells. The result is a disjoint collection of clusters, each cell having a single replica in any cluster to which its progenitor cell in the PCA belonged. For example, the progenitor cells ϵ, ϕ, κ, and λ at the top of Fig. 6-4 each belong to an even-phase cluster (2×2 solid squares) and an odd-phase cluster (2×2 dashed squares). Therefore, in the center and bottom of the figure they have even-phase replicas: ϵ', ϕ', κ', and λ', and odd-phase replicas: ϵ, ϕ, κ, and λ.

The second arrow represents the final step of the transformation, which *relinks* the

[5] If the progenitor PCA has two phases, as will be the case in our quantum-dot example, the ring degenerates into a segment.

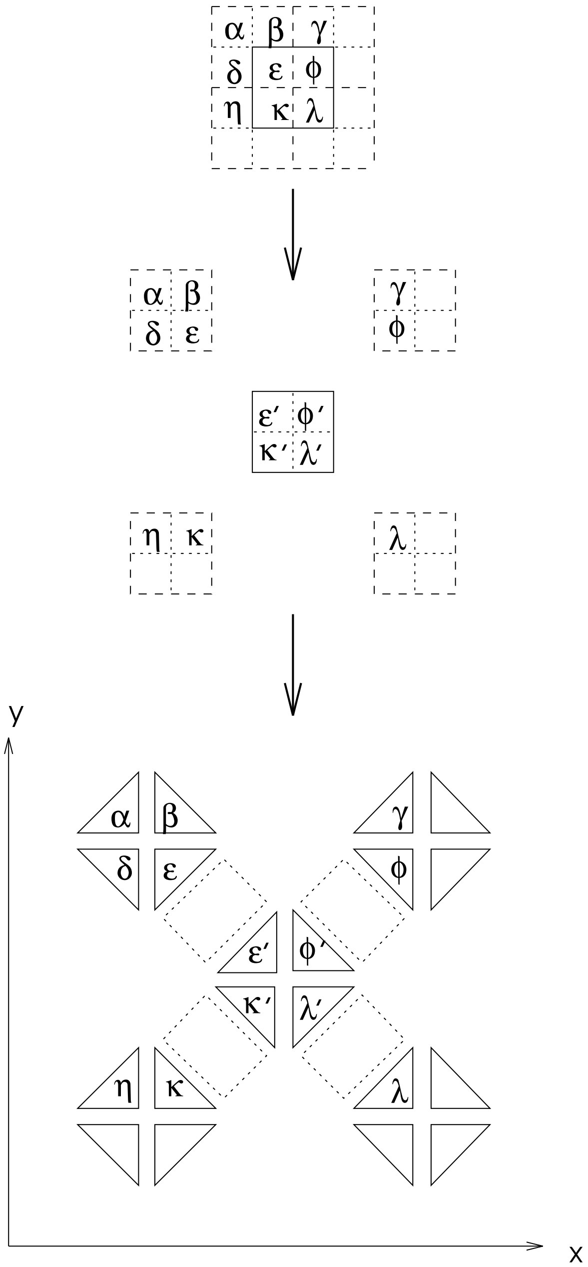

Figure 6-4: The replication transformation $\mathcal{R}$ applied to a 2-phase partitioning CA on the square lattice (top). The corresponding replica CA is shown at bottom. In the general case, $\mathcal{R}$ introduces m physical replicas of each site of the abstract automaton, where m is the number of partitions in the progenitor PCA .

replicas of each cell so that, for example, the data that occupy cell λ of the progenitor PCA can propagate between the two corresponding replica cells, λ and λ' in the replica CA . This requires the addition a minimum of $m - \delta_{m2}$ cells (shown as dotted rectangles in the bottom part of the figure) to link all replicas of the cell. Since $m = 2$ here, only a single cell is needed to relink the two replicas.

6.2.2 Time Evolution

At time t, only the clusters of $(t \bmod m)^{\text{th}}$ replicas will be occupied by data, and this is precisely the data needed to perform the m^{th} phase of the PCA evolution. Once this phase of the evolution (specified by σ_m) is complete, the relinking cells transport the new state of each cell of the cluster to the respective locations of their $(m+1)^{\text{th}}$ replicas, so that the next phase of the evolution can be performed. For example, at the bottom of Fig. 6-4, once data in the cluster containing ϵ', ϕ', κ', and λ', has evolved, the resulting contents of the cells are transported via the intervening relinking cells to the cells labelled ϵ, ϕ, κ, and λ, which reside in four different clusters. In those clusters the data again evolves, this time according to σ_{m+1}.

We can now see how replica CA alleviate the first problem described above, that of eliminating the unwanted influence of neighbors in other clusters. During the m^{th} phase, data occupying the m^{th}-phase clusters is literally clustered together in space. Because of this, we can reduce the effect of cells in other clusters by increasing the separation between the m^{th}-phase clusters. In effect, the structure of replica CA permits us to treat the intercluster separation as a parameter; it can be increased until physical isolation of clusters is obtained, while simultaneously preserving the logical operation of the evolution within the clusters.

For many physical couplings, there is some screening length λ_{SC} beyond which changes in the state of more distant cells have negligible effect. This is the class of interactions to which replica CA can be sensibly applied. By increasing the intercluster separation D until $D \gg \lambda_{\text{SC}}$, we can insure that our logical model, based on PCA , reflects physically realistic assumptions about the couplings used to implement it.

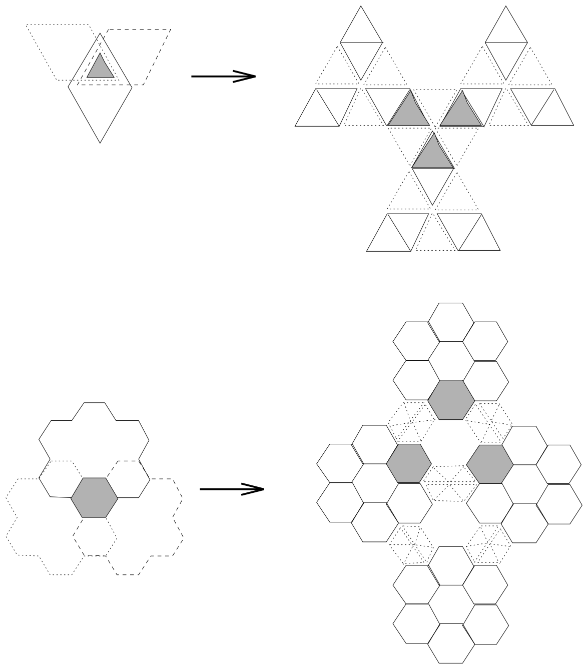

Figure 6-5: Spatial replication of 3 phase partitioning automata on triangular and hexagonal lattices.

Of course, many screenable interactions between nanoscale devices would require a value of D so large that construction of the resulting arrays would be technologically infeasible.

Moreover, the second obstacle enumerated above is simultaneously alleviated: the spatial clustering of cells at each phase allows us to implement rules that vary from phase to phase ($\sigma_i \neq \sigma_{i+1}$). To do so just entails finding a suitable *static* tailoring of the spatial environment around all i^{th} phase and $(i+1)^{\mathrm{th}}$ phase clusters, rather than the substantially more daunting task of finding a suitable time-dependent interaction.

One can see that the replication transform $\mathcal{R}$ is applicable to any partitioning automaton. The result of applying it to three-phase PCA on the other two regular tilings: the triangular tiling (which is denoted $\langle 3^6 \rangle$ in the tiling literature[61], to indicate that each vertex is the juncture of 6 regular 3-gons), and the hexagonal tiling (denoted $\langle 6^3 \rangle$) is shown in Fig. 6-5.

On the left of the figure, the outlines of representative clusters of each of the three partitions are shown; for the $\langle 3^6 \rangle$ tiling, the clusters contain two triangular cells; for the $\langle 6^3 \rangle$ tiling, they contain six hexagonal cells. The right side of figure shows the location of the three replicas of the shaded cell on the left. The relinking cells are shown on the right side of the figure as dotted cells. The resulting replicated tilings have two and three different vertex types, respectively. In the notation of tiling theory, the results of the replication transform on the tilings shown in Fig. 6-5 can be summarized

$$\mathcal{R} : (3^6) \longrightarrow (3^6; 3^4.6)_2 \tag{6.3}$$

$$\mathcal{R} : (6^3) \longrightarrow (6^3; 6^2.3^2; 3^6) \tag{6.4}$$

where, by convention (see [61, p.66]), the subscript indicates the second so-called enantiomorphic (i.e., mirror image) form of the tiling.

6.3 Charge-transfer Quantum Dot

In the preceding sections, we have introduced the replication transform $\mathcal{R}$, which systematically transforms abstract partitioning cellular automaton rules into the physically realizable form of a replica CA possessing a well-defined spatial structure. Below, we will illustrate the application of replica CA by generating a computation-universal CA architecture that employs a type of charge-transfer quantum dot device. This device was proposed and analyzed in substantial detail by Obermayer *et al.*[130]. In this section, we briefly review the essential features of their device.

6.3.1 Geometry

The geometry and energy-level structure of the charge-transfer quantum dot device are depicted in Fig. 6-6a. In the form investigated by Obermayer *et al.*, the device consists of two 4 nm wide quantum dots of GaAs separated by a 12 nm $\mathrm{Ga_{1-x}Al_xAs}$ barrier. The charge-transfer mechanism depends on the fact that the lowest conduction level and highest valence levels can be made to have radically different localization characteristics.

Because the effective mass in the conduction band is smaller than that in the valence band, a given band-offset produces a smaller effective barrier in the conduction band than in the valence band. For a small range ($0.8 < \beta < 1.8$) of values of the band-offset parameter

$$\beta \equiv \frac{E_{CB}^{\mathrm{GaAs}} - E_{CB}^{\mathrm{GaAlAs}}}{E_{VB}^{\mathrm{GaAs}} - E_{VB}^{\mathrm{GaAlAs}}}, \tag{6.5}$$

the effective barrier in the valence band is large enough to localize the two highest valence band states (denoted VB_L and VB_R) in the left- and right-hand wells respectively, whereas for the same β, the corresponding barrier in the conduction band is not sufficient to localize the lowest conduction state (denoted CB), which therefore remains delocalized over both wells and the barrier, as shown in Fig. 6-6b. In the remainder of this chapter, we will consider only these three states, and only in the case that (as indicated by the Fermi level, ϵ_F) a single electron populates one of the three levels. Computationally, the quantum dots will each represent a particular cell of a replica CA , and the presence of a single electron in one of the valence states will represent a bit.

Since we are using the charge-transfer device solely to illustrate the use of replica CA , we will henceforth ignore such important (and difficult) practicalities as how one might initialize an array of singly occupied quantum dots (or maintain its operation in the presence of significant sources of error); instead, we focus on applying replica CA to a fault-free array of ideal charge-transfer devices.

6.3.2 Single Device Dynamics

Obermayer and his co-workers proposed a switching dynamics based on the observation that, for certain geometrical and materials parameters, optically induced transfer of the single electron from one dot to the other is possible. The process is shown schematically in figures 6-7a–c. The upper part of each figure represents a top view of the double-dot device; the shaded area indicates roughly where the corresponding electronic wavefunction has appreciable amplitude. The lower part of each figure shows the corresponding energy level diagram for the two valence levels, VB_L and

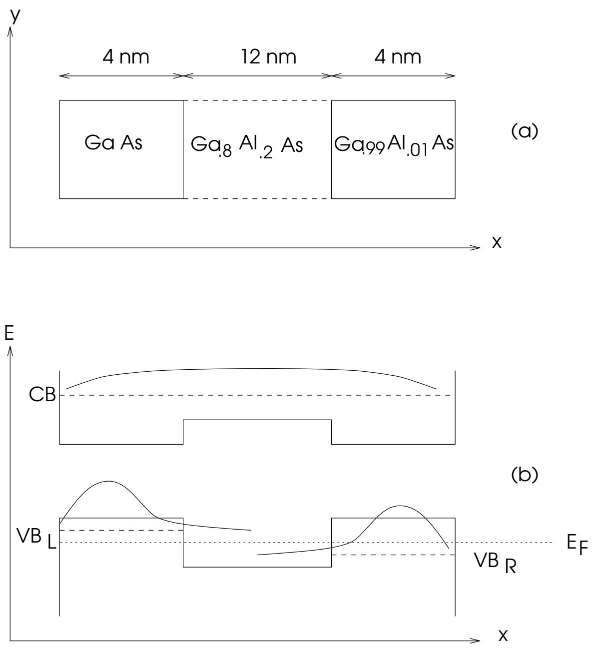

Figure 6-6: The charge-transfer quantum dot device. (a) Geometrical structure, (b) Confinement structure of two highest valence band and lowest conduction band states.

VB_R, and the conductance band level CB; electron occupancy is indicated by the small solid dot.

Suppose that initially the electron is localized in the left dot, as shown in Fig. 6-7a. Illumination with optical radiation tuned to the transition frequency ω of $VB_L \longrightarrow CB$ induces a transition to the delocalized conduction band level (Fig. 6-7b). This state has lifetime τ_r, and quickly decays to either VB_R, as shown in Fig. 6-7c, or back to VB_L.

Obermayer *et al.* point out that two time scales characterize this process: τ_r, the time it takes to decay from the transient state back to one of the states localized in the dots, and τ_d, the time for phonon-assisted tunneling through the barrier from one dot to the other. On the latter time scale, information is irretrievably lost and the system eventually comes to a computationally uninteresting thermal equilibrium. Since the fastest possible switching time τ_s is equal to the fastest relaxation time τ_r, a necessary prerequisite[130] for the system to be computationally useful is $\tau_r \ll \tau_d$.

Fortunately, the authors of refs. [128, 130] have shown that this particular device

Figure 6-7: Switching dynamics of an isolated charge-transfer quantum dot.

exhibits the necessary hierarchy of time scales, so that there exists an illumination time τ_s satisfying

$$\tau_r < \tau_s \ll \tau_d. \tag{6.6}$$

For the double quantum-dot illuminated on resonance, the authors apply a rotating-wave approximation to solve the Pauli master equation that governs the dynamics of the reduced density matrix for this system. Solutions for the diagonal elements, which give the population ρ_{ii} of each dot, indicate that if the system is illuminated by monochromatic light at frequency $\omega_{31} = E_{CB} - E_{VB_L}$ for a time $\tau_s \approx 10^{-8}s$, then the electron is transferred to the opposite dot with probability $p_{\mathrm{L}\to\mathrm{R}} > 0.99$ (see esp. Fig. 6a of [128]).

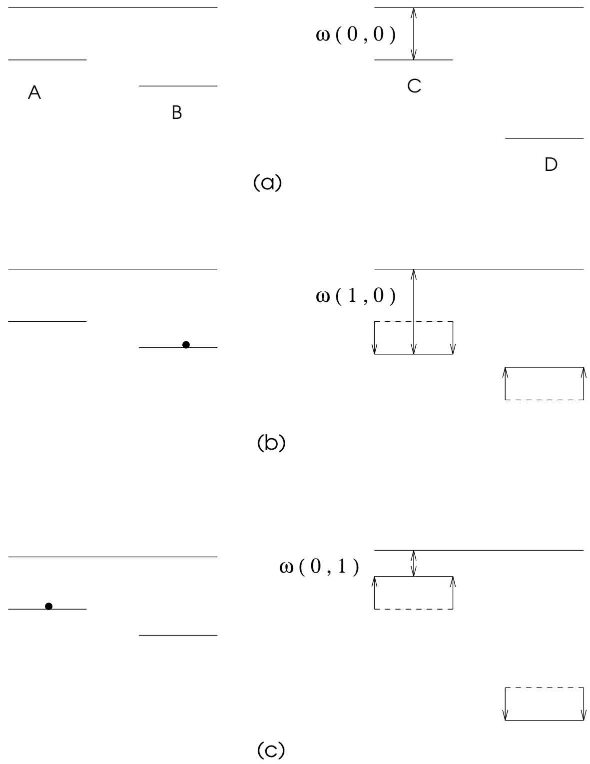

Figure 6-8: Conditional switching of the charge-transfer quantum dot device. Due to Coulomb effects, the laser frequency needed to transfer an electron from the left dot to the right dot, $\omega(N_1, N_2)$, depends on the numbers N_1 of nearest and N_2 of next-nearest neighbor dots that are occupied.

6.3.3 Interaction Between Neighboring Devices

Since it occurs regardless of the state of any other quantum dot, the electron transport mechanism described in the previous section is not yet sufficient to implement universal computation. Obermayer *et al.* also identified and analyzed an interaction mechanism. That is, they identified a charge transport mechanism that depends on the state of nearby dots and can therefore, in principle, provide the basis for CA -like computation. The interaction they propose for this purpose is the Coulomb interaction between electrons localized in nearby dots, which shifts the energy levels slightly (typically $\Delta\omega/\omega \approx .01$) depending on how many neighboring dots are occupied.

The conditional transfer effect is depicted schematically in Fig. 6-8. Figures 6-8a–c show the energy levels of two charge-transfer devices lying next to each other.

The frequency $\omega(0,0)$ indicates the frequency needed to excite an electron in dot C to the transient conduction level when neither the nearest-neighbor dot, B, nor the second-nearest neighbor dot, A, is occupied.

Part (b) of the figure shows the relative shift of the valence levels of the righthand device for the cases of one nearest, zero next-nearest neighbor dots occupied, and part (c) shows the case of zero nearest, one next nearest dot occupied.

In order for a device to support universal computation, it must have two simple properties: *nonlinearity*, the property that its output is not merely the superposition of its inputs, and *nonmonotonicity*, the property that its output is not a strictly increasing function of its inputs[120]. In addition, it must be possible to connect the output of one device to the input of another. Since it can be shown that the conditional charge transfer mechanism is a nonlinear and nonmonotonic logical function on the states of the quantum dots, its ability to support universal computation is highly plausible.

However, to describe a computation-universal CA based on this mechanism requires more; it requires us to explicitly define both a geometrical arrangement of quantum dots and an illumination sequence; the sequence should specify the order and frequency of the optical pulses that induce precisely those charge transfers needed to simulate the evolution of a universal CA . The geometrical arrangement is given in sections 6.4 and 6.5.1; the illumination sequence is specified in section 6.5.2.

6.4 Planarization of Ballistic Computation

Unfortunately, even with the aid of the replication transformation the billiard ball rule of Fig. 6-3 appears to be incompatible with the single-electron transfer mechanism described in section 6.3. The problem stems from the difficulty of *crossing signals* in two dimensions, a difficulty that arises when we try to permit the diagonal propagation of isolated particles for all four possible incident directions (see row 2 of Fig. 6-3). Since there is only one valence level in each dot, if two electrons try to cross paths they will be attempting to occupy a single available valence state; therefore, the billiard ball CA shown in Fig. 6-3 cannot be used. We must "planarize" the rule. That is, we must modify it so that signals never need to cross, but without losing its computation-universality.

The original billiard ball model CA though developed independently, was closely related to the HPP (Hardy-dePazzis-Pomeau) lattice gas[64, 63], which has $\sigma_1 = \sigma_2$. As it turns out, we can planarize this model simply by starting from the TM (Toffoli-Margolus) lattice gas[126, 145] instead. The resulting billiard ball CA is still computation universal, but has $\sigma_1 \neq \sigma_2$ (as mentioned above, this poses no problem for physical implementations based on replica CA). In this variant, tokens propagate parallel to the x and y directions of the square lattice, rather than diagonally as in the HPP lattice gas and in the billiard ball model of Fig. 6-3). The remainder of the rule is constructed consistent with this requirement and with the action of the original billiard ball model. At a linear cost in both space and time efficiency, it has been shown that such variants can always be constructed[110, p. 91].

The construction consists of taking the transition table shown in Fig. 6-3 and rotating the updated clusters (righthand sides of the arrows) in the clockwise direction for the even partition (i.e, the clustering which holds on the even time-steps), and in the counter-clockwise direction for the odd clustering. The resulting evolution operators σ_1 and σ_2 are shown in Fig. 6-9.

We see that for the even partition, the evolution σ_2 consists of rotating the states in the clockwise direction for the majority of the cases. The two exceptions are a counter-clockwise evolution when a single cell in the cluster is occupied (second row), and *no* rotation at all when a diagonal is occupied. When the odd partition is in effect, all rotational directions are reversed: cluster configurations with two non-diagonal or three occupied cells rotate their occupied cells counter-clockwise, while the single occupied cell rotates clockwise. The diagonally occupied states again remain stationary.

It is important to note that for both σ_1 and σ_2 we can describe all three classes of behaviors: 'rotate counter-clockwise', 'rotate clockwise' and 'no change', solely in terms of the *number* of nearest and next-nearest neighbors occupied by a 1s bit.

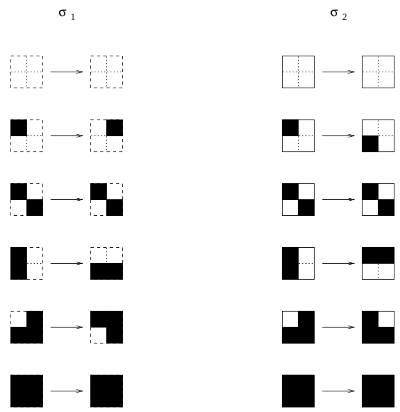

Figure 6-9: The complete transition table for the odd (left) and even (right) clusterings of the TM-variant of the Billiard Ball CA, up to rotations of the neighborhood cluster.

Consider σ_1. If an occupied cell has neither nearest nor next-nearest neighbors occupied, the bit is alone in its cluster: move counter-clockwise; if a cell has no nearest neighbors and one next-nearest neighbor, diagonal cells are occupied: do not move; if an occupied cell has any other combination of occupied nearest and next-nearest neighbors, its bit moves clockwise. It is easy to see that σ_2 has the same property.

This abstract computational property strongly resembles the nearest and next-nearest neighbor dependence of the Coulomb shift of energy levels described in section 6.2. We will take advantage of this resemblance by applying the replica transformation to this TM-variant of Fredkin's billiard ball model.

6.5 Construction of the Quantum Dot Automaton

6.5.1 Geometrical Arrangement

Now that the planarization constraint has been dealt with, the geometrical arrangement follows almost directly from the definition of replica CA . Since the PCA in Fig. 6-9 is a 2-phase rule on a square tiling, in order to derive the geometrical arrangement of quantum dots (shown in Fig. 6-10), we start with the replicated tiling at the bottom of Fig. 6-4. The unshaded solid polygons and narrow dotted rectangular regions separating them in Fig. 6-10 represent quantum dots and barriers respectively (as in Fig. 6-6a and Fig. 6-7a–c). The role of the large shaded octagons is described later. The principal difference between the replica CA of Fig. 6-4 and that of Fig. 6-10 is the addition of the small square regions that convert the cluster of four triangular quantum dots in Fig. 6-4 into an octagonal cluster of triangular and square quantum dots.

These intermediary square dots function as “latching cells” when transferring electrons during the updating of the cluster. They prevent undesired multiple transitions that would occur because the optically-induced transfer process does not take place at a perfectly well-defined instant. By the analysis in ref. [130], if neighboring dots are illuminated on resonance, charge transfer is guaranteed to have occurred with very high probability after time $\tau_s \ll \tau_d$. But there is nothing to prevent it from occurring as quickly as the shortest relaxation time τ_r. Without the intermediate latching cells to buffer the charge-transfer transitions, charges would often make several consecutive transfers within the allowed switching time time τ_s, consequently arriving at an unintended location. Unless we break the transition into two latched steps to prevent this, the intended computation of the cellular automaton rule would be irrevocably corrupted [6].

The function of the remaining unshaded regions is the same as in Fig. 6-4. The triangular regions correspond to the replicated, triangular cells of the tiling depicted at the bottom of Fig. 6-4. Triangular regions in the even rows of octagonal clusters (labelled by a σ_2 at the center of the cluster) correspond to the even replicas of cells in the progenitor PCA . The triangular regions in the rows labelled ‘σ_1’ correspond to the second complete replica of cells of the original, square cellular array, and are used to compute the new state of the automaton on the odd-numbered time-steps. The larger square cells represent additional quantum dots that correspond to the relinking cells. They provide a connecting stepping-stone to shuttle charge back and forth between the two triangular quantum dots that physically implement its even and odd replicas.

[6] Since the billiard ball model is logically reversible, it has no intrinsic fault-tolerance.

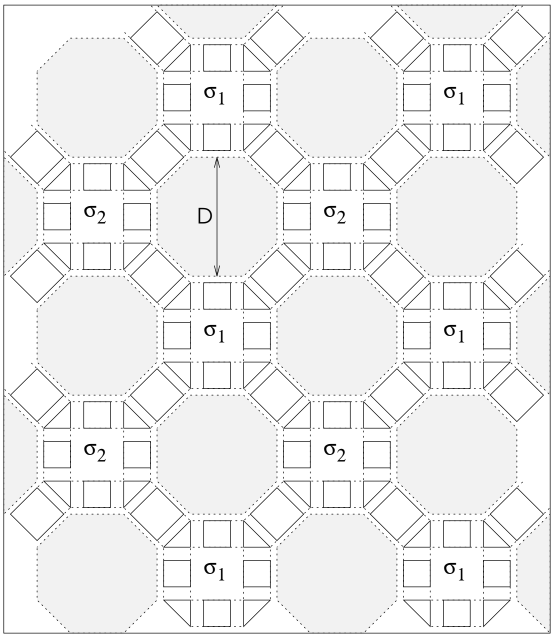

Figure 6-10: The geometrical layout of the proposed automaton. Three even rows and two odd rows, each containing two octagons, are shown.

The role of the remaining regions, the large shaded octagons, is more passive, but no less important: they maintain the logical isolation of the clusters by screening interactions that would otherwise cause the state of cells in one cluster to depend on the states of cells in other clusters. Since replica CA are a valid computational model only if each cluster evolves independently, it is essential to maintain the physical validity of this assumption by reducing the effects of inter-cluster interactions to negligible magnitudes.

In fact, whenever replica CA are used, the issue of screening must be considered. Replica CA can legitimately be applied to only those systems in which intercluster effects on the computational degrees of freedom can be effectively screened. We briefly describe how a satisfactory degree of screening between clusters might plausibly be achieved for the array of charge-transfer quantum dots interacting by Coulomb renormalization of the three energy levels in Fig. 6-8. In principle, the level shifts depend on the position of charges in every cluster; however, if the shaded region contains a free electron gas, all but nearest and second-nearest neighbors (i.e., the *intra*cluster neighbors) will be exponentially screened.

If the electron gas is deep enough that it is no longer effectively two-dimensional, we can get a rough idea of what density is required for effective screening. In the static long-wavelength limit, the full three-dimensional Lindhard dielectric function yields the same expression for the screening length λ as the elementary Thomas-Fermi result

$$\lambda_{TF} = \sqrt{\lim_{q\to 0}\epsilon(q, w = 0)} = \frac{1}{\sqrt{4\pi e^2 \left(\frac{\partial n(E)}{\partial E}\right)_{E_F}}} \tag{6.7}$$

Inverting, we find the required density of the electron gas as a function of screening length

$$n(\lambda_{TF}) = \frac{\pi}{3}\left(\frac{a_o}{(2\lambda_{TF})^2}\right)^3 \tag{6.8}$$

By definition, replica CA have the property that the closest possible approach between two charges not occupying the same cluster (i.e., the third-nearest neighbor distance) is D (Fig. 6-10). For $\lambda_{TF} \approx D/10 = 10$nm, the required density, $n(\lambda_{TF}) = 2.05 \times 10^9\text{cm}^{-3}$, is in the range of attainable values.

Since we can then neglect any level shifts caused by third-closest neighbors, the optical illumination frequency $\omega(N_1, N_2)$ required to induce a charge-transfer process can be written as a function of just two arguments, the number N_1 of nearest neighbors and the number N_2 of second-nearest neighbors.

6.5.2 Optically-induced Evolution

Finally, to complete the description of the quantum dot cellular automaton we need to describe the sequence of illumination frequencies that will induce the arrangement of quantum dots in Fig. 6-10 to exhibit the alternating (σ_1, σ_2) evolution of the replica CA billiard ball model in Fig. 6-9.

At the end of section 6.4 we noted that both the σ_1 and σ_2 parts of the replica CA billiard ball model evolution have a special property: their behavior depends only on the number of nearest and next-nearest neighbors. Since the Coulomb interaction responsible for the conditional charge-transfers also depends only on the distance between neighbors, we will see that something remarkable occurs: The lookup table can be broadcast by sequentially illuminating the array.

Each frequency in the sequence induces the appropriate transfer for one particular case (N_1, N_2) of the number of nearest and next-nearest neighbor dots occupied. (In practice, of course, it would be very difficult to provide illumination sufficiently stable and with sufficiently narrow bandwidth to keep the cases distinct.) Granted this, the illumination sequence can now be *data-blind*; that is, the illumination source does not need to have any knowledge of which quantum dots are actually occupied. It merely serves as an elaborate external clocking mechanism for the computation taking place in the array.

For the geometric arrangement of quantum dots in Fig. 6-10, the arguments (N_1, N_2) now denote the number of nearest and next-nearest neighbor cells of the same polygonal type (i.e., triangular, small square or large square) for which the valence level is occupied. Different polygonal types are never simultaneously occupied for times greater than τ_s.

As described in section 6.5.1, in order to prevent undesirable multiple hops, each transfer is broken into a latched, two-step process. In describing the illumination sequence, we denote the optical frequency that induces the second step of each two-step process by a bar over the frequency corresponding to the first step. It can be made distinct from every other frequency by imposing some static, but spatially varying conditions, such as dopant concentrations established at fabrication time[129]. The actual numerical values of the illuminating frequencies are not important, only the fact that distinct frequencies exist for each necessary operation is essential.

If the charges initially occupy only dots in the σ_1 clusters in the quantum-dot array depicted in Fig. 6-10, then we can verify that the σ_1 evolution of the computation-universal CA in Fig. 6-9 will be induced by irradiating the array with the following sequence, where $(n \rightarrow n+1)$ denotes the transition from frame n to $n+1$ in Fig. 6-11:

Frames $(1 \rightarrow 2)$

$\omega(0,0)$ — When a single electron (no nearest neighbors, no next-nearest neighbors) occupies any one of the four triangular dots of a σ_1 cluster, illumination with frequency $\omega(0,0)$ activates an electron transfer from the triangular region to the latching site in the clockwise direction.

Frames $(2 \rightarrow 3)$

$\bar{\omega}(0,0)$ — Illumination with this frequency activates transfers from the latching sites to the clockwise triangular region, thereby completing the transition in the second row of the first column (σ_1) of Fig. 6-9. At this point in the cycle, all σ_1 clusters of the automaton which began with a single occupied dot have now completed their σ_1 updating for this step. So long as the remainder of the sequence consists of sufficiently narrow-band illumination, all such clusters will be far enough off-resonance that they will not satisfy any of the transition conditions used in the remainder of this step, and so will remain stationary.

Frames $(3 \rightarrow 4)$

$\omega(1,0)$ and $\omega(1,1)$ — Illumination with $\omega(1,0)$, (one nearest neighbor, no next-nearest neighbors) activates the first hop (triangular to counter-clockwise latching site) of the two step process that implements the case shown in the fourth row and first column of Fig. 6-9. Transitions are induced in all σ_1 clusters occupied by exactly two electrons in non-diagonal dots. Similarly, $\omega(1,1)$ activates the first hop of the transfer corresponding to the evolution operator σ_1 in all clusters with three dots occupied (row 5 of Fig. 6-9).

Frames $(4 \rightarrow 5)$

$\bar{\omega}(1,0)$ and $\bar{\omega}(1,1)$ — Illumination with these frequencies activates the second and final charge-transfer hops, thereby completing the update of all σ_1 clusters that satisfy the fourth and fifth row cases of Fig. 6-9.

At this point, each cluster has undergone its appropriate update according to σ_1. Now, the data must be transferred to the locations where it will undergo an update by σ_2.

Frames $(5 \rightarrow 6)$

ω_{tr} — Illumination with ω_{tr} activates the transfer of charge from all triangular regions,

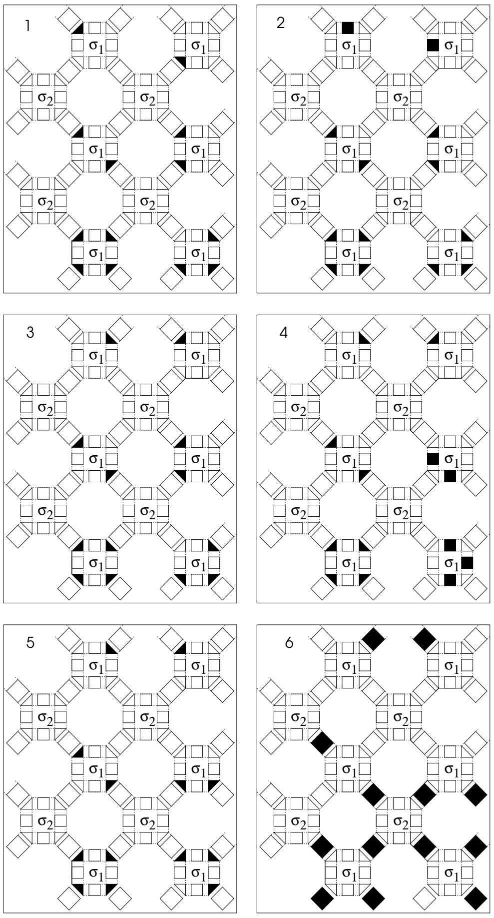

Figure 6-11: Time evolution of the automaton array.

regardless of their neighborhood environment, to the relinking sites (large squares) that will carry the data in each replica cell (triangular cell) to its replica in the corresponding σ_2 cluster.

Illumination with $\bar{\omega}_{tr}$ completes the process of transferring all results of the σ_1 evolution to the corresponding replica cell of the σ_2 clusters.

The entire σ_1 evolution (left half of Fig. 6-9) is now complete for every cluster in the entire array, and the resulting data are now gathered in clusters of the σ_2 partition, where a similar process occurs.

Repetition of this two-phase cycle of illumination, (each phase consisting of the above sequence of six illumination frequencies) induces the pattern of occupied dots to evolve according to the universal billiard ball cellular automaton of Fig. 6-9. If it were somehow possible to meticulously program the initial pattern of dot occupancies, any Boolean circuit could in principle be simulated (under the interpretation described in section 6.1.2). Hence, any computable function could be computed by the array of quantum dots.

6.6 Discussion

In this chapter, we have addressed the question, "What form must a cellular automaton take when, rather than being at liberty to define an arbitrary lookup table, we are constrained to make direct use of existing physical interactions between cells?" We have argued that the most obvious form, namely rote translation of the regular array of the abstract von Neumann CA into a regular array of device structures, suffers from serious drawbacks: for realistic interactions, one loses all ability to influence important properties of the dynamics, such as which cells interact (i.e., are neighbors), and when they interact. The resulting dynamical system has randomly fluctuating, long-range couplings between the cells. Consequently, the goal of attaining the dynamics of a cellular automaton becomes unreachable. This problem is what originally suggested the need to consider a new class of CA whose form is more amenable to physical realization, a class we have called few-body cellular automata.

Starting with lattice-gas cellular CA rather than von Neumann CA , we have defined shown how certain few-body cellular automata, "replica CA ", can be generated from lattice-gas cellular automata by replicating cells. Replica CA inherit the computational characteristics of lattice-gas CA , including their potential to perform universal computation. The most significant attribute of replica CA is that, for a large class of interactions between cells, they let us regain enough control to impose a genuinely CA -like dynamics on the array of interacting device structures.

The class of interactions to which replica CA apply can be characterized by two properties. First, the interactions between cells must possess a screening length D, which will determine the minimum separation between clusters (see Fig. 6-9). Screening lets us reduce the task of imposing a CA -like evolution on the array to one of imposing a CA -like evolution on a collection of non-interacting few-body systems, the clusters. Secondly, it must be possible to transport the information-bearing degrees of freedom between the replica sites, also separated by a distance of order D, without changing their logical state,

We have used the construction of a two-dimensional, computation-universal replica CA composed of idealized quantum-dot devices as an illustration of how replica CA might be applied to more realistic nanometer-scale devices. The substantial obstacles that separate such idealized constructions from genuinely feasible computers have already been extensively investigated by R. Landauer[97] and others [89, 85, 84, 87, 88, 86, 67]. Among the issues that any practical implementation must successfully address are error correction in the presence of various noise sources, tolerance to fabrication variation among cells, and power-dissipation limits. Nevertheless, the class of interactions to which replica CA can be applied has many members. Among these we may be fortunate enough to find one for which these obstacles can eventually be overcome, thereby enabling us to build a new generation of cellular automata machines at nanometer scales.

Chapter 7

Concluding Remarks

In this thesis, we have tried to understand computation as a physical process by noting the analogy between a computational model: a reversible N-input/N-output logic gate, and a physical model: the N-body $\widehat{S}$-matrix. In §2.1–2.2, we have worked out explicit physical limits that determine the class of physical interactions for which the notion of an N-input/N-output logic gate is physically valid.

For that class of interactions, we have defined few-body automata, a form of cellular automaton that takes into account a property of real physical interactions not captured by previous models of computation: their finite range. Using few-body automata, we were able to make some progress on one fundamental and one more practically-oriented problem of physical computation. The latter was described in Chapter 6, *Ballistic Computation in an Array of Quantum Dots*, where we used few-body automata to solve a problem posed by Obermayer, Teich and Mahler[130], the problem of describing a computation-universal cellular automaton based on Coulomb interactions between the quantum dot structures that they proposed.

The more fundamental result was presented in Chapter 5, *Parallel Quantum Computation*. Responding to Landauer's criticism that previous models of quantum computation entailed Hamiltonians so contrived as to be physically irrelevant, we showed that the few-body form leads to a model of quantum computation in which most of the contrived character of the Hamiltonian is removed. In particular, the part of this model that determines the computation rate is governed by the one-dimensional XY

model of Lieb, Shultz and Mattis, a model whose physical realism is attested to by the fact that it accurately predicts the heat-capacity and transverse electric susceptibility of the rare earth compounds praseodymium trichloride and praseodymium ethyl sulphate. Finally, we found exact solutions for the eigenstates and eigenvectors of the computation-rate operator. This enabled us, for the first time, to analyze informational properties—such as the dependence of the computation-rate on the excitation energy—using the established tools of quantum mechanics.

Appendix A

Computation in Coulomb Blockade Arrays

This section presents another illustration of how the concept of few-body automata gives us a framework for associating physical interactions with computation. A Fredkin gate is a nonlinear and nonmontonic element with three inputs and three outputs. Therefore, it can be regarded as a single cell of a computation-universal three-body automaton. Coulomb blockade effects govern the physical interactions of electron tunneling in linear arrays of tunnel junctions. Where two linear arrays cross, the conductivity of one junction depends on the state of a neighboring junction. This interaction can be used for computational purposes we impose appropriate geometric constraints on the arrangement of junctions at the intersection.

Likharev has already proposed a NOR gate whose operation depends on these effects. Since NOR is logically complete, this demonstrates that the Coulomb blockade effects can be used to compute. His NOR construction has several undesirable features: one of the two outputs—a leftover input really—must always be dumped, and the inputs are time multiplexed—in computing $f_{\mathrm{NOR}}(A, B)$, the input A must always arrive one clock before B.

The Fredkin gate construction is able to overcome some of these because its form—the few-body form—is more congruent with the form of the underlying physical effects—the "scattering" of charge in one linear array from that in another. In

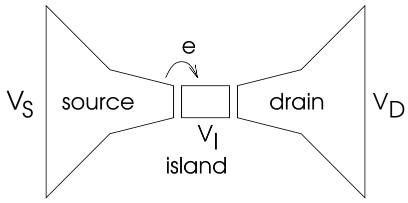

Figure A-1: Coulomb blockade occurs when the energy acquired by e in dropping from V_S to V_I is not sufficient to compensate for the energy $e^2/2C$ required to charge the small capacitance of the island.

the Fredkin gate implementation presented below, inputs to each logical gate arrive on the same clock and it is not necessary to dump any outputs to the substrate.

A.1 Coulomb Blockade Effects

A.1.1 Classical origin of Charging Effects

Consider a tunneling experiment where the tunneling from source to drain must take place through a small conducting island surrounded by insulator as in Fig. A-1. In such an arrangement, tunneling can be prohibited even if $V_S > V_I$. The condition for an electron to tunnel is not $-e(V_S - V_I) < 0$, as one might expect. Instead, an additional "charging energy" must be taken into account when the total capacitance C of the island becomes very small. An electron can tunnel from the source electrode to the island only when $\Delta E = -eV + e^2/2C < 0$. This effect was first observed almost thirty years ago—albeit indirectly, since the tunneling occurred through many islands in parallel— in granular metallic films (Zeller and Giaever[155]). The correct classical explanation for the observed reduction in tunneling through small grains was given at that time. During the resurgence of interest in the 1980s, this effect was named the "Coulomb blockade effect". And a tunnel junction is said to be Coulomb blockaded whenever the potential V across the junction satisfies $-eV + e^2/2C \geq 0$.

The Coulomb blockade effect can only be observed when the temperature T and junction resistance R fall in a narrow range. The development of cryogenic and

Technology	Junction Area (nm^2)	Junction Capacitance $10^{-18}F$	Characteristic Temperature $T_0(K)$	Time Scale $RC(ps)$
State-of-the-art[57]	30×30	30	30	3
Nanolithography limit	10×10	3	300	0.3
Molecular day dream	3×3	0.3	3000	0.03

Table A.1: Likharev's Single Electron Logic Family[5, 105]. In order to reduce the rate of logically erroneous tunneling events to a manageable level, the actual operating temperature is estimated[5] to be of order $T_0 \times 10^{-2}$.

nanofabrication technology has only recently made it practical to observe Coulomb blockade effects in single junctions. The dependence on T is easy to understand. If the thermal energy $k_B T$ exceeds the charging energy $e^2/2C$ responsible for the Coulomb blockade, then the blockade effect will not be observed because the tunneling electrons will be able to "steal" the necessary energy from the heat bath. A characteristic temperature[4] T_0 for observing the effect can be defined by

$$T_0 = e^2/2k_B C. \tag{A.1}$$

The dependence on junction resistance is almost as simple; it follows from the uncertainty principle and can be thought of as the lifetime energy broadening of states on the island. The energy of quantum fluctuations across a junction with resistance R and capacitance C is $\delta E = \hbar/RC$. Unless $\delta E \ll e^2/2C$, the picture of electron tunneling sequentially through a state localized on the island electrode is not quantum-mechanically valid. Instead, electrons tunnel by "borrowing" the required energy from quantum fluctuations, and the Coulomb blockade effect is not observed. So, in addition to the requirement $T \ll T_0$, we must have

$$R \gg 2\hbar C/e^2. \tag{A.2}$$

Likharev[105] has summarized the magnitudes of the relevant quantities for soft metal oxide junctions, where the capacitance per unit area and junction resistance are estimated to be $3 \times 10^{-6} F/cm^2$ and $100k\Omega$, respectively.

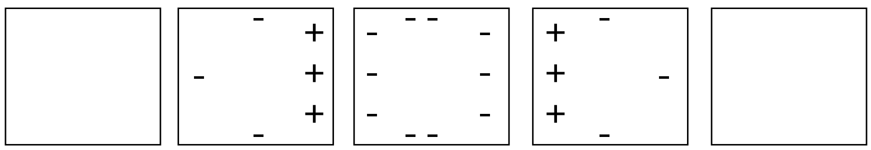

Figure A-2: Soliton in a linear array (adapted from [4]). The soliton extends over 3 neighboring junctions.

A.1.2 Spatial correlation: Wires

If we consider configurations consisting of a one-dimensional array of islands separated by thin, insulating barriers, the Coulomb blockade effect can cause the single-electron tunneling events to become spatially correlated. Consider a linear array of junctions biased from both ends in such a way that the potential drop across all junctions is below the tunneling threshold. If the potential across the array is raised, eventually the potential across one of the junctions, say the leftmost, will exceed the threshold $e/2C$ and a tunneling event will occur onto the first island. For appropriate values of the voltage bias and capacitance of the first island, this tunneling event can raise the potential drop across the second junction above its threshold, thereby inducing a second tunneling event, and so on until the tunneling electron exits the other end of the array. This correlated sequential tunneling can be utilized as a discrete wire for single-electron logical circuits.

The charge disturbance that actually propagates in the linear array is not completely localized on a single island. Rather, it is a charge soliton composed of a core containing the single electron along with the induced polarization of charge on nearby junctions (Fig. A-2). Bakhvalov et al. have shown that the charge of these solitons decays exponentially with a screening length of about $1/\cosh^{-1}(1 + C_0/2C)$ islands, where C is the junction capacitance and C_0 the stray capacitance of each island. Typical screening lengths are 3-5 islands[41]. The propagation of the charge solitons is dispersionless—that is, they maintain their shape—but not dissipationless. To move them along the one-dimensional array requires a voltage bias, the energy being supplied by the voltage sources.

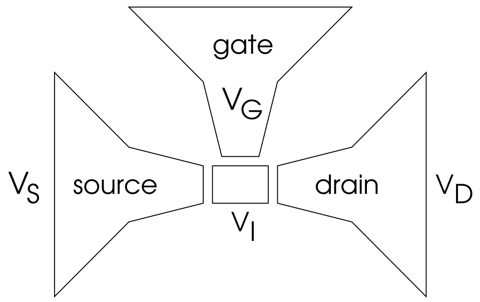

Figure A-3: Capacitively coupled single-electron transistor (CSET[4]).

A.1.3 Capacitively-coupled Single Electron Transistor (CSET): Logic Gates

Likharev[105] has pointed out that if an island is capacitively coupled to a gate electrode (in addition to source and drain electrodes, Fig. A-3), the effect of the Coulomb blockade can be used to produce a single-electron transistor (SET). If we let $V_S = V = -V_D$ and let V_G denote the gate voltage, it is straightforward to show that the potential of the island is controlled by the gate voltage,

$$V_I = \frac{V(C_{SI} - C_{DI}) + C_{GI}V_G}{C_{SI} + C_{DI} + C_{GI}}. \tag{A.3}$$

By varying the gate voltage V_G, it is possible to continuously vary the potential of the island, thereby controlling whether or not tunneling from source electrode to island is Coulomb blockaded or not. The expected behavior has been observed in the laboratory. Fulton and Dolan[57] were the first to observe this effect. They made three Al-Al junctions, each with area about 30nm $\times$ 30nm, on an Al bar of length 1 μm. Kastner[81] and co-workers have also made extensive investigations on single-electron transistors in GaAs systems, in which the island is formed by the electrostatic patterning of a two-dimensional electron gas.

A.1.4 Single electron OR

Likharev[105] also noted that if V_G can be produced by single electron charge on a neighboring junction, rather than a large external electrode, then it may be possible to produce single-electron logic circuits in complete analogy to MOSFET circuits.

Figure A-4 shows the arrangement of junctions proposed by Likharev[4] as a single-electron OR gate. The two input signals to the gate are represented by single-electron solitons propagating along the linear array of junctions labelled Data In. A logical value of 1 is represented by the presence of an electron soliton and a logical 0 by its absence. A regular array of clock pulses, also represented as a sequence of solitons, is incident from the left on the lower linear array. Each junction is initially biased so that an incident data soliton tunnels via the spatial-correlation effect along $1 \to 2 \to 3 \to 6$ where it is trapped in a memory cell formed by the capacitance C' and the junctions 6 and 7.

Voltage biases and junction capacitances are designed in such a way that the presence of the trapped soliton repels any subsequent soliton incident along Data In—shunting it along the alternate path $1 \to 2 \to 3 \to 4 \to 5$.

Finally, a clock pulse incident along $8 \to 9$ becomes lodged at island 9 where it acts as a gate electrode, opening the CSET transistor composed of junctions $(6, 7, 9)$. This frees the trapped soliton from the memory cell. The clock pulses is dissipated into the substrate through resistor R, while the freed soliton propagates to Output 2. Since a soliton will be present at Output 2 if and only if at least one of the two input signals on Data In contained a soliton, the logical value of Output 2 is easily seen to be that of an OR gate. The value of Output 1 is dumped.

A.2 Fredkin Gate via Coulomb Blockade

Using the techniques Likharev employed in the single-electron OR, we can avoid two of its objectionable features: dissipating the clock into the substrate and dumping the logical value of one of the outputs. To do so, we

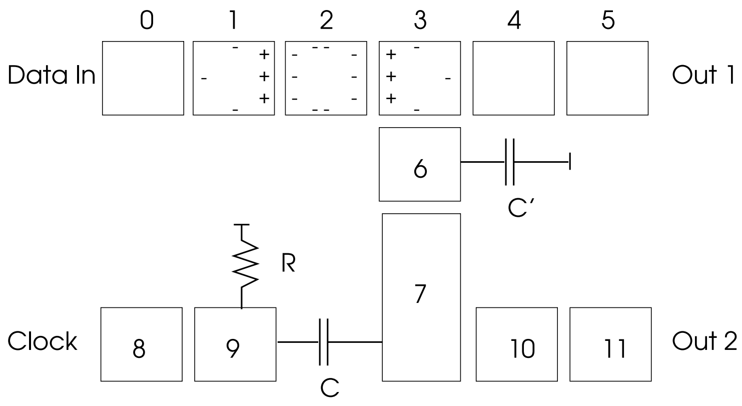

Figure A-4: Single-electron NOR gate (Likharev[4]).

A	B	F	A’
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Table A.2: Fredkin gate.

A.2.1 Fredkin Gate

The Fredkin gate[54] is a reversible logic gate that is as computationally powerful as the (irreversible) NAND gate. It has three inputs A, B and F, and three outputs A', B' and F'. The input F and the output F' are distinguished by the fact that they function as control, rather than data, signals. Its logical operation is quite simple: If $F = 1$, $A' = B$ and $B' = A$, otherwise the value of an output is the same as the value of the corresponding input (See Table A.2). Note that the gate is also bit conserving—the number of 1s that occur among the inputs is the same as the number that occur among the outputs. Bit conservation is an important property for Coulomb blockade logic, since the creation and destruction of charge solitons is energetically expensive.

Since the Fredkin gate is less well-known than the NAND gate, we will show

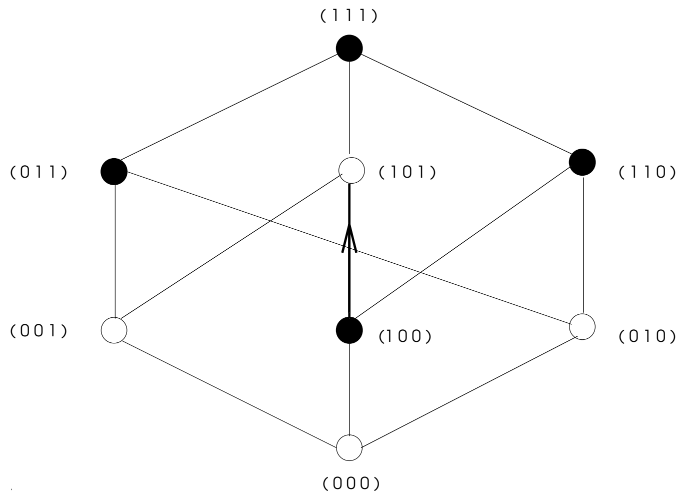

Figure A-5: Lattice representation of the output A' for a Fredkin gate.

that, like the NAND gate it is computation-universal, or, more accurately, it has the property of weak logical completeness. A gate G is (weakly) logically complete if, for any computable Boolean function f, a circuit that computes f can be constructed by interconnecting copies of G. It is known from classical switching theory[120] that a gate is logically complete if it is both nonlinear and nonmonotonic.

Using the diagrammatic notation developed in Chapter 3, it is straightforward to determine whether any specified gate posseses these properties. Figure A-5 shows the partially-ordered lattice representation of output A'. Open nodes indicate $A' = 0$, solid nodes indicate $A' = 1$. The nonmonotonic character of A' is evident from the transition, (shown as a directed bold line) as input (ABF) goes from (100) to (101). The nonlinear character of A' is apparent from the fact that input (011) yields nonzero input even though both of the inputs of which it is a superposition, namely (010) and (001) yield output zero.

Therefore, the Fredkin gate is computationally universal in the sense of weak completeness. This is also demonstrated more concretely in Figure A-6, where a composition of two Fredkin gates is shown to produce a NAND gate.

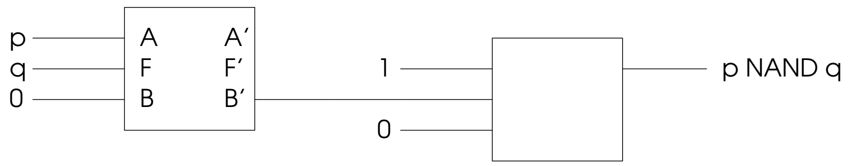

Figure A-6: Building a NAND gate from two Fredkin gates. The Fredkin gate is logically reversible, but the NAND is not. Notice the use of constants **0** and **1**.

A.2.2 Construction and Operation

The logical behavior of a Fredkin gate—-if $F = 1$, then interchange A and B to produce $A' = B$ and $B' = A$—is, in principle, as easy to implement as Likharev's single-electron OR. To do so, we only need to arrange the junctions in such as way that a soliton on the control line F acts to deflect incident solitons on the A and B lines so that they take alternate paths, crossing at a "junction-overpass", and then merging into the other line to complete their interchange. The mechanism for doing this can be adopted from Likharev's OR gate, where junction biases were arranged in such a way that the first incident soliton raised the potential enough to deflect the propagation of the following soliton (if any) onto an alternate track.

Figure A-7 shows one arrangement of junctions for achieving this. For simplicity, we only show the junctions necessary to deflect input A to B' under the influence of control signal F. A symmetric arrangement is required to implement the part of the Fredkin gate behavior that deflects the input B to output A'. Initially, junctions are biased in such a way that $A \to A'$ and $B \to B'$. When a soliton is present on F, an incident soliton arriving at island α is deflected to island β rather than to the linear array leading to output A'. The biasing of the overpass array γ propagates the soliton to output B'.

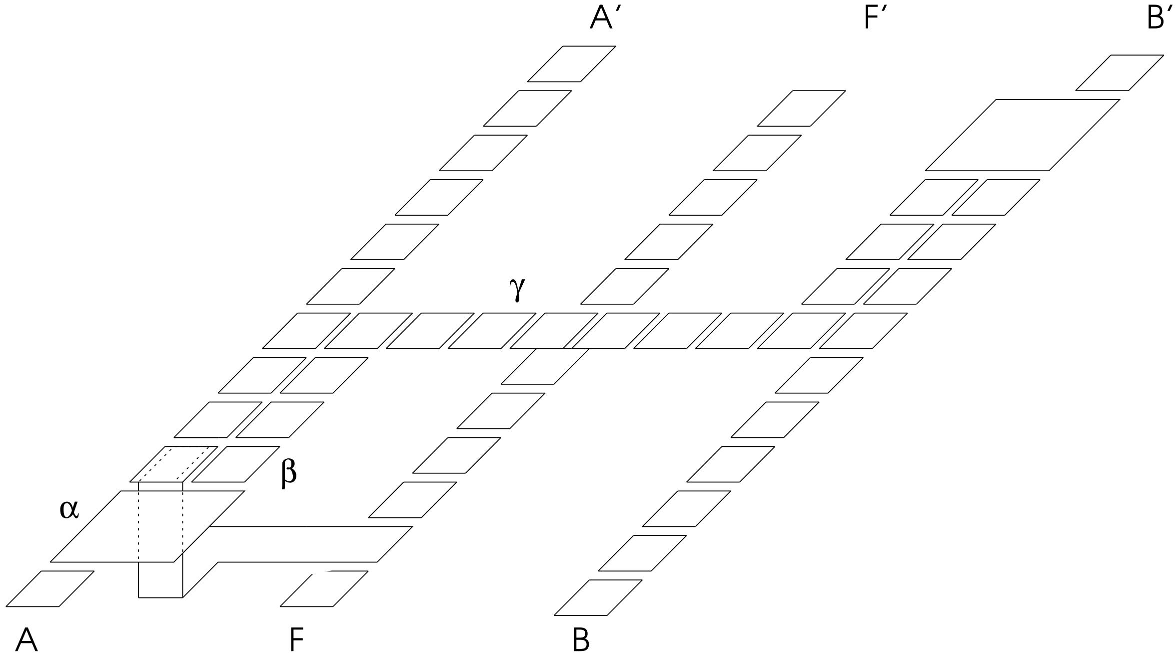

Figure A-7: Arrangement of junctions for one of the Fredkin gate crossovers. Each polygonal region is a conducting island, and each small gap between islands a tunneling junction.

Bibliography

[1] ABRAGAM, A. and BLEANY, B., *Electron Paramagnetic Resonance of Transition Ions*, Clarendon Press, Oxford (1970).

[2] ALBERT, J. and CULIK, KAREL, "A Simple Universal Cellular Automaton and its One-Way and Totalistic Versions", *Complex Systems* **1**, 1–16 (1987).

[3] ANDERSON, PHILLIP, "Limits on the Energy of the Antiferromagnetic Ground State", *Physical Review* **83**, 1260 (1951).

[4] AVERIN, D. V. and LIKHAREV, K. K., "Single Electronics: A Correlated Transfer of Single Electrons and Cooper Pairs in Systems of Small Tunnel Junctions", In *Mesoscopic Pheomena in Solids* (Altschuler, B. L., Lee, P. A., and Webb, R. A., eds.), 173 (1991).

[5] AVERIN, D. V. and LIKHAREV, K. K., "Possible Applications of Single Charge Tunneling", In *Single Charge Tunneling: Coulomb Blockade Phenomena in Nanostructure* (Grabert, H. and Devoret, M. H., eds.), 311–332 (1992).

[6] BANKS, E. R., *Information Processing and Transmission in Cellular Automata,*, PhD thesis, MIT (1971).

[7] BARGMANN, V., "On the connection between the phase shifts and scattering potential", *Review of Modern Physics* **21**, 488–493 (1949).

[8] BARGMANN, V., "Remarks on the determination of a central field of force from teh elastic scattering phase shifts", *Physical Review* **75**, 301–303 (1949).

[9] BATE, R. T., "The future of microstructure technology", *Superlattices and Microstructures* **2**, 9–11 (November 1986).

[10] BATE, R. T., "Nanoelectronics", *Solid State Technology* **32**, 101–108 (1989).

[11] BATE, R. T., FRAZIER, G. A., FRENSLEY, W. R., LEE, J. W., and REED, M. A., "Prospects for quantum integrated circuits", In *Quantum Well and Superlattice Physics* (Dohler, Gottfried H., ed.), 26–34 (1987).

[12] BEENAKKER, C. W. J. and VAN HOUTEN, H., *Quantum Transport in Semiconductors*, Academic Press (1991).

[13] BENIOFF, PAUL A., "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines", *Journal of Statistical Mechanics* **22**, 563–591 (1980).

[14] BENIOFF, PAUL A., "Quantum Mechanical Models of Turing Machines that Dissipate No Energy", *Physical Review Letters* **48**, 1581–1585 (1982).

[15] BENIOFF, PAUL A., "Quantum Mechanical Hamiltonian Models of Computers", *Annals of the New York Academy of Sciences* **480**, 475–486 (1986).

[16] BENNETT, CHARLES H., "Logical Reversibility of Computation", *IBM Journal of Research and Development* **17**, 525–532 (1973).

[17] BENNETT, CHARLES H., "Thermodynamics of Computation", *International Journal of Theoretical Physics* **21**, 905–940 (1982).

[18] BENNETT, CHARLES H., "Maxwell's Demon", *Scientific American* **257**, 108–116 (1987).

[19] BENNETT, CHARLES H., "Notes on the history of reversible computation", *IBM Journal of Research and Development* **32**, 16–23 (January 1988).

[20] BERLEKAMP, E. R., CONWAY, J. H., and GUY, R. K., *Winning Ways for Your Mathematical Plays*, Vol. 2, Academic Press (1982).

[21] Biafore, Michael, "Cellular Automata for Nanometer Scale Computation", *Physica D* **(to appear)** (Spring 1993).

[22] Biafore, Michael, "Universal computation in few-body automata", *Complex Systems* **7**, 221–239 (1993).

[23] Büttiker, M., "Four-Terminal Phase Coherent Conductance", *Physical Review Letters* **57**, 1761–1765 (1986).

[24] Bridgman, P. W., *The Logic of Modern Physics*, Macmillan (1927).

[25] Bridgman, P. W., "A physicist's second reaction to Mengenlehre", *Scripta Mathematica* **2**, 101–234 (1934).

[26] Burks, A. W., *Essays on Cellular Automata*, Univ. Illinois (1970).

[27] Capel, H. W., Dongen, E. J., and Siskens, Th. J., "On the Transverse Susceptibility χ_{xx} of the Anisotropic One-dimensional XY Model in the Presence of a Magnetic Field in the z Direction", *Physica* **76**, 445–485 (1974).

[28] Carter, F. L., "The Molecular Device Computer: Point of Departure for Large Scale Cellular Automata", *Physica* **10D**, 175–194 (1984).

[29] Case, K. M., "On discrete inverse scattering problems. II", *Journal of Mathematical Physics* **14**, 916 (1973).

[30] Case, K. M., "Orthogonal polynomials from the viewpoint of scattering theory", *Journal of Mathematical Physics* **15**, 2166–2174 (1973).

[31] Chadan, K. and Sabatier, P. C., *Inverse problems in quantum scattering theory*, Springer-Verlag (1977).

[32] Cheney, M., "Inverse scattering in dimension two", *Journal of Mathematical Physics* **25**, 94–107 (1984).

[33] Chew, Geoffrey Foucar, *S-matrix theory of strong interactions*, Benjamin (1962).

[34] Chiba, T., IEEE *Trans. on Comput.* **C-27**, 319 (1975).

[35] Codd, E. F., *Cellular Automata*, Academic Press (1968).

[36] Culvahouse, J. W. and Pfortmiller, L. G., "Spin-spin Interaction Constants for Nearest Neighbor Axial Pairs of Praseodymium Ions in Lanthanum Ethyl Sulfate and Lanthanum Trichloride", *Bulletin of the American Physical Society* **15**, 394 (1970).

[37] Culvahouse, J. W., Schinke, D. P., and Pfortmiller, L. G., "Spin-spin Interaction Constants from the Hyperfine Structure of Pairs of Coupled Ions", *Physical Review* **177**, 454 (1928).

[38] Culvahouse, J. W., Schinke, D. P., and Pfortmiller, L. G., *Journal of Applied Physics* **39**, 690 (1968).

[39] Datta, S., "Quantum Devices", *Superlattices and Microstructures* **6**, 83–93 (January 1989).

[40] Datta, S., Cahay, M., and McLennan, M., "Scatter-matrix approach to quantum transport", *Physical Review B* **36**, 5655–5659 (1987).

[41] Delsing, Per, *Single Electron Tunneling in Ultrasmall Tunnel Junctions*, PhD thesis, Chalmers University of Technology, Göteborg (1990).

[42] Dennard, R. H., "Physical Limits to VLSI Technology Using Silicon MOSFETS", *Physica B* **117**, 39–43 (1983).

[43] Dennard, R. H., Gaensslen, F. H., Yu, H., Rideout, V. L., and Bossons, E., IEEE *J. Solid-State Circuits* **SC-9**, 256 (1974).

[44] Deutsch, David, "Quantum theory, the Church-Turing principle and the universal quantum computer", *Proceedings of the Royal Academy of London A* **400**, 97–117 (1985).

[45] Donath, W. E., "Placement and Average Interconnection Lengths of Computer Logic", IEEE *Trans. on Circ. and Sys.* **CAS-26**, 272–276 (1979).

[46] Doolen, G. D., *Lattice Gas Methods for PDE's*, North-Holland, Amsterdam (1989).

[47] Faddeev, L. D., *Mathematical Aspects of the Three-body Problem in Quantum Scattering Theory*, Israel Program for Scientific Translations (1965).

[48] Ferry, D. K., "Two-Dimensional Automata in VLSI", In *Submicron Integrated Circuits* (Watts, R. K., ed.), 377–413 (1988).

[49] Ferry, D. K., Akers, L. A., and Greeneich, E. W., *Ultra Large Scale Integrated Microelectronics*, Prentice-Hall (1988).

[50] Ferry, D. K. and Porod, W., "Interconnections and Architecture for Ensembles of Microstructures", *Superlattices and Microstructures* **2**, 41 (1986).

[51] Feynman, Richard P., "Quantum Mechanical Computers", *Foundations of Physics* **16**, 507 (1986).

[52] Folinsbee, J. T., Harrison, J. P., McColl, D. B., and Taylor, D. R., "Specific heat, electric susceptibility and thermal conductivity of praseodymium ethyl sulphate: A one-dimensional electric dipolar XY system", *Journal of Physics C* **10**, 743–759 (1977).

[53] Frazier, G., Interconnections and Architecture for Ensembles of Microstructures, Technical Report 4769644, U.S. Patent, (1988).

[54] Fredkin, E. and Toffoli, T., "Conservative Logic", *International Journal of Theoretical Physics* **21**, 219–253 (1986).

[55] Fredkin, Edward and Toffoli, Tommaso, "Conservative Logic", *Int. J. Theor. Phys.* **21**, 129–253 (1982).

[56] Frisch, U., Hasslacher, B., and Pomeau, Yves, "Lattice-Gas Automata for the Navier-Stokes Equation", *Physical Review Letters* **56**, 1505–1508 (1986).

[57] Fulton, T. A. and Dolan, G. J., "Observation of Single-Electron Charging Effects in Small Tunnel Junctions", *Physical Review Letters* **59**, 109 (1987).

[58] GEL'FAND, I. M. and LEVITAN, B. M., "On the determination of a differential equation by its spectral function", *Doklady Akadamia Nauk USSR* **77**, 557–560 (1951).

[59] GIBBONS, A., *Efficient Parallel Algorithms*, Cambridge University, Cambridge (1987).

[60] GLUSHKOV, V. M., *Introduction to Cybernetics*, Academic Press (1963).

[61] GRUNBAUM, BRANKO and SHEPARD, G. C., *Tilings and Patterns*, Freeman (1987).

[62] GUSTAFSON, JOHN L., "Reevaluating Amdahl's Law", *Communications of the ACM (Association for Computing Machinery)* **31**(5), 532–533 (May 1988).

[63] HARDY, J., PAZZIS, O. DE, and POMEAU, YVES, "Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions", *Physical Review A* **13**, 1949–1960 (1976).

[64] HARDY, J., POMEAU, YVES, and PAZZIS, O. DE, "Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions", *Journal of Mathematical Physics* **14**, 1746–1759 (1973).

[65] HARRISON, J. P., HESSLER, JAN P., and TAYLOR, D. R., "One- and three-dimensional antiferromagnetic ordering in $PrCl_3$", *Physical Review B* **14**, 2979–2982 (1976).

[66] HART, P. A. H., HOF, T. VAN'T, and KLAASEN, F. M., IEEE *J. Solid-State Circuits* **SC-14**, 343 (1979).

[67] HECHT, J., "Physical Limits of Computing", *Computers in Physics* 34 (July 1989).

[68] HEILMEIER, G. H., Microelectronics: End of the Beginning or Beginning of the End?, In *Proceedings*, 2–5, IEDM, (1984).

[69] Heisenberg, W., "Die beobachtbaren Grössen in der Theorie der Elementarteilchen III.", *Z. Phys.* **123**, 93 (1944).

[70] Hepp, K., "On the Quantum Mechanical N-body Problem", *Helvetica Physica Acta* **42**, 425–458 (1969).

[71] Hirsh, J. E., Scalapino, D. J., Sugar, R. L., and Blankenbecler, R., "Efficient Monte Carlo Procedure for Systems with Fermions", *Physical Review Letters* **47**, 1628–1631 (1981).

[72] Hirsh, J. E., Scalapino, D. J., Sugar, R. L., and Blankenbecler, R., "Monte Carlo simulations of one-dimensional fermion systems", *Physical Review B* **26**, 5033–5055 (1982).

[73] Hopcroft, J. E. and Ullman, J. D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (1979).

[74] Iagolnitzer, D., *The* S *matrix*, North-Holland (1978).

[75] III, A. R. Smith, "Cellular Automata Complexity Trade-Offs", *Information and Control* **18**, 466–482 (1971).

[76] III, A. R. Smith, "Simple computation-universal cellular spaces", *Journal of the ACM* **18**, 339–353 (1971).

[77] Itzykson, C. and Drouffe, J. M., *Statistical Field Theory*, Cambridge University, Cambridge (1989).

[78] Jackson, J. D., *Classical Electrodynamics*, Wiley, 2 edition (1975).

[79] Jordan, P. and Wigner, E. P., "Über das Paulische Äquivalenzverbot", *Z. Phys.* **47**, 631–651 (1928).

[80] Kac, M., "Can one hear the shape of a drum?", *American Math Monthly* **73**, 1–23 (1966).

[81] KASTNER, M., "The single-electron transistor", *Reviews of Modern Physics* **64**, 849–858 (1992).

[82] KATSURA, S., "Statistical Mechanics of the Anisotropic Linear Heisenberg Model", *Physical Review* **127**, 1508 (1962).

[83] KAUFMAN, B., "Crystal Statistics II Partition Function Evaluated by Spinor Analysis", *Physical Review* **76**, 1232 (1949).

[84] KEYES, R. W., "Physical Problems and Limits in Computer Logic", *IEEE Spectrum* 36 (May 1969).

[85] KEYES, R. W., "Power Dissipation in Information Processing", *Science* **168**, 796 (May 1970).

[86] KEYES, R. W., "The evolution of digital electronics toward VLSI", *IEEE Transactions on Electron Devices* **ED-26**, 271–279 (April 1979).

[87] KEYES, R. W., "Fundamental Limits in Digital Information Processing", *Proc. IEEE* **69**, 267 (Feb 1981).

[88] KEYES, R. W., "Miniaturization of electronics and its limits", *IBM Journal of Research and Development* **32**, 24 (January 1988).

[89] KEYES, R. W., "Physics of Digital Devices", *Reviews of Modern Physics* **61**, 279 (1989).

[90] KIRK, W. P. and REED, M. A., *Nanostructures and Mesoscopic Systems*, Academic Press (1991).

[91] KOGUT, J., "Lattice gauge theory and spin systems", *Reviews of Modern Physics* **51**, 659–713 (1979).

[92] KRISHNAMURTHY, E. V., *Parallel Processing: Principles and Practice*, Addison-Wesley, Sydney (1989).

[93] LANDAUER, R., "Irreversibility and Heat Generation in the Computing Process", *IBM Journal of Research and Development* **5**, 183–191 (July 1961).

[94] LANDAUER, R., "Wanted: a physically possible theory of physics", *IEEE Spectrum* **4**, 105–109 (1967).

[95] LANDAUER, R., "Uncertainty Principle and Minimal Energy Dissipation n the Computer", *International Journal of Theoretical Physics* **21**, 283–297 (1984).

[96] LANDAUER, R., "Computation and Physics: Wheeler's Meaning Circuit?", *Foundations of Physics* **16**, 551–564 (1986).

[97] LANDAUER, R., "Can We Switch by Control of Quantum Mechanical Transmission?", *Physics Today* **32**, 119 (October 1989).

[98] LANDAUER, R., "Information is Physical", *Physics Today* **41**, 23–29 (May 1991).

[99] LANDAUER, R. and BÜTTIKER, M., "Drift and Diffusion in Reversible Computation theory and spin systems", *Physica Scripta* **T9**, 155–164 (1984).

[100] LANDMAN, B. S. and RUSSO, R. L., IEEE *Trans. on Comput.* **C-20**, 1469 (1970).

[101] LEIGHTON, T., *Introduction to Parallel Algorithms and Architectures*, Moran Kaufmann, New York (1992).

[102] LEVINSON, N., "Determination of the potential from the asymptotic phase", *Physical Review* **75**, 1445 (1949).

[103] LEWIS, H. R. and PAPADIMITRIOU, C. H., *Elements of the Theory of Computation*, Prentice-Hall (1981).

[104] LIEB, E. H., SHULTZ, T., and MATTIS, D. C., "Two Soluble Models of an Antiferromagnetic Chain", *Annals of Physics (N.Y)* **16**, 407–466 (1961).

[105] Likharev, K. K., "Correlated discrete transfer of single electrons in ultrasmall tunnel junctions", *IBM Journal of Research and Development* **32**, 144–158 (January 1988).

[106] Lindgren, K. and Nordahl, M., "Universal Computation in Simple One-Dimensional Cellular Automata", *Complex Systems* **4**, 299–318 (1990).

[107] Lubkin, E., "Keeping the Entropy of Measurement: Szilard Revisited", *International Journal of Theoretical Physics* **26**, 523–535 (1987).

[108] Mahler, G. and Obermayer, K., "Towards the Quantum Computer: Information Processing with Single Electrons", In *Computational Systems–Natural and Artificial* (Haken, H., ed.), (1987).

[109] Marchenko, V. A., "The construction of the potential energy from the phases of the scattered waves", *Doklady Akadamia Nauk USSR* **104**, 695–698 (1955).

[110] Margolus, Norm, "Physics-like Models of Computation", *Physica* **10D**, 81–95 (1987).

[111] Margolus, Norman, "Physics-like models of computation", *Physica D* **10**, 81–95 (1984).

[112] Margolus, Norman, "Quantum Computation", *Annals of the New York Academy of Sciences* **480**, 487–497 (1986).

[113] Margolus, Norman, *Physics and Computation*, PhD thesis, MIT (1987).

[114] Margolus, Norman, *Physics and Computation*, PhD thesis, MIT (1987).

[115] Margolus, Norman, "Parallel Quantum Computation", In *Complexity, Entropy and the Physics of Information* (Zurek, W., ed.), 301–318 (1990).

[116] Margolus, Norman and Toffoli, Tommaso, "Cellular Automata Machines", In *Lattice Gas Methods for Partial Differential Equations* (Doolen, G., ed.), 219—248 (1990).

[117] Mattis, D. C., *Theory of Magnetism*, Vol. 1, Springer, Berlin (1985).

[118] Minsky, Marvin, *Computation: Finite and Infinite Machines*, Prentice-Hall, New York (1967).

[119] Morton, J. A., "From Physics to Function", *IEEE Spectrum* **1**, 62–66 (September 1963).

[120] Mukhopadhyay, Amar, *Recent Developments in Switching Theory*, Academic Press (1971).

[121] Newton, R. G., *Scattering Theory of Waves and Particles*, McGraw-Hill, New York (1966).

[122] Newton, R. G., *Journal of Mathematical Physics* **21**, 1698 (1980).

[123] Newton, R. G., *Journal of Mathematical Physics* **22**, 2191 (1981).

[124] Newton, R. G., *Journal of Mathematical Physics* **23**, 594 (1982).

[125] Newton, R. G., *Inverse Schrödinger Scattering in Three Dimensions*, Springer-Verlag, Berlin (1989).

[126] Norman Margolus, , Tommaso Toffoli, , and Gérard Vichniac, , "Cellular-Automata Supercomputers for Fluid-Dynamics Modeling", *Physical Review Letters* **56**, 1694–1696 (1986).

[127] Nussbaum, Daniel and Agarwal, Anant, "Scalability of Parallel Machines", *Communications of the ACM (Association for Computing Machinery)* **34**(3), 57–61 (March 1991).

[128] Obermayer, K., Mahler, G., and Haken, H., "Multistable Quantum Systems: Information Processing at Microscopic Levels", *Physical Review Letters.* **58**, 1792 (1987).

[129] Obermayer, K., Teich, W., and Mahler, G., "II. Effective few-particle dynamics", *Physical Review B* **37**, 8121–8137 (1988).

[130] Obermayer, K., Teich, W., and Mahler, G., "Structural basis of multistationary quantum systems. I. Effective single particle dynamics", *Physical Review B* **37**, 8111–8121 (1988).

[131] Peres, Asher, "Reversible logic and quantum computers", *Physical Review A* **32**, 3266–3276 (1985).

[132] Post, E. L., "Introduction to a general theory of elementary propositions", *Amer. J. Math* **43**, 163–185 (1921).

[133] Press, William H., Flannery, Brian P., Teukolsky, Saul A., and Vetterling, William T., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University, Cambridge (1988).

[134] Prince, J. L., *Very Large Scale Integration*, Springer (1980).

[135] Reed, M. A. and Kirk, W. P., *Nanostructure Physics and Fabrication*, Academic Press (1989).

[136] Simon, Barry, *Quantum mechanics for Hamiltonians defined as quadratic forms.*, Princeton, Princeton (1971).

[137] Stone, A. D. and Szafer, A., "What is measured when you measure the resistance?—The Landauer formula revisited", *IBM Journal of Research and Development* **32**, 384–412 (May 1988).

[138] Striny, K. M., "Assembly Techniques and Packaging of VLSI Devices", In *VLSI Technology* (Sze, S. M., ed.), chapter 13, 26–34 (1988).

[139] Taylor, J. R., *Scattering Theory*, Academic Press (1972).

[140] Teich, W. and Mahler, G., "Optically controlled multistability in nanostructured semiconductors", *Physica Scripta* (1989).

[141] Thaker, H. B., Quigg, C., and Rosner, J. L., *Physics Letters B* **74**, 350 (1978).

[142] Toffoli, Tommaso, *Cellular Automata Mechanics*, PhD thesis, University of Michigan (1977).

[143] Toffoli, Tommaso, "Integration of the phase-difference relations in asynchronous sequential networks", In *Automata, Languages, and Programming* (Ausiello, G. and öhm, C. B, eds.), 457–463 (1978).

[144] Toffoli, Tommaso, "CAM: a high-performance cellular-automaton machine", *Physica* **10 D**, 195–205 (1984).

[145] Toffoli, Tommaso and Margolus, Norman, *Cellular Automata Machines*, MIT Press, Cambridge (1987).

[146] Ulam, S., "Random Processes and Transformations", *Proc. Int. Congr. Mathem.* **2**, 264–275 (1952).

[147] von Neumann, J., *Theory of Self-Reproducing Cellular Automata*, Univ. Illinois (1966).

[148] Weisbuch, C. and Vinter, B., *Quantum Semiconductor Structures: Fundamentals and Applications*, Academic Press (1991).

[149] Wheeler, J. A., "On the mathematical description of light nuclei by the method of resonating group structure", *Physical Review* **52**, 1107 (1937).

[150] Williams, F. I. B., "Paraelectric resonance of praseodymium in yttrium ethyl sulphate", *Proceedings of the Physical Society* **91**, 111–123 (1967).

[151] Wolfram, Stephen, "Universality and Complexity in Cellular Automata", *Physica* **10 D**, 1–35 (1984).

[152] Wordeman, M. R., Schweighart, A. M., and Dennard, R. H., IEEE *Transactions on Electron Devices* **ED-31**, 452 (1984).

[153] Zakhar'ev, B. N., Mel'nikov, V. N., Rudyak, B. V., and Suz'ko, A. A., "Inverse scattering problem (finite-difference approach)", *Soviet Journal of Particles and Nuclei* **8**, 120–137 (1977).

[154] Zakhar'ev, B. N. and Suz'ko, A. A., *Direct and inverse problems : potentials in quantum scattering*, Springer-Verlag, Berlin (1990).

[155] Zeller, H. R. and Giaever, I., "Tunneling, Zero-Bias Anomalies, and Small Superconductors", *Physical Review* **182**, 789 (1969).

[156] Zurek, W., "Quantum Mechanical Models of Turing Machines that Dissipate No Energy", *Physical Review Letters* **48**, 1581–1585 (1982).

[157] Zurek, W., "Reversibility and Stability of Information Processing Systems", *Physical Review Letters* **53**, 391–395 (1984).

[158] Zuse, K., *Rechnender Raum*, Vieweg, Braunschweig (1969).