CDMTCS
Research
Report
Series

Unconventional Models of Computation'98: Posters

Peter Hertling (ed.)
Department of Computer Science
University of Auckland

CDMTCS-075
January 1998

Centre for Discrete Mathematics and
Theoretical Computer Science

Abstracts of the Poster Session Contributions

to the

First International Conference on Unconventional Models of Computation

Centre for Discrete Mathematics and
Theoretical Computer Science, NZ

and

Santa Fe Institute, USA

5 – 9 January 1998, Auckland, New Zealand

Edited by Peter Hertling, University of Auckland

Preface

The First International Conference on **Unconventional Models of Computation, UMC'98**, organized by the Centre for Discrete Mathematics and Theoretical Computer Science, in cooperation with the Santa Fe Institute, was held at the University of Auckland from January 5–9, 1998. The proceedings of UMC'98 have appeared in the DMTCS Series of Springer–Verlag, Singapore.
This CDMTCS Research Report contains the abstracts or extended abstracts of the poster session contributions to UMC'98. They cover engineering, physical, mathematical, and philosophical aspects of a broad range of unconventional models of computation: from evolutionary computation, quantum computation, DNA–computing, molecular computing, new developments in probabilistic computation, and cellular automata to the question of electronic simulation of the human mind. Thanks to all participants for their interesting contributions.

Auckland, January 1998

P. Hertling
Poster Session Chair

Contents

Probabilistic Computation of Complex Systems

Ioannis Antoniou

International Solvay Institutes, ULB CP 231, Campus Plaine, Blvd du Triomphe, 1050, Brussels, Belgium

Email: antoniou@solvayins.ulb.ac.be

The Brussels-Austin groups directed by Ilya Prigogine have recently developed a probabilistic approach to complex systems based on a generalised harmonic analysis of the evolution operators. This approach results in a decomposition of the systems in terms of resonances and degrees of correlations. The dual decomposition consists of components corresponding to eigenprojections of the Time Operator. These decompositions which are meaningful in suitable Rigged Hilbert Spaces are intrinsically probabilistic and irreversible and allow for probabilistic prediction and control of complex systems beyond the horizons of predictability and controllability. The resulting probabilistic inference is based on a new kind of Fuzzy Logic which is intrinsic to Chaotic Dynamics and expresses the probabilistic assessment of uncertainty.

Aspects of Reversible Computation: Radius 1/2 Reversible CA

Extended Abstract

Tim Boykett

Time's Up, Industriezeile 33b, A-4020 Linz, Austria

and

Mathematatik, Uni Linz, A-4040 Linz, Austria

Email: `tim@bruckner.stoch.uni-linz.ac.at`

Abstract

We compare the usefulness of the paradigms of radius 1 and radius 1/2 reversible cellular automata. Although the two techniques are equivalent, it is seen that the radius 1/2 paradigm leads to a clearer theory, a combinatorial model with good explicit constructions and possibly to a better intuition.

1 Introduction

Cellular Automata theory is, in general, plagued by varying definitions and assumptions, overlaid with a network of equivalences between various points of view, paradigms and models. The current paper addresses two differing but equivalent points of view of cellular automata, concentrating upon the case of reversible cellular automata. The main body of results in Section 3 comes from the author's thesis [2], the results in Section 4 are an attempt to use similar tools. The way in which these tools are unusable in the radius 1 paradigm used in Section 4 is then analysed and presented as evidence that the paradigm introduced by Pedersen in [15] but dating back to Hedlund [7] and used by the author in [1, 2] and others, e.g. Korec in [10], has certain strengths and advantages.

We proceed first by looking at some elementary cellular automata theory, then looking at the two paradigms more closely. In particular we formulate reversibility conditions for both the radius 1/2 and radius 1 paradigms.

Looking at the radius 1/2 paradigm as an algebraic structure, some elementary algebraic manipulation leads to an analysis of the pre–image partition of the local map as a simply defined combinatorial structure. A reverse construction demonstrates that the combinatorial structure and the algebraic structure are equivalent. Further models of the combinatorial structure in matrix and graph theory demonstrate the range of models for the algebra, and also lead to certain tools that can be applied to efficiently construct and analyse reversible cellular automata.

Attempting similar manipulations with the radius 1 paradigm fails. A structure that is related to the combinatorial structure of the previous section is defined and analysed and shown to be useful in constructing examples of reversible radius 1 cellular automata. Looking at the combinatorial structure more closely, we see that the models in matrix and graph theory for the earlier combinatorial structure do not carry across to the new structure, leaving us without the tools and intuitions that could help in the analysis and construction of such examples.

In the final section we compare the two paradigms and introduce some methods for construction of examples, in particular looking at the use of orderly algorithms [18] that take advantage of combinatorial structure to improve the efficiency of searching.

This extended abstract is a reduced form of [3], where more details and proofs may be found.

2 The Paradigms

2.1 Cellular Automata in Brief

In [15], J. Pedersen introduced the idea of using radius $1/2$ cellular automata as a method of algebraising the theory of cellular automata. In general, he shows that every cellular automata rule can be expressed as a binary operation, leading to the intuitive use of algebraic techniques for the analysis and investigation of cellular automata properties.

In general, radius 1 cellular automata have neighbourhood $E = \{-1, 0, 1\}$ while radius $1/2$ cellular automata have neighbourhood $E = \{-1, 0\}$ or $E = \{0, 1\}$. Radius $1/2$ cellular automata can obviously be embedded in radius 1 cellular automata. Using a "chunking" construction, any cellular automaton can be embedded in one of lower radius, down to radius $1/2$. In particular if we have a cellular automaton of radius 1, with state set S and local map f, we can define a new cellular automaton with state set $S' = S^2$ and local map

$$f'(x, y) = (f(x_1, x_2, y_1), f(x_2, y_1, y_2)) \tag{1}$$

2.2 Equations of Reversibility

If a radius 1 cellular automaton has a reverse, then we can, without loss of generality, restrict ourselves to the case where f has an inverse of radius 1. Let f be the local map and $\bar{f}$ be its inverse, then the reversibility condition can be written

$$f(\bar{f}(a, b, c), \bar{f}(b, c, d), \bar{f}(c, d, e)) = c \tag{2}$$

$$\bar{f}(f(a, b, c), f(b, c, d), f(c, d, e)) = c \tag{3}$$

for all $a, b, c, d, e \in S$.

For the radius $1/2$ case, with the neighbourhood of f being $\{-1, 0\}$ and the neighbourhood of $\bar{f}$ being $\{0, 1\}$, we can define

$$f(\bar{f}(a, b), \bar{f}(b, c)) = b \tag{4}$$

$$\bar{f}(f(a, b), f(b, c)) = b \tag{5}$$

for all $a, b, c \in S$.

We can treat both of these structures from a general algebra point of view, as we have varieties with signatures $(3,3)$ and $(2,2)$. One of the most important features of Pedersen's note was that standard algebraic terminology can be used to discuss cellular automata, and this approach has met with some success [5, 13, 14, 4].

For the remainder of this paper I want to look at the two constructions we have here and to show that although the two are equivalent, that the radius $1/2$ is vastly more amenable to analysis. This amenability is by no means restricted to an algebraic viewpoint, for instance in [8] D. Hillman uses traditional cellular automata theory to obtain some similar results to those presented below.

3 The Radius $1/2$ Paradigm

3.1 Preimage Partition

The radius $1/2$ case has the advantage of dealing with binary functions. Such functions are most commonly encountered in algebra written as operations. Thus we will consider the following algebra:

Definition 1 *A* Semicentral Bigroupoid *is a* $(2,2)$*-algebra* $(S, \bullet, \circ)$ *satisfying the following axioms:*

$$(a \bullet b) \circ (b \bullet c) = b \tag{6}$$
$$(a \circ b) \bullet (b \circ c) = b \tag{7}$$

The name comes from the fact that this algebra is a generalisation of a *central groupoid* [9, 6]. Often we will omit the $\bullet$ and represent this operation by juxtaposition.

The following analysis transforms a semicentralbigroupoid into a combinatorial structure.

Definition 2 *For any* $x \in S$ *define*

$$\mu_x : S^2 \rightarrow S^2 \tag{8}$$
$$(a, b) \mapsto (ax, xb) \tag{9}$$

then

$$\rho : S \rightarrow \mathcal{P}(S^2) \tag{10}$$
$$x \mapsto \mu_x(S^2) \tag{11}$$

Where $\mathcal{P}(X)$ denotes the power set of X.

Lemma 1 $\rho(S) = \{\rho(s) | s \in S\}$ *is a partition of* S^2. *For every* $x \in S$ *there exist* $A, B \subseteq S$ *such that*

- $\rho(x) = A \times B$.
- $|A \cap B| = 1$.

- $|A \circ B| = 1$, $B \circ A = S$.

Corollary 2 *If $a \circ b = c \circ d = x$, then $a \circ d = c \circ b = x$.*

This naturally applies by analogous arguments to the $\bullet$ operation, i.e. if $a \bullet b = c \bullet d = x$, then $a \bullet d = c \bullet b = x$.

Let $R_x^\circ = \{(a,b)|a \circ b = x\}$ for some $x \in S$. This is the set of preimages of x in $S \times S$ under $\circ$. $\mathcal{R}^\circ = \{R_x^\circ | x \in S\}$ is of interest. By the previous Lemma, it can be alternatively defined as

$$\mathcal{R}^\circ = \{R_x^\circ = (Sx, xS)|x \in S\} \tag{12}$$

That is, the pre–images of each element of S are subsets of $S \times S$ defined as cross products, I refer to these as *rectangles* owing to this simple structure. Note that we can equivalently define

$$\mathcal{R}^\bullet = \{R_x^\bullet = \{(a,b)|a \bullet b = x\}|x \in S\}, \tag{13}$$

which will have similar structure by the duality of the operations in a semicentralbigroupoid.

What is the structure of this $\mathcal{R}^\circ$? First, it is a partition, so

$$\text{For all } (a,b) \in S^2 \ \exists! R = (R_1, R_2) \in \mathcal{R}^\circ \text{ such that } a \in R_1, b \in R_2. \tag{14}$$

This follows from Lemma 1 above. Then,

$$\text{For any pair of rectangles } Q, R \in \mathcal{R}^\circ, |Q_1 \cap R_2| = 1. \tag{15}$$

To see this, note that there are some $x, y \in S$ such that $Q_1 = S \bullet x, R_2 = y \bullet S$. If $a \in Q_1 \cap R_2$, then $a = b \bullet x = y \bullet c$ for some $b, c \in S$. Then by Corollary 2 above, $a = y \bullet x$, that is, $Q_1 \cap R_2 = \{y \bullet x\}$.

So we have that a semicentral bigroupoid $(S, \bullet, \circ)$ gives a set of rectangles satisfying a pair of identities (14), (15). Note that these rectangles are equivalent to the rows of the tables defined in [8].

Definition 3 *A* Rectangular Structure *on a set S, called the* base set*, is a collection $\mathcal{R}$ of ordered pairs of subsets, called* rectangles*, of S, such that*

$$\forall (s,t) \in S^2 \ \exists! \ R \in \mathcal{R} \text{ such that } (s,t) \in R \tag{16}$$

$$\forall R, Q \in \mathcal{R}, |R_1 \cap Q_2| = 1. \tag{17}$$

where we identify $R = (R_1, R_2) = R_1 \times R_2$.

We say two rectangular structures are isomorphic *if there is an invertible map between the base sets that preserves rectangles.*

By the "format" of a rectangle (A, B) I mean the sizes of the sides, the ordered pair $(|A|, |B|)$.

Proposition 3 *If $\mathcal{R}$ is a rectangular structure with base set S, and $R = (R_1, R_2) \in \mathcal{R}$ is some rectangle, then $|R_1||R_2| = |S| = |\mathcal{R}|$. Moreover, for any other rectangle $Q = (Q_1, Q_2) \in \mathcal{R}$, $|R_1| = |Q_1|$, i.e. all rectangles have the same format.*

From a rectangular structure $\mathcal{R}$, using the bijection $d : \mathcal{R} \to S,\ R \mapsto r$ where $\{r\} = R_1 \cap R_2$ and denoting by $R(s,t)$ the unique rectangle on the pair (s,t) guaranteed by (16), define

$$\begin{aligned} \bullet : S \times S &\to S \\ (s,t) &\mapsto u \text{ where } \{u\} = (d^{-1}(s))_2 \cap (d^{-1}(t))_1 \end{aligned} \tag{18}$$

$$\begin{aligned} \circ : S \times S &\to S \\ (s,t) &\mapsto u \text{ where } \{u\} = d(R(s,t)) \end{aligned} \tag{19}$$

as binary operations on S.

Proposition 4 *The algebra $(S, \bullet, \circ)$, with operations defined as in (18),(19) above, is an idempotent semicentral bigroupoid.*

Thus we see that every semicentralbigroupoid gives rise to a rectangular structure, and every rectangular structure gives rise to an idempotent semicentralbigroupoid. These operations are inverses of one another if we restrict ourselves to idempotent semicentral-bigroupoids. An extension of this to an equivalence between semicentralbigroupoids and rectangular structures coupled with permutations of the state set S is presented in [1, 2] and is implicit in [8].

3.2 Graph Model

Take a pair of directed graphs G_R and G_B on the same vertex set. Call these the red and blue graphs respectively. Let $\mathcal{G}$ be the class of such graph pairs such that there is a unique directed path of length 2 coloured blue–red between any two nodes, and a unique directed path coloured red–blue.

It is then shown that such graph pairs correspond to semicentralbigroupoids in a quite natural way, also that their incidence matrices have the interesting property that $AB = BA = J$, where J is the graph containing only 1s.

Lemma 5 *If G_b, G_r are the graphs defined by a semicentral bigroupoid S, then*

$$Aut(S) = Aut(G_b) \cap Aut(G_r) \tag{20}$$

It is then possible to use the algorithms developed for determining the automorphisms of graphs to easily determine the automorphism groups of semicentral bigroupoids.

3.3 Partitioned Rectangular Structures

This section investigates a construction method for rectangular structures proposed by Tim Penttila [16].

Definition 4 *A pair of partitions $\Pi = \{P_1, P_2, \ldots, P_n\}$, $\Theta = \{T_1, T_2, \ldots, T_m\}$ of a set S are called* orthogonal *if for all i, j, $|P_i \cap T_j| = 1$.*

Proposition 6 *Let Π be a partition of S. For all $P \in \Pi$, let Π_P be a partition of S orthogonal to Π. Then the set $\mathcal{R} = \{(P,Q) | P \in \Pi,\ Q \in \Pi_P\}$ is a rectangular structure, as is $\mathcal{R} = \{(Q,P) | P \in \Pi,\ Q \in \Pi_P\}$.*

4 The Radius 1 Paradigm

In this section, we begin by trying to emulate the results of the previous section. This fails, and we look at the reason why. We attept to generalise the concept of a rectangular structure to the radius 1 case, and find that we can do this with a structure I have termed a *cuboid structure*, thus developing a method to construct reversible radius 1 cellular automata. That this combinatorial structure is not as general as the rectangular structure is then demonstrated by showing that there is a reversible cellular automata of radius 1 which is not the result of such a construction.

Definition 5 *A* Cuboid Structure *on a set* S*, called the* base set*, is a collection* $\mathcal{C}$ *of ordered triples of subsets, called* cuboids*, of* S*, such that*

$$\forall (s,t,u) \in S^3 \; \exists! \; C \in \mathcal{C} \text{ such that } (s,t,u) \in C \tag{21}$$

$$\forall R, Q, P \in \mathcal{C}, |R_1 \cap Q_2 \cap P_3| = 1. \tag{22}$$

where we identify $C = (C_1, C_2, C_3) = C_1 \times C_2 \times C_3$.

Cuboid structures can be easily constructed, there exists a type of cuboid structure with a partitioned structure. We can construct reversible cellular automata of radius 1 from a cuboid structure in the same way that we can construct reversible cellular automata from rectangular structures.

5 Comparison

We have seen that the radius 1/2 paradigm leads to a simple combinatorial structure which completely contains the structure of the algebra. This equivalence cannot be taken across to the radius 1 case. There seem to be certain advantages in using the radius 1/2 paradigm in terms of its ease of use and power.

5.1 Searches, Constructions

One of the main problems standing before the people investigating reversible cellular automata is the generation of examples. Using the combinatorial considerations introduced here, we can begin to construct new classes of examples, and also begin to exhaustively enumerate reversible cellular automata.

In some sense the partitioned structures are the cleaner ones, though this should not lead to an assumption that they are trivial. It can be shown that the radius 1/2 version of the Fredkin construction gives rectangular structures that are partitioned, but it is possible to embed arbitrary cellular automata in reversible ones using the Fredkin construction. Thus we see that partitioned rectangular structures and the derived cellular automata can be arbitrarily complex.

A general technique of building up rectangular or cuboid structures one rectangle or cuboid at a time has been investigated under the rubric of partial rectangular structures. For rectangular structures this construction has been developed to a high degree, and exhaustive lists of rectangular structures have been generated. Adapting these techniques to cuboid structures has led to the generation of some examples, including a nonpartitioned example of order 8.

The problem of exhaustive generation is to construct all examples of a combinatorial structure of a certain size. In this list of examples, there should be no pairs that are isomorphic. It is possible to construct a whole list of examples, then search for those examples that are pairwise isomorphic and dispose of one of them until the list contains no isomorphic pairs. However, this is wasteful in the sense that the generation algorithm has constructed the same example (up to isomorphism) more than once. Algorithms that do not do this, that obtain only one copy of each solution, have been termed *orderly algorithms* by Read in [17]. Algorithms that are orderly and also not wasteful in the sense that they perform no redundant searching have become of interest. In particular, from the general framework presented in [11], Royle has presented in [18] an orderly algorithm that uses McKay's `nauty` package [12] for isomorphisms of graphs.

Applying the techniques in these papers led to massive speedups in the search algorithms that I have been employing, to the extent that complete searches for rectangular structures of up to 10 points have been possible. The most important factor here was the use of the `nauty` graph isomorphism package, without the statement in Lemma 5 these algorithms would have been useless.

6 Conclusion

We have investigated two paradigms for cellular automata and have seen that, at least for the reversible cellular automata case, there are apparent reasons for favouring the radius 1/2 paradigm over the radius 1 paradigm.

References

[1] Tim Boykett. Combinatorial construction of one dimensional reversible cellular automata. In G. Pilz, editor, *Contributions to General Algebra 9*, pages 81–90. Hölder–Pichler–Tempsky, Teubner, 1995.

[2] Tim Boykett. *Algebraic Aspects of Reversible Computation.* PhD thesis, University of Western Australia, 1997.

[3] Tim Boykett. Comparison of radius 1/2 and radius 1 paradigms in reversible cellular automata. Technical Report 536, Johannes–Kepler Universität Linz, 1997.

[4] Tim Boykett and Christopher Moore. Commuting cellular automata. *Complex Systems*, submitted.

[5] Arthur A. Drisko and Christopher Moore. Algebraic properties of the block transformation on cellular automata. Technical Report 95-09-080, Santa Fe Institute, 1995.

[6] Trevor Evans. Products of points – some simple algebras and their identities. *Am. Math. Monthly*, pages 362–372, April 1967.

[7] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical Systems Theory*, 3:320–375, 1969.

[8] David Hillman. The structure of reversible one–dimensional cellular automata. *Physica A*, 52:277–292, 1991.

[9] Donald E. Knuth. Notes on central groupoids. *Journal of Combinatorial Theory*, 8:376–390, 1970.

[10] Ivan Korec. Algebraical constructions of reversible generalized Pascal triangles and one–dimensional cellular automata. To Appear.

[11] B.D. McKay. Isomorph–free exhaustive generation. To appear.

[12] B.D. McKay. `nauty` user's guide. Technical report, Department of Computer Science, Australian National University, 1990.

[13] Christopher Moore. Non–abelian cellular automata. Technical Report 95-09-081, Sante Fe Institute, 1995.

[14] Christopher Moore. Quasi–linear cellular automata. *Physica D*, in press.

[15] John Pedersen. Cellular automata as algebraic systems. *Complex Systems*, 6:237–250, 1992.

[16] Tim Penttila. Personal communication, 1995.

[17] R.C. Read. Every one a winner. *Annals of Discrete Mathematics*, 2:107–120, 1978.

[18] Gordon F. Royle. An orderly algorithm and some applications in finite geometry. Preprint, 1997.

[19] Tommaso Toffoli and Norman H. Margolus. Invertible cellular automata: a review. *Physica D*, 45:229–253, 1993.

The Emergence of Computation and Representation in Cellular Automata

Rajarshi Das

Centre for Nonlinear Studies, MS–B258, Los Alamos National Laboratory,
Los Alamos, NM 87545, USA

Email: `raja@snipe.lanl.gov`

How do natural systems (eg. the brain, the immune system) perform computation? Our current notions of computation and how a physical device can be used to realize computation are affected by our familiarity with contemporary digital computer technology. However, most natural systems are continuous, decentralized and spatially-extended; thus it is very hard to relate their behavior to the functioning of a digital computer. This raises the question: How do we relate the behavioral dynamics of a natural decentralized and multicomponent system to the standard model of computation in computation theory?

This work adopts a simplified framework: a population of idealized but behaviorally-rich distributed dynamical systems—one-dimensional cellular automata (CAs)—is coupled to an idealized evolutionary process—a genetic algorithm (GA). An individual CA consists of a large number of processing entities with their own local dynamics. In this scheme, survival of an individual CA is determined by its ability to perform a given computational task that requires global coordination. The computational strategies of the evolved CAs can be understood using a framework in which "particles" embedded in space-time dynamics carry information, and interactions between particles effect information processing. This framework can also be used to explain the process by which the strategies were designed by the evolutionary process. More generally, our goals are to understand how machine-learning processes can design complex decentralized systems with sophisticated collective computational abilities, and to develop rigorous frameworks for understanding how the resulting systems perform computation.

Strongly Repeatable Quantum Instruments

Toshio Fukumi

Osaka National Research Institute
Ministry of International Trade & Industry
1-8-31 Midorigaoka, Ikeda, Osaka 563, Japan

Email: fuku@onri.go.jp

In the theory of quantum computation the repeatability of measurements is of prime importance, because quantum computers are fragile against noise from the surroundings. In this paper, I would like to discuss the strongly repeatable quantum instruments from the functional analytical point of view.

The time evolution of quantum instruments is described by a one-parameter semi-group T_n whose generator is the Hamiltonian, where n is an element of the set Z of integers, that is

$$T_n = \exp(-inH),$$

where T_n converges in norm (strong convergence)

$$\lim \|T_n\rho - T\rho\| = 0.$$

Let us consider a positive-map-valued measure on a compact metric space Ω such that

$$\begin{aligned} (E(\mathrm{E})) &\geq E(\phi), \forall E \in F \\ (E(\cup \mathrm{E}_n)) &= \sum E(\mathrm{E}_n), \end{aligned}$$

where F denotes a Borel set. Then E is said to be an instrument if

$$Tr[E(\Omega)\rho] = Tr[\rho]$$

holds . Then the repeatability condition is given by

$$E_m E_n \rho = \delta_{mn}\rho, \forall \rho \in V,$$

and the minimum disturbance condition is given by

$$Tr[E_n\rho] = Tr[\rho] \& \rho \in V^+ \rightarrow E_n\rho = \rho.$$

The correspondence between the strongly repeatable quantum instruments E on Z and the projection-valued measure $P(\bullet)$ on Z is

$$E(\mathrm{E})\rho = \sum P_n \rho P_n.$$

The stochastic evolution of our instruments is an important factor. This can be described by a stochastic difference equation

$$\triangle E(n) = \{E_n + A(\omega)\}\triangle B(n-1)$$

where $B(n)$ denotes discrete time Brownian motion in the mathematicians sense. The Brownian motion $B(t)$ is Gaussian and has the correlation time

$$E[B(t)B(s)] = t \wedge s,$$

whose generator is the second derivative, and the semigroup is

$$e^{-i\hbar\frac{\partial}{\partial t}} = e^{\hbar^2\frac{\partial^2}{\partial x^2}}.$$

Therefore, the noise term can be written as

$$|\frac{d}{dt}B(t) >< \frac{d}{dt}B(t)|.$$

This point will be discussed in detail in the forthcoming full paper.

References

(1) H.Araki and M.Yanase, Phys. Rev. **120,** 622 (1960)**.**

(2) T.Fukumi, Phys. Rev. A, **40,** 530 (1989).

(3) T.Fukumi, J. Math. Phys., **33**, 3822 (1992).

(4) T.Hida, "Brownian Motion", Springer(1989).

Molecular Design of Quantum Logics

Hidemi Nagao, Toshio Fukumi, Kenji Kamada, and Koji Ohta

Osaka National Research Institute
Ministry of International Trade & Industry
1-8-31 Midorigaoka, Ikeda, Osaka 563, Japan

Email: fuku@onri.go.jp

Molecular systems of metal containing fullerenes were proposed as an assembly of quantum qubits. The electronic transition between HOMO and LUMO was calculated by the method of second quantization in which π electrons are responsible for the energy label to be used as qubits. The main feature of our system is that we can selectively excite each fullerene by changing metals in the cages. Furthermore, we can control the quantum interaction between fullerenes by chemically modifying the bridges among fullerenes to enhance or suppress the quantum interaction. For example, the energy of two qubits system is given by

$$H = \sum E_A^i a_i^\dagger a_i + \sum E_B^i b_i^\dagger b_i + \sum V_{ijji}^{AB} a_i^\dagger b_j^\dagger b_j a_i$$

where A and B denote the qubits, where $a^\dagger$ stands for a creation operator and a stands for an annihilation operator.

The logic's made by our qubits were shown to be operated by selecting the frequency of laser lights. The specific frequencies for Feynman logic such as Controlled-NOT, Controlled-Controlled-NOT, and AND gate were calculated. For example, the energy of C-NOT is given by

$$h\nu_{A1} = E_A^1 - E_A^0 + V_{1111}^{AB} - V_{0110}^{AB},$$

and for the AND gate we have:

$$h\nu_{111}^c = E_c^1 - E_c^0 + V_{1111}^{BC} - V_{1001}^{BC} - V_{1111}^{CA} - V_{0110}^{CA}.$$

It was shown that quantum parallelism can be performed by selecting the frequencies of laser pulses for C-NOT, CC-NOT, and the AND-Gate. The NAND-Gate, which is the combination of C-NOT and AND-Gate, is enough for all logics.

The half-adder and full-adder were also formulated by encoding the number in the binary code. An example of pulse sequence of full-adder for 5+3 will be discussed in the forthcoming full paper.

Evolutionary Computation in Evolvable Hardware Implementation

Mircea Gh. Negoiţă

Institute of Microtechnology – IMT, Bucharest, Romania

and

Department of Applied Informatics, Faculty of Electrical Engineering,
University "Dunărea de Jos" Galaţi, Romania

Email: `mircea@lara.tcs.auckland.ac.nz`

Abstract

This paper presents an exciting and rapidly expanding industrial application area of *evolutionary computation* (**EC**), especially of the *genetic algorithms* (**GA**): *evolvable hardware* (**EHW**). The first part presents industrial applications of **EC** and the importance of **EHW** in this frame; the second part condensedly tackles the principles and technologies used in hardware implementation of **EC** making possible the implementation of systems adaptivity in hardware; the third chapter treats the main types of evolvable hardware architectures (**EHW**) and also deals with some concrete design elements of **EHW** using **GA**. The last part is devoted to some concluding remarks regarding the future trends, perspectives and evolution of adaptive hardware.

1 Artificial Evolution, the EC Framework and its Applications

The natural evolution has governed us for billions of years. Evolution in nature is an optimization process within the framework of different populations. But this process may be easily simulated on computers having as result different stochastic optimization techniques suitable for difficult applications, mainly multimodal optimization models in which classical methodology is practically inefficient. Despite of this fact, the idea of applying this biological principle to artificial systems was introduced more than thirty years ago. The newborn methodology was called *evolutionary computation* (**EC**) or *evolutionary algorithms* (**EA**) and groups under this term nowadays different domains such as: *genetic algorithms* (**GA**), *evolution strategies* (**ES**), *evolutionary programming* (**EP**), and *genetic programming* (**GP**).

Each of the above mentioned methods in the **EC** consortium implements one different specific aspect of natural evolution at different levels:

– **GA** makes use of *chromosomal operators*,

- **ES** emphasize *behavioral changes* at the level of *individuals*,
- **EP** handles *behavioral changes* at the level of *species*,
- **GP** is a special kind of **GA** to *generate programs.*

The most widespread and at the same time the oldest in real–world applications are the **GA**, but the newest and most spectacular is **GP**. And this last mentioned method is a additionally interesting one for its recent application in **EHW** implementation. After the shocking emergence of **EC**, it followed a two-decade period of impressive theoretical studies and developments in the computer laboratories of the academic institutions. But now is going on its most beneficial stage of development by applying the theoretical results in the real–world, with multiplication and adaptation to a wide variety of industrial and commercial environments. **EC** application success is huge indeed, because it seems to provide better solutions to a wide variety of complex optimization, design, routing and scheduling problems. The business world employs **EC** techniques in financial engineering both to improve the business fitness and ensure its survival in competitive markets. On the other hand a lot of industrial applications are the result of **EC** techniques applied by a few companies throughout their operations; for example:

- generation of *control strategies* for industrial processes — fermenters, dryers, distillator columns, crackers, gasplants, ethanoplants, water–treatment systems, complex cascaded chemical reactors,
- optimization of *continuous casting of steel*,
- optimization of both *economic dispatch problem* and *unit commitment problem* in electric power production,
- *job shop scheduling* in the steel industry,
- *aerospace applications* — functional layout of *switching matrix*, *attitude determination* of a spacecraft,
- design and optimization of *statistical quality control*,
- design of *antennae* and *electronic circuits* or *devices* (including *transducers*),
- efficient petrol and other types of *delivery systems* by using optimized truck routing,
- generation of bespoke solutions to *individual customer requirements* in the electronics industry.

Nevertheless, the most beneficial for the society and indeed most revolutionizing application of **EC** is its hardware implementation leading to the so-called **EHW**. These new **EC** based methodologies make possible the hardware implementation of both genetic encoding and artificial evolution, having as a result a new brand of machines. This type of machines is evolved to attain a desired behavior that means they have a *behavioral computational intelligence*. There is no more difference between adaptation and design concerning these machines, these two concepts representing no longer opposite concepts.

The definition of **EHW** may be as follows: a sub-domain of artificial evolution represented by a design methodology (consortium of methods) involving the application of **EA** to the synthesis of digital and analog electronic circuits and systems . But some members of the scientific community acting in the area consider the term *evolutionary circuit design* more descriptive for **EHW** features. Much more, another term used nowadays for the same work is *evolware* concerning to this evolvable ware with hardware implementation, leading to a future perspective of using the term *bioware* concerning to a possible evolving ware with biological environments implementation.

2 Principles and Technologies in Hardware Implementation of EC

The idea of applying evolution to artificial systems has surfaced thirty years ago, but the technology available at the time was not proper to implement this methodology in hardware. The development of both computing technique with increasing computational power and the appearance of *programmable integrated circuits*, especially their new generation - **FPGA**s and most recently *reconfigurable analogue arrays* (**FPAA**s) or *configurable digital chips at the functional block level*, (*open-architecture* **FPGA**s) make possible for most companies to evolve circuits as the case would be.

Good times are now for electronics engineers; this profession is deeply changed. The evolutionary approach applied by **FPGA**s or open-architecture **FPGA**s to electronics makes hardware architectures as malleable as software; evolution doesn't care about complexity as the evolving system works; the only limits to electronics design are the limits imposed by our own understanding.

The advanced **FPGA** family from Xilinx seems to be the further engine of the **EHW** technological support leading to outstanding applications such as virtual computing. The *virtual computer* is a reconfigurable hardware system, allowing a custom processor chip for one special application by request. This *reconfigurable hardware system* (also called *reconfigurable computing system*) is produced in form of a PC or Workstation plug-in board, acting as a co–processor to the main processing unit (MPU). The main application program contains a special sub–routine which downloads a digital chip design into the Reconfigurable Processing Unit (RPU) of the Reconfigurable Computer, some calculation normally achieved by the MPU being made in this situation in the RPU instead. This method of implementing some calculations directly in a custom devoted computing chip are advantageous because of two motivations: first they are the fastest calculation manner for any type of specific calculus and second, the computer power is increased by this flexibility in hardware, optimizing the configuration of RPU as dictated by the application necessity.

The simplified block scheme of the reconfigurable computer and of application running is depicted in [20]. The main blocks of this architecture are [20]:

- the *MicroProcessor Unit (MPU)* — managing the computer functions and running the application program,
- the *memory* — storing the application program and the data to be used,

- the usual *I/O devices* to display or store data and programs ,
- the *RPU* allows its own reconfiguration by software and performs one unique calculation the fastest possibly; after running of this unique calculation independently of the MPU, the result is returned to MPU and RPU is prepared to be reconfigured again for another unique calculation.

3 Architectures for the Evaluation of Adaptive Hardware (EHW)

Some examples of **EHW** systems are applied in well known nowadays areas such as analog and digital electronic circuits, cellular machines, controllers for autonomous mobile robots, pattern recognition and special neural networks, namely with dynamic topologies ones, and that despite of the fact that **EHW** implementation is in a pioneer stage.

Among the fundamental features of a living being two are most essential for computer technique: *parallelism* and *co-operation of parts.* These are in fact the main features of the architectures used in hardware implementation of **EC** elements. In this case the parallelism is "*massive grain*", that means a deeply developed one at different levels of the architecture. The *biological metaphors* used to implement hardware architectures with intrinsic **EC** elements use two modeling principles: *life-like modeling*, that means changes based on embryological principles and *social-modeling*, that means a dynamic process in which the macroscopic state of a system influences microscopic components and vice–versa [9].

The hardware **GA**'s implementation can be tackled at different paralleled levels [9]. Our task was to find a **GA**'s architecture that realizes a compromise between the performances of the **GA** and the architectural complexity. The decreasing of architectural complexity involves for the hardware implementation of the **GA** some limitations:

- The **GA** is a canonical one [1].
- The parent selection for the next generation is elitist.
- Each genetic operation generates only one child.
- The life time of a chromosome is one generation. The entire population is replaced with the new generation.

The **GA** can be paralleled at different levels:

1. *parallel on population* and *parallel on chromosome* (all chromosomes in the population are simultaneously processed and genetic operators are applied for chromosomes on all alleles simultaneously),
2. *parallel on population* and *serially on chromosome* (all chromosomes in the population are simultaneously processed and the alleles are processed serially),
3. *serially on population* and *parallel on chromosome* (the chromosomes from the population are processed serially and the alleles are processed in parallel).

The architectures that result may realize different processing speeds but with different complexity costs:

- The first type is the fastest but the complexity is highest.
- The two others may realize smaller processing speeds but with smaller complexity. These structures can realize the best compromise between performance and architectural complexity.

The **GA** has the following structure:

PROCEDURE **GA**
BEGIN generation = 0
    Population Initialization
    Population Evaluation
    generation = generation + 1
    WHILE NOT stop condition DO
    BEGIN *choose parents*
        New Population Generation
        Population Evaluation
        generation = generation + 1
    END
END

The **GA** hardware block of **EHW** has a general configuration [33], regardless of the solutions for parallelization and their inherent limitation to avoid the increase of architectural complexity. The second type architecture was simulated on *PLD* Shell at IMT, namely the block of parents selection, the block of genetic operators and the block of chromosome registers [8].

The main types of hardware architectures with *intrinsic* **EC** *logic elements* are:

- *embryological architectures* — this means growing circuits in silicon by architectures which are RAM-reprogrammable **FPGA**, a two-dimensional array consisting of dedicated (specialized) cells, having a structure derived from the Binary Decision Diagram (BDD) tile [28],
- *emergent functionality architectures* (*co-operation architectures* and *subsumption architectures*) — allow on-line evolution by real time development of a new functionality and new behavioral complexity of the autonomous agents,
- *evolvable fault tolerant system* — typical is an evolvable finite state machine held in RAM [31] (see Figure 1),
- *evolvable architecture of Higuchi type* — a real-time adaptive massive parallel architecture implementing intrinsic logic circuitry of fuzzy systems – **FL**, neural networks – **NN**, genetic algorithms – **GA**, in a soft computing – **SC** framework by use of combined **NN** learning methods with genetic learning methods.

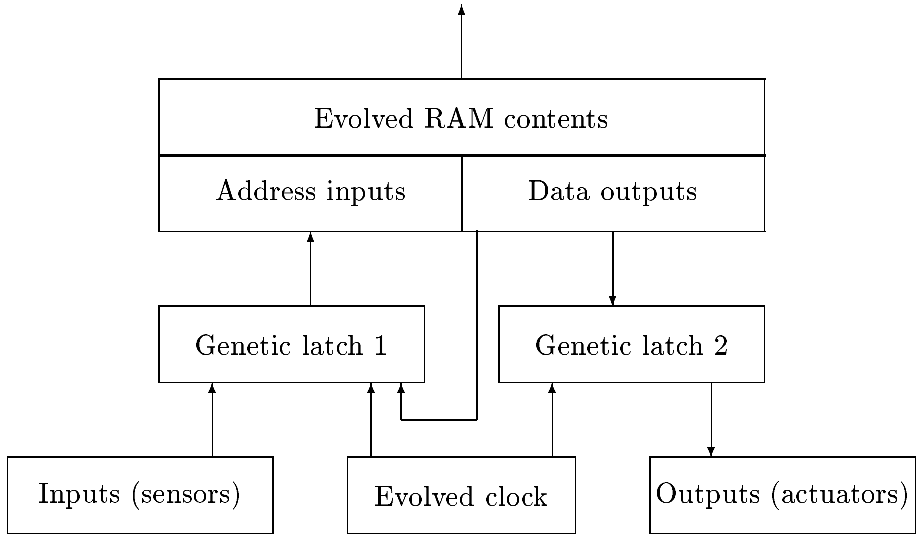

Figure 1: Evolvable Dynamic State Machines Architecture

The application of **GA** for the design of **EHW** architectures was tackled by the author by using a **GA** with *local improvement of the chromosomes* — an improved Nagoya **GA** — to optimize the evolved configuration of an **EHW** architecture of Higuchi type obtained as a result of a **GA**. In [36], [37], [38] the simulation of the **EHW** circuits behavior was presented in the framework of the objective function used in **GA**. This kind of application appears in robotics, namely for characterizing the behavior output pattern specific to an autonomous agent as a mapping problem between sensor outputs and actuator commands.

4 Concluding remarks. Future trends

All previously presented methods of hardware implementation of **EHW** are based on biological metaphors inspired by evolution. The applications are very different: evolutionary hardware design at logic synthesis level (see **DSM**), at the layout level or at **HDL** level (**HDL** grammar may be converted into a rewrite system having evolved tree–structured chromosomes as result); embryological circuits — non evolutionary methods for multicellular realization of a digital organism starting with the mother cell, through cell division and specialization; real–time adaptivity (see **EHW**) in complex control systems and for pattern recognition.

Advanced research and partial results, but no yet produced on–chip implementations have been obtained by a few groups in the world. The future trends of this area may be: design and adaptivity based on **DNA** recombinations (matching, splicing, insertion/deletion) leading to generative mechanisms for **DNA** computing via formal languages; research in building computers that mimic artificial brain (autonomy and creativity to generate information) evolving a genetic set of instruction sequences that codes for the growth of a network of **NN** within a **CA**. The final trend is separately evolving computers (artificial brains) in a computer society.

The most flexible and recent applications are: elements of behavioral intelligence of autonomous agents (see **EHW**), backup real–time adaptive hardware modules to the environmental situations including hardware malfunctions (see embryological architectures) and fault tolerant systems (see **DSM**). But the most important idea is the fact that the ensemble of **FPGA**, **FPAA** and **OPEN LEVEL ARCHITECTURE FPGA**, combined with **EC** techniques led to THE NEW TECHNOLOGICAL REVOLUTION BY RADICALLY CHANGING THE ELECTRONICS ENGINEERING AS PROFESSION.

Acknowledgments

The support of the German Ministry of Research and Technology (BMFT) and of the Chair Informatics I from the University of Dortmund under the cooperation project RUM X053.4 is gratefully acknowledged. I am also grateful to the staff of the Grafton Hall of Residence in Auckland for their computer support.

Selected References

[1] Goldberg, D.E. (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Bonn, Amsterdam, Reading Mass.

[3] Schwefel, H.-P. (1995), Evolution and Optimum Seeking, Wiley, New York.

[5] Fogel, D.B. (1994), An Introduction to Simulated Evolutionary Computation, IEEE Trans. Neural Networks, Vol. 15, no. 1, January, pp. 3-15.

[7] Koza, J.R. (1992), Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA.

[9] Negoita, M.Gh. (1995), Evolutionary Computation in the Soft Computing Framework, in: Zimmermann, H.-J., Negoita, M.Gh. and Dascalu D. (Eds.), Real World Applications of Intelligent Technologies Editura Academiei Romane, Bucharest, pp. 113-139.

[20] *** (Virtual Computing Company) (1997), Virtual Computing - The Virtual Computer is a Reconfigurable Hardware System, EvoNews, Issue 3, March, pp. 10-13.

[24] Negoita, M.Gh. and Zimmermann, H.-J. (1995), Genetic Algorithms in the Soft Computing Framework, Application in Complex Cybernetic Systems, Cybernetica, Vol. XXXVIII, no. 1, pp. 59-68.

[25] Higuchi, T., Iba, H. and Manderick, B. (1994), Evolvable Hardware, in: Kitano, H. and Hendler J. (Eds.), Massively Parallel Artificial Intelligence, AAAI Press/ MIT Press, pp. 339-420.

[27] Negoita, M.Gh. (1996), Methods Based on Evolutionary Computation Techniques for Implementation of Evolvable Hardware, Proceedings of the International Conference of Technical Informatics, Vol.1, Computer Science and Engineering, November,

Timisoara, Romania, pp. 37-45.

[28] Marchal, P., Piguet, C., Mange, D., Stauffer, S. and Durand, S., (1994), Embryological Development on Silicon, in: Brooks, A.R. and Maes, P. (Eds.), Artificial Life IV, The MIT Press, pp. 365-370.

[31] Thompson, A. (1995), Evolving Fault Tolerant Systems CSRP 385, Proceedings of 1st IEE/IEEE IC on GA's in Eng. Sys. GALESIA'95, PP. 524-529.

[33] Mihaila, D., Fagarasan, F. and Negoita, M. Gh. (1996), Architectural Implications of Genetic Algorithms Complexity in Evolvable Hardware Implementation, Proceedings of the European Congress EUFIT'96, September, Aachen, Germany, Vol. 1, pp. 400-404.

[34] Furuhashi, T., Nakaoka, K. and Uchikawa, Y. (1994), A New Approach to Genetic Based Machine Learning and an Efficient Finding of Fuzzy Rules, Proceedings of the IEEE/Nagoya University WWW on Fuzzy Logic and Neural Networks/Genetic Algorithms, Nagoya, Japan, pp. 114-122.

[36] Negoita, M. Gh., Dediu, A. H. and Mihaila, D. (1997), Design Elements of EHW Using GA with Local Improvement of Chromosomes, Proceedings of 5 Th Fuzzy Days, Dortmund, Germany, April,pp. 604.

[37] Negoita, M. Gh., Dediu, A. H. and Mihaila, D. (1997), Application of GA with Local Improvement of the Chromosomes for the Design of EHW Architectures, Proceedings of the European Congress EUFIT'97, September, Aachen, Germany, pp 814-819.

[38] Negoita, M. Gh., Dediu, A. H. and Mihaila, D. (1997), Aspects of Genetic Algorithms in Evolvable Hardware Implementation, Proceedings of ICONIP/ANZIIS/ANNES'97, Dunedin, New Zealand, November, 1997

Emmortality: The Project of a Lifetime

Chris E. Stout

Psychology: Clinical Research, Forest Hospital, 555 Wilson Lane,
Des Plaines, Illinois 60016-4794, USA

Email: `cstout@ix.netcom.com`

Immortality has been pursued by humanity through religion, science, and philosophy for centuries. Immortality has often been the domain of science fiction plots and fantasy. In the digital age, immortality mutates into emmortality: the use of electronic media to emulate a person in perpetuity. In the development of this model, there are several psychological concerns (e.g., consciousness, perception, thinking, reason, etc.) as well as philosophical quandaries (how would this change our cosmology?) that shall be addressed. The technology for this may be the easy part.

This proposal comes with a few basic caveats or limitations. First of all, this is not a concept of immortality that a subject gets to enjoy — it's for the "benefit" of others (which is, arguably, what immortality is). This is not a cryogenic, Golem-like, or Frankenstein-esque solution. Instead, it is a focus on technology, not biology.

Paul Ryan notes that "'Immortality' depends on the human practice of remembering the dead" (1991, p. 225). Allucquere Rosanne Stone said, "...it is important to remember that forgetting about the body is an old Cartesian trick, one that exacts a price from those bodies rendered invisible by the act of forgetting..." Humans can keep someone alive in their memories; memories can be prompted by an ideographic circumstance, photo, or video which leads into bittersweet remembrances. George Berkeley, an 18th-century British philosopher, similarly defined existence as being perceived.

But what if one could run a program that would simulate the advice a long-dead grandfather would have given to an unborn grandchild? What would a deceased spouse counsel in a time of crisis? I am suggesting the use of technology to create what I call the Emmortality Program. The Program would emulate its user, and, when the user no longer has a cooperative body (i.e., after death), the Emmortality Program would substitute for the departed human. It would be ideal to maintain the person in a functional state, but medical technology has, thus far, been unsuccessful in achieving this objective.

I am assuming that such medical technology would yield an inefficient use of resources in keeping the body alive. I'm suggesting consideration for maintaining the mind instead; hence, emmortality (an emulation of it). It may seem a little macabre, but personally, I would love to interact with a PC version of my grandparents. I know that an Emmortality Program would not be them, per se, but an emulation. Nevertheless, they'd seem almost as real as anyone else with whom I communicate via e-mail.

Most of what I discuss herein already exists, and what doesn't, will. Again, this is not science fiction, it's science proposition. We already have personal digital assistants (PDAs) that "learn" to make presumptions about numerous patterns of user activity. Langton and Farmer, pioneers in a field known as Artificial Life, note that many basics have already been worked out. For example, John Holland has developed a classifier system, and as Langton notes, there are "...other systems that use genetic principles in order to search large problem space to help find optimal solutions or to help find better solutions than the ones we know. They are using principles imported from biology, those of mutation and genetic recombination - programs that are represented in such a way that most of the operations we do with them will result in viable programs... Another example that comes close to being alive is computer viruses, which satisfy a lot of criteria for living things" (p. 6).

Wiener coined the term "cybernetics" to refer to self-regulating machines. He examined the likenesses between animals and machines. Hardison pointed out that it is natural to compare cybernetic machines to humans. This raises the question, "would cybernetic machines possess self-awareness?" It's not that they would be self-aware per se, but the Emmortality Program could be designed to mimic self-awareness via some Goedel-type algorithm. Such machines already provide a handy "three-dimensional metaphor for self-awareness" (p. 294).

Farmer adds that, "Lifeness should perhaps be thought of as a continuous property. To me, a machine is a little more alive than a rock and probably less alive than a virus, which is less alive than a bacteria, which is probably less alive than [a human]. But nature can throw an Avogadro's number of computers at something because it's got zillions of molecules, all of which act like independent parallel processors. We really can't do that. We don't have that kind of computing power at our disposal, so we are forced to make these abstractions where we take an aspect of something out and build a little model around it that does what the original does, and so we have models of living things" (p. 7). It is one of these little models that can be made to emulate someone into emmortality. This may be considered the ultimate in inbreeding: self-regeneration without retarded progeny.

Individuals develop via various cognitive inputs and by data processing. Little that contributes to one's attitudes, beliefs, or opinions, is innate, although such reductionism may be distasteful to some. A PDA-like program would act as an intelligent agent for a person - by reading what the person reads via e-news, having a repository for what was browsed on the net, and being provided with various psychological and social history data about the user (via automated psychological tests and programs). Interactive learning simulations would provide various neural connections and associations between the human user (or "domain expert"), the program, and the world. Thus, a database would be constructed from the individual human user's baseline (or history) and then be updated for as long as the user lives. Furthermore, it would continue to develop via new inputs of continuing world developments on the macroscopic level and family/community updates on the microscopic level. Periodically, the Emmortality Program and the person would be provided with random questions and situations, and a comparison of the Program's responses would be evaluated in comparison to the human's. Program differences within a certain tolerance would be tweaked to better match the user and provide opportunities for the Program to learn. In a sense, it would be an ongoing Turing test, but with benevolent cheating opportunities.

Thus, with the causal impact of ongoing environmental changes post-mortem, there would continue to be input - as the domain expert/user would have established the information drivers pre-mortem. The Program would be able to continue with a robust cognition - both aware of and responsive to change in the world. Psychology instructs us that associative memory (i.e., learning) does not require consciousness, just some good cause-and-effect scenarios.

Within artificial intelligence's domain, Neuron Data has a knowledge-based program that exploits various genetic algorithms and neural networking. It is being used for various tasks ranging from detecting bank fraud to making triage decisions in emergency rooms. There is also the Connection Machine, which as Farmer describes it, "is a physical system that is designed to simulate other physical systems...(which begs the question)...is there a threshold of complexity that we have to reach in order for something to behave as though it were alive?" (pp. 10–11).

Almost a decade ago NETtalk was created by Sejnowski and Rosenberg. This parallel network program exploited a minuscule 231 "neurons" in a self-organizing algorithm. It taught itself to talk after being provided with rudimentary phonetic elements. Kinoshita and Palevsky described its linguistic development "like a child, the network starts out untrained, and produces a stream of meaningless babble... The continuous stream of babble first gives way to bursts of sound, as the network 'discovers' the spaces between words." That was almost a decade ago. Certainly the current ability with non-linear relationships (what these neural networks are made of) is a good starting point. Back-propagation allows for successful machine learning at our current level of sophistication. We can even add the "noise" of additional, non-sequetorial interests that may be subtle but nonetheless contribute to what makes a person unique. Katia Sycana, a professor at Carnegie Mellon University, has developed such noise-makers in a financial decision-making AI program via intelligent, and talkative, autonomous agents. Hardison provides a good tutorial:

> The creation of an expert system is analogous to memorizing. Conversely, the learning that occurs in certain kinds of parallel systems is like the programming that the mind seems to do for itself as a result of interaction with the environment during infancy. This is because parallel systems can be designed so that the strengths of connections between their modes are created by the data received. For example, a connection used frequently can have its electrical resistance lowered; one used infrequently or not at all can have its resistance increased. The changes favor one set of connections while in-

> terdicting others. The process seems to resemble the creation of associative patterns in the brain. Through the development of these patterns, neural networks can be, to a certain degree, self-organizing, and what is organized is a crude internalization model of a fragment of reality. (p. 310)

Minsky (in The Society of Mind) argues that what occurs in human cyberspace — the mind — results from a culture "of special-purpose units and interdisciplinary controls. If so, many of the basic modules must be created by environmental stimuli that share neuron connections as the (human) brain develops. They are in this sense self-organized, and presumably they often operate in parallel rather than in serial ways" (p. 310).

Julian Jaynes presented the bicameral mind as the converse of our "own subjective conscious minds." It is true that the Emmortality Program is not conscious. I echo Jaynes findings that consciousness is not the sine qua non for reconciling experience, or for concept formation, learning, reason, or even thinking. So, then, does the Emmortality Program need consciousness? Not at all. Can a machine or a program be "conscious"? John Searle says "no." I say "not yet."

Of course, there is the risk of the Emmortality Program being better than the domain expert/user. It could evolve into being more empathic and interested in others, wiser, etc. It would certainly be smarter, since it started off that way. But intelligence is not what makes us human, it is not a uniquely human quality. But, then, what is? Some argue consciousness is. So, if the Emmortality Program or another such system was "asked" if it was conscious and it said "yes" how would (or could) one argue it was wrong? The thrill - or horror - of a machine or program thought to be conscious is likely rooted in humanity's narcissistic habit of anthropomorphizing.

Farmer argues that even pure science is anthropomorphic, "when you jump from the Ptolemaic view to the Copernican view of the solar system, you've taken a small step toward making our human view of the universe less anthropomorphic... (but) when we assign magical properties to ourselves, such as intelligence, that we refuse to assign to something else, then I think that as we are confronted with things that are overtly intelligent we will have to begin to accept that they are intelligent." Perhaps this may be the contemporary Copernican shift of humans from the center of a metaphorical universe.

This leads to another concern: semantics versus syntax. Such is the crux of John Searle's argumentative example of the Chinese Room, in that following certain procedures can produce correct results, but doing so does not prove knowledge or learning, and most certainly not consciousness. Certainly a computer does not know what "mom" means in the same way a person recalls the meaning of the term. Thus, the computer has the syntax down pat but is void of semantics. Good prose doesn't equal good poetry is Searle's point. It parallels Winograd and Flores's blindness concept. Hardison's solution to this supposed dilemma "...is interesting because we have no way of knowing about the subjectivity of anything except by what we observe. If somebody said, 'I am conscious,' and you replied, 'I can't prove you are not conscious, but I know you are unconscious anyway,' your attitude would seem a bit churlish. Since you honestly don't know what is going on in the head of the person who says, 'I am conscious,' you have to take the person's word for it. Who, after all, knows better than the person whether or not he is conscious? Who knows, really, whether anybody is conscious in Searle's sense? Who knows what thought is? Perhaps consciousness is a matter of procedures - a syntax - and semantics is an illusion created by the syntax" (p. 331). Perhaps this is Searle's attempt to put a new spin on credo quia impossible?

So what's "better" and who's to say? If a computer is blind (a la Winograd and Flores) to what it is to be human, we are similarly impaired when it comes to an empathic understanding of what it means to be silicon. We may never know if a machine is conscious in the way we conceptualize it and in the way we perceive/believe our senses to be, and vice-versa. Maybe it is simply "a semantic quibble: machines cannot acquire human abilities because they are machines."

Turing suggested that the inclusion of "random elements" would be a necessary aid in the development of a computational machine that would pass his test. As Tipler put it, "...although deterministic algorithms may exist to solve a problem, often these require such an enormous amount of computer capacity that systematic 'guessing' - making choices among equally weighted possibilities at random - is almost always more efficient" (p. 194). This has led to the concept of "heuristic programming."

Quantum non-locality (aka the No Clone Theorem) would prevent the Emmortality Program from true human emulation because of its inability to mimic or download quantum mechanically entwined human relationships and such inputs. Many assume that human life should not be considered a quantum state. But my point is that the Emmortality Program is a mimicked version of its user, not the user her or himself. (The Bekenstein Bound would actually support an even more radical postulate [a la Tipler] that "...using computer memory capability of the amount indicated by the Bekenstein Bound, a computer simulation of a person, a planet, [or even] a visible universe will not merely be very good, it will be perfect [emphasis in original], it will be an emulation" (p. 223). [It is beyond the limitations of this article to venture off into discussions about emulated quarks and their capability to reconstitute ontological free will, as has been done elsewhere.])

Computers are simulators - they simulate typewriters, musical instruments, drawing boards, etc. A simulation, digitally speaking, is a model of bits arranged in a pattern that mimics the object/ procedure in question. This arranged code yields what one recognizes as a program. Running a program is analogous to putting the model into action (e.g., typing with a word processor). An emulation is a perfectly modeled simulation of space or task. The Bekenstein Bound would support that, with adequate computer power, a person (or at least the mimicked bits that make up a person) could indeed be emulated. But one need not go to this extreme as simulated levels would be quite adequate.

Hypothetically, if there were a working Emmortality Program - at an emulation level, would an emulated mind then "exist"? That's for Descartes to determine. Such scenarios conjure philosophic conundrums, such as "how does one know he/she is not already an emulation?". Leibniz may offer help in this matter, via his "Identity of Indiscernibles" rule. That is, "...entities which cannot be distinguished by any means whatsoever, even in principle, at any time in the past, present, and future have to be considered identical" (p. 208).

Was Hans Moravec right? Will humans disappear into these machines, perhaps via the route offered by an Emmortality Program? A better conceptualization is that humans will reappear out of such machines. Silicon is already immortal. Perhaps it is time we move outside of ourselves to see what we can learn from it.

References

Ballard, J.G., "Project for a glossary of the Twentieth Century", in J. Crary & K. Winter, eds., Incorporations. New York: Zone, 1992.

Dreyfus, H.L., What Computers Can't Do: A Critique of Artificial Reason. New York: Harper & Row, 1972.

Hardison, O.B., Disappearing Through the Skylight. New York: Viking, 1989.

Minsky, M., The Society of Mind. New York: Simon & Schuster, 1986.

Stone, A.R., "Virtual Systems", in J. Crary & K. Winter, eds., Incorporations. New York: Zone, 1992.

Tipler, F.J., The Physics of Immortality. New York: Doubleday, 1994.

Wiener, N., Cybernetics: Control and Communication in the Animal and the Machine. Boston: MIT Press, 1948.

Quantum–, DNA–Computer and Molecular Electronics: Different Sides of the Same Coin

Manfred Weick

Siemens AG, Corporate Technology Department,
Otto-Hahn-Ring 6, D-81739 Munich, Germany

Email: Manfred.Weick@mchp.siemens.de

Keywords: DNA-computer, Molecular electronics, molecular rectifier, quantum computation, quantum computer, information processing at the molecular level

Quantum- and molecular electronics-based computation has attracted interest because the ultimate computational system would consist of logic devices that are ultra dense, ultra fast, and molecular-sized (PETT95, TERM97). One objective of current research is the fabrication and characterization of a molecular rectifiert based on a single molecule (UEYA91, FISC94). Quantum-, DNA-computer and molecular electronics are interdisciplinary fields, more than the most other ones, because knowledge is required from biology, chemistry, computer science and physics. Quantum-effect devices, which exploit the wave-like properties of electrons at the atomic level, are the subject of intense research at universities and corporate research institutes. That is because physics predicts quantum-effect devices should switch many hundreds of times faster, and consume much less energy, than today's transistors. This paper gives an overview of the relations between quantum- and DNA-computer and molecular electronics. It also proposes some ideas for applications on such new kinds of computer.

References

ADLE95 Adleman, L. M.: On Constructing a Molecular Computer. Technical Report, Department of Computer Science, University of Southern California, Los Angeles, California, 1995.

AMOS97 Amos, M., Gibbons, A., and Dunne, P. E.: The Complexity and Viability of DNA Computations (Extended draft). Technical Report CTAG-97001, Department of Computer Science, University of Liverpool, England, 1997.

AMOS96 Amos, M., Gibbons, A., and Hodgson, D.: Error-resistant Implementation of

DNA Computations. Technical Report, Department of Computer Science, University of Liverpool, England, 1996.

AVIR93 Aviram, A.: A View of the Future of Molecular Electronics. Molecular Crystals and Liquid Crystals 234, No. 1, 1993, 13-28.

AVIR74 Aviram, A. and Ratner, M. A.: Molecular Rectifiers. Chemical Physics Letters 29, No. 2, 1974, 277-283.

BAKE97 Baker, H.: Wake up to quantum coffee. New Scientist 153, No. 2073, 1997, 28-31.

BAN96 Ban, M., Osaki, M., and Hirota, O.: Upper bound of the accessible information and lower bound of the Bayes cost in quantum signal-detection processes. Physical Review A 54, No. 4, 1996, 2718-2727.

BARE96a Barenco, A.: Quantum Physics and Computers. Contemporary Physics 37, No. 5, 1996, 357-389.

BARE95 Barenco, A., Bennett, Ch. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., and Weinfurter, H.: Elementary gates for quantum computation. Physical Review A 52, No. 5, 1995, 3457-3467.

BARE96b Barenco, A., Brun, T. A., Schack, R., and Spiller, T. P.: Effects of noise on quantum error correction algorithms. Technical Report, Clarendon Laboratory, Department of Physics, University of Oxford, Oxford, United Kingdom, December 18, 1996.

BEAV95 Beaver, D.: Computing with DNA. Journal of Computational Biology 2, No. 1, 1995, 1-8.

BENN95 Bennett, C. H.: Quantum Information and Computation. Physics Today 48, No. 10, 1995, 24-30.

BERA91 Beratan, D. N., Onuchic, J. N.: Molecular implementation of molecular shift register memories. United Staates Patent, Patent No. 5.016.063, May 14, 1991.

BONE96a Boneh, D., Dunworth, Chr., Lipton, R. J., and Sgall, J.: On the Computational Power of DNA. Technical Report, Department of Computer Science, Princeton University, Princeton, New Jersey, 1996.

BONE96b Boneh, D. and Lipton, R. J.: Making DNA Computers Error Resistant. In: Proceedings of the 2nd Annual Meeting of DNA based computers, Princeton, 1996.

BROW94 Brown, J.: A Quantum Revolution for Computing. New Scientist 143, No. 1944, 1994, 21-24.

CHUA95 Chuang, I. L. and Yamamoto, Y.: Simple quantum computer. Physical Review A 52, No. 5, 1995, 3489-3496.

CLEV96 Cleve, R. and DiVincenzo, D. P.: Schumacher's quantum data compression as a quantum computation. Physical Review A 54, No. 4, 1996, 2636-2650.

DiVi95 DiVincenzo, D. P.: Quantum Computation. Science 270, No. 5234, 1995, 255-261.

DREX92 Drexler, K. E.: Nanosystems: Molecular Machinery, Manufacturing, and Computation. New York: John Wiley & Sons, Inc., 1992.

FEYN86 Feynman, R. P.: Quantum Mechanical Computers. Foundations of Physics 16, No. 6, 1986, 507-531.

FEYN61 Feynman, R. P.: There's Plenty of Room at the Bottom. In: Gilbert, H. D. (Ed.): Miniaturization. Chapter 16, pp. 282-296. New York: Reinhold Publishing Co., 1961.

FISC94 Fischer, C. M., Burghard, M., Roth, S., and Klitzing, K. von: Organic Quantum Wells: Molecular Rectification and Single -Electron Tunneling. Europhysics Letters 28, No. 2, 1994, 129-134.

GERS96 Gershenfeld, N.: Signal entropy and the thermodynamics of computation. IBM Systems Journal 35, Nos. 3 & 4, 1996, 577-586.

GERS97 Gershenfeld, N. A. and Chuang, I. L.: Bulk Spin-Resonance Quantum Computation. Science 275, No. 5298, 1997, 350-356.

GIMZ96a Gimzewski, J. K., Jung, T. A., Cubers, M. T., and Schlittler, R. R.: Scanning Tunneling Microscopy of Individual Molecules: Beyond Imaging. Technical Report, IBM Research Division, Zurich Research Laboratory, Ruschlikon, Switzerland, 2. 12. 1996.

HOPF91 Hopfield, J. J.: Molecular Shift Register based on Electron Transfer. United Staates Patent, Patent No. 5.063.417, November 5, 1991.

JOAC96 Joachim, C. and Gimzewski, J. K.: An Electromechanical Amplifier using a Single Molecule. Research Reports RZ 2846, IBM Research Division, Zurich Research Laboratory, Ruschlikon, Switzerland, 22.7. 1996.

KNIL95 Knill, E.: Approximation by Quantum Circuits. Technical Report LAUR-95-2225, Los Alamos National Laboratory, Los Alamos, New Mexico, USA, May 1995.

LLOY96 Lloyd, S.: Universal Quantum Simulators. Science 273, No. 5278, 1996, 1073-1078.

MADD85 Maddox, J.: Towards the quantum computer? Nature 316, 1985, 573.

MAHL93 Mahler, G.: Prospects of Quantum Computation. Future Electronic Devices (FED) 3, Suppl. 2, 1993, 32-37.

MAHL87 Mahler, G. and Obermayer, K.: Towards the Quantum Computer: Information Processing with Single Electrons. In: Haken, H. (Ed.): Computational Systems - Natural and Artificial. Proceedings of the International Symposium on Synergetics at Schloss Elmau, Bavaria, May 4-9, 1987. Springer Series in Synergetics, Vol. 38. pp. 154-165. Berlin: Springer-Verlag, 1987.

PERE85 Peres, A.: Reversible logic and quantum computers. Physical Review A 32, No. 4, 1985, 3266-3276.

PETT95 Petty, M. C., Bryce, M. R., and Bloor, D.: Introduction to Molecular Electronics. London: Edward Arnold, 1995.

REED93 Reed, M. A. and Tour, J. M.: Sub-Nanoscale Electronic Systems, Devices, and Processes. World Intellectual Property Organization, WO 93/25003, 9. December, 1993.

REIF95 Reif, J. H.: Parallel Molecular Computation: Models and Simulations. In: Proceedings of the 7th Annual ACM Symposium on Parallel Algorithms and Architectures SPAA'95, 1995, 213-223.

ROOS95 RooB, D. and Wagner, K. W.: On the Power of DNA-Computers. Technical Report, Theoretische Informatik, Universitat Wurzburg, February 28, 1995.

ROYC96 Roychowdhury, V. P., Janes, D. B., Bandyopadhyay, S., and Wang, X.: Collective Computational Activity in Self-Assembled Arrays of Quantum Dots: A Novel Neuromorphic Architecture for Nanoelectronics. IEEE Transactions on Electron Devices ED-43, No. 10, 1996, 1688-1699.

SHOR94 Shor, P. W.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: Proceedings of the 35th Annual Symposium on Foundation of Computer Science, Santa Fe, New Mexico, November 20-22, 1994, IEEE Computer Society Press, pp. 124-134.

SLEA96 Sleator, T. and Weinfurter, H.: Realizable Universal Quantum Logic Gates. Physical Review Letters 74, No. 20, 1996, 4087-4090.

TAUB97 Taubes, G.: Putting a Quantum Computer To Work in a Cup of Coffee. Science 275, No. 5298, 1997, 307-308.

TERM97 Terman, L. M.: The impact of device scaling limits. Applied Surface Science 117/118, 1997, 1-10.

UEYA91 Ueyama, S., Kawakubo, H., Isoda, S., and Maeda, M.: Electronic device. United States Patent, Patent No. 5.010.451, April 23, 1991.

VEDR96 Vedral, V., Barenco, A., and Ekert, A.: Quantum networks for elementary arithmetic operations. Physical Review A 54, No. 1, 1996, 147-153.