Bull. Nov. Comp. Center, Comp. Science, 17 (2002), 9–15
© 2002 NCC Publisher

Modeling of self-replication in cellular space using the Parallel Substitution Algorithm

S.M. Achasova

The Parallel Substitution Algorithm – a spatial model used to represent cellular computations – is applied to designing self-replicating structures in cellular space. The implementation of the Parallel Substitution Algorithm enables one to develop more compact (according to the number of states and the number of transition rules or substitutions) and structured programs in comparison with the proper cellular automata. As an example, a typical self-replicating loop is taken.

1. Introduction

The creation of self-replicating structures began with von Neumann's universal constructor – a cellular automaton which is capable to build a copy of itself [1]. A logical organization of the universal constructor is sufficient for self-replication. The question of the minimum logical organization necessary for self-replication was studied by Langton, who designed a simple and compact structure embedded in a cellular array, capable of self-replication only [2]. It is referred to as a self-replicating loop. Langton has stated a criterion for the true self-replication which required that a program embedded in the cellular automaton should be both used as instructions to be executed and copied as uninterpreted data. With application of Langton's criterion the construction of a copy is actively directed by the structure itself rather than being merely a consequence of the transition rules, which enables one to rule out trivial cases of self-replication.

It is known that the design of transition rules of a cellular automaton (CA) to perform a specific task is not an easy problem. Two approaches to creating self-replicating structures in a cellular space are available in the literature. The first one involves the application of genetic algorithms [3] and the second one – L-systems (Lindenmayer systems) [4, 5] to discover the CA rules governing self-replicating structures. In this paper, we introduce an extended paradigm of the CA, called the Parallel Substitution Algorithm (PSA) [6], to specify a self-replicating structure. The PSA has the following extra possibilities as compared to the classic CA: an arbitrary template of substitution, the change of states of several cells in the same substitution, functional substitutions. These enable us to develop a laconic description of self-replication and computation in cellular space.

There are two lines of application of the investigation results into self-replication. The first one is a new cellular architecture called embrionics, or embryonic electronics, endowed with properties of self-replication and self-repair [7–9]. The second line is a novel massively parallel structure capable of solving a specific class of problems while self-replicating [10, 11]. It is evident that the self-replication is one of the key themes of research into these applications.

2. A self-replicating loop

Langton's self-replicating loop is based on Codd's periodic emitter that is a storage element and timing device in Codd's universal constructor [12]. It is a square loop embedded in the 2D cellular automata space consisting of 8-state cells (Figure 1). Langton's loop is surrounded by internal and external sheaths, i.e., cells in the sheath state (in Figure 1, state "2" is the sheath state). The sheathed path serves as a means for signal propagation in the loop. Signals are represented by cells in the other states, a signal sequence being a program of self-replication.

```
 2 2 2 2 2 2 2 2
2 1 7 0 1 4 0 1 4 2
2 0 2 2 2 2 2 2 0 2
2 7 2         2 1 2
2 1 2         2 1 2
2 0 2         2 1 2
2 7 2         2 1 2
2 1 2 2 2 2 2 2 1 2 2 2 2 2
2 0 7 1 0 7 1 0 7 1 1 1 1 1 2
 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Figure 1

Replication is obtained in the following way. The construction first extends its constructing arm by six cells, next it turns left and adds other six cells. After three such turns the arm folds upon itself. When the new loop is closed, the constructing arm retracts, and the new loop starts the process over again.

The replication process in more detail is demonstrated on a simple typical self-replicating loop [4], which is obtained using L–system. It is just that loop which will be further modelled using the PSA. It is an unsheathed loop. (The point is that at a later time Langton's loop was simplified. One of simplifications is to eliminate first the internal sheath and then both of them. This brings about smaller self-replicating structures that have simpler transition functions.) The loop is embedded into a two-dimensional, 9-state, 5-neighbor cellular automata space. In Figure 2, the replication process and the letter used as states are shown.

The whole replication process consists of 44 derivation steps. The replication time is defined as the number of steps it takes for both the new loop to appear and the original loop to turn back to its initial state. The total number of the transition rules is 52.

```
o o o          s l o            o o o   s w l     o o o   o w w     s w l   o o o
o   o          s   o     o      o   o       o     l   o   o   l     o   o   o   o
l e e o o      o o o o o l n    l e e o o o o     e e o   o c b     o o o   l e e o o
 time = 0       time = 13        time = 32         time = 39          time = 44
```

Figure 2. Letters used in the replication process: **o** is a building component; **e**, **n**, **w**, and **s** are east, north, west, and south moving growth signals respectively; **l** is a left turn signal; **b** is a first branch signal; **c** is a second branch signal; and blank space is a quiscent state

3. The Parallel Substitution Algorithm

The PSA, as the CA, belongs to the class of spatially distributed dynamic system models in which many simple components locally interact to produce complex patterns of global behavior. As in the CA, time in the PSA is discrete, and space is divided into in the two-dimensional (in theory, N-dimensional) lattice of cells, representing an automaton or a processor. Each cell can be in one of n possible states synchronously updated in this paper according to a local transition rule or a substitution. A cell is represented by a pair (a, m), where a is a cell state, $a \in A$, m is a cell name, $m \in M$, A is a set of states, M is a naming set of the Cartesian coordinates in a discrete space, and in our case $m = (x, y)$. On the set M, the naming functions $\phi_i(m)$ are defined, $\phi_i(m) : M \to M$, for any $m \in M$ $\phi_i(m) \neq \phi_j(m)$. In this case, shift functions are taken as naming ones.

Operations on a cellular array are given by a set of parallel substitutions $\Phi = \{\theta_i\}$, $i = 1, \ldots, n$,

$$\theta_i : S_{i1}(m) * S_{i2}(m) \Rightarrow S_{i3}(m),$$

$$\begin{aligned} S_{i1}(m) &= \{(a_{i1}, \phi_{i1}(m)) \ldots (a_{ip}, \phi_{ip}(m))\}, \\ S_{i2}(m) &= \{(b_{i1}, \psi_{i1}(m)) \ldots (b_{iq}, \psi_{iq}(m))\}, \\ S_{i3}(m) &= \{(c_{i1}, \phi_{i1}(m)) \ldots (c_{ip}, \phi_{ip}(m))\}, \end{aligned}$$

where $\phi_{ik}(m)$ $(k = 1, \ldots, p)$ and $\psi_{il}(m)$ $(l = 1, \ldots, q)$ are naming functions, their values for any $m \in M$ must be different. A subset of the naming functions $\{\phi_{ik}(m)\}$ contains an identical function $\phi_{ij}(m) = m$, and a cell with the name m is referred to as central cell of the substitution. A set of the naming functions $\{\phi_{ik}(m)\} \cup \{\psi_{il}(m)\}$ defines a geometrical figure in the space which is called the template of the substitution. Note, that $c_{i1}, \ldots, c_{ip}$ can be both merely states and functions of states $a_{i1}, \ldots, a_{ip}, b_{i1}, \ldots, b_{iq}$. In the latter case, the substitutions are named functional.

A parallel substitution θ_i is applicable to a cellular array K if there is at least one cell with the name $m \in M$ such that $S_{i1}(m) \cup S_{i2}(m) \subseteq K$. It is executed by substituting the cells of the right-hand side S_{i3} for the cells of the base S_{i1}. The cells of the context S_{i2} remain unchanged.

A set of parallel substitutions $\Phi = \{\theta_1, \dots, \theta_n\}$ is applied to K according to the following iterative procedure. Let us assume that $K(t)$ is a cellular array being a result of execution of a set of the parallel substitutions Φ in K in t iteration steps. Further, if no substitution $\theta_i \in \Phi$ is applicable to $K(t)$, then $K(t)$ is a result of the computation, else all applicable to $K(t)$ substitutions are executed simultaneously, and $K(t)$ is transformed to $K(t+1)$, that is the result of the $(t+1)$-th iteration step.

Determinacy of a parallel computation performed according to the above synchronous procedure is provided with the property of non-contradiction of a set of parallel substitutions Φ. This property lies in the fact that the application of Φ to any K (specified over the given sets A and M) cannot give a set of cells in which there are if only two cells with the same names and different states.

The non-contradictory set of parallel substitutions Φ together with the above iterative procedure of its execution is called the parallel substitution algorithm.

4. A self-replicating parallel substitution algorithm

A PSA for self-replication of the loop, shown in Figure 2, consists of 18 parallel substitutions, 6 symbolic ones θ_1, ..., θ_6 and 12 functional ones θ_7, ..., θ_{18}. The cells require six states, designated 1, 2, 3, 4, 5, $\emptyset$. States 1 and 3 as a branch signal represent intermediate ones during the growth process. A template of each substitution is 2×2 square. Either of 12 functional substitutions takes one of four templates p_1, p_2, p_3, p_4 as its left-hand side and uses one of the five functions f_1, ..., f_5 in its right-hand side. The substitutions as geometrical figures are shown in Figure 3.

Two of the symbolic substitutions and one of the functional substitutions can be represented as the following formulas, the rest ones can be written down in a similar manner.

$$\begin{aligned}
\theta_1 :\ & \{(3,\langle x,y\rangle)\,(\emptyset,\langle x+1,y\rangle)\} * \{(5,\langle x,y+1\rangle)\,(\emptyset,\langle x+1,y+1\rangle)\} \Rightarrow \\
& \{(\emptyset,\langle x,y\rangle)\,(1,\langle x+1,y\rangle)\}, \\
\theta_4 :\ & \{(1,\langle x+1,y\rangle)\} * \{(5,\langle x,y\rangle)\,(5,\langle x,y+1\rangle)\,(4,\langle x+1,y+1\rangle)\} \Rightarrow \\
& \{(2,\langle x+1,y\rangle)\}, \\
\theta_{10} :\ & \{(v,\langle x+1,y\rangle)\} * \{(t,\langle x,y\rangle)\,(s,\langle x,y+1\rangle)\,(u,\langle x+1,y+1\rangle)\} \Rightarrow \\
& \{(f_2,\langle x+1,y\rangle)\}.
\end{aligned}$$

The substitutions θ_7, θ_8, θ_9, and θ_{10} are responsible for signal propagation through the initial loop and a finished one. The regular internal sheath

p_1 $\boxed{\begin{array}{cc} v & u \\ t & s \end{array}}$ $f_1 : v = u \text{ if } (s = 2 \vee t = 2)$

p_2 $\boxed{\begin{array}{cc} u & s \\ v & t \end{array}}$ $f_2 : v = u \text{ if } [(s = 2 \vee t = 2) \wedge u \neq 3]$

p_3 $\boxed{\begin{array}{cc} s & t \\ u & v \end{array}}$ $f_3 : v = u \text{ if } [(s = 3 \vee t = 3) \wedge v \neq \emptyset]$

p_4 $\boxed{\begin{array}{cc} t & v \\ s & u \end{array}}$ $f_4 : v = 5 \text{ if } [(s = 1 \vee t = 1) \wedge v = \emptyset \wedge u = 4]$

$f_5 : v = u \text{ if } [(s = 1 \vee t = 1) \wedge v \neq \emptyset]$

$\theta_1 : \boxed{\begin{array}{cc} 3 & \emptyset \\ 5 & \emptyset \end{array}} \longrightarrow \boxed{\begin{array}{cc} \emptyset & 1 \end{array}}$

$\theta_2 : \boxed{\begin{array}{cc} 3 & 5 \\ 5 & 4 \end{array}} \longrightarrow \boxed{\emptyset}$

$\theta_3 : \boxed{\begin{array}{cc} \emptyset & 5 \\ 5 & 5 \end{array}} \longrightarrow \boxed{\begin{array}{cc} \emptyset & 3 \end{array}}$

$\theta_4 : \boxed{\begin{array}{cc} 5 & 1 \\ 5 & 4 \end{array}} \longrightarrow \boxed{2}$

$\theta_5 : \boxed{\begin{array}{cc} 5 & \emptyset \\ 3 & \emptyset \end{array}} \longrightarrow \boxed{\begin{array}{c} 3 \\ 3 \end{array}}$

$\theta_6 : \boxed{\begin{array}{cc} 3 & \emptyset \\ 3 & \emptyset \end{array}} \longrightarrow \boxed{\begin{array}{c} 3 \\ 5 \end{array}}$

θ_7 : in p_1 do f_1
θ_8 : in p_2 do f_1
θ_9 : in p_3 do f_1
θ_{10} : in p_4 do f_2
θ_{11} : in p_3 do f_3
θ_{12} : in p_1 do f_4
θ_{13} : in p_2 do f_4
θ_{14} : in p_3 do f_4
θ_{15} : in p_4 do f_4
θ_{16} : in p_1 do f_5
θ_{17} : in p_3 do f_5
θ_{18} : in p_4 do f_5

Figure 3. Substitution templates: 1 – auxiliary internal sheath element, 2 – regular internal sheath element, 3 – branch signal and external sheath element, 4 – growth signal, 5 – building component, $\emptyset$ – quiscent state

state 2 is used for changing the direction of signal propagation. The substitution θ_{11} is responsible for signal propagation through the constructing arm for which the external sheath state 3 is used. The substitutions θ_{12}, θ_{13}, θ_{14}, and θ_{15} are responsible for the growth of a new loop which goes on with an auxiliary internal sheath state to be equal to 1. The substitutions θ_{16}, θ_{17}, and θ_{18} are responsible for signal propagation through a loop under construction and that proceeds also with the auxiliary internal sheath state 1. The substitution θ_1 starts to build a new loop with inserting an auxiliary internal sheath element into the center of a new loop as the first component of the loop. The substitution θ_2 partially removes a bridge between the loops. The substitution θ_3 inserts the branch signal for growing the constructing arm and removes fully the bridge between the old loop and

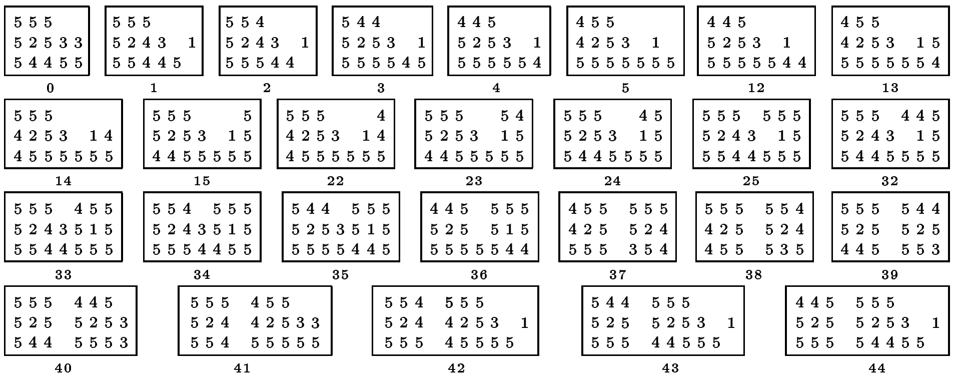

Figure 4

the new one. The substitution θ_4 replaces the regular internal sheath state 2 for the auxiliary internal sheath state 1. The substitutions θ_5 and θ_6 build the new constructing arm.

The self-replication process to be governed by the PSA is shown in Figure 4, the quiescent state designated by $\emptyset$ is represented by a blank space in this figure.

The expressive capabilities of the PSA enable one to reduce the number of states from 9 to 6 and the number of the transition rules from 52 to 18, which is the least number of transition rules for many known self-replicating loops [13]. Note, that introduction of the sheath states into the self-replicating loop does not result in increased complexity of the algorithm. And moreover, we do not know a self-replicating loop which would have a number of transition rules to be equal to or less than 18.

References

[1] Neumann J. Theory of self-replicating automata / Edited and completed by A.W. Burks. – Urbana: University of Illinois Press, 1966.

[2] Langton C.G. Self–reproduction in cellular automata // Physica D. – 1984. – Vol. 10. – P. 135–144.

[3] Lohn J.D., Reggia J.A. Automatic discovery of self-replicating structures in cellular automata // IEEE Transactions of Evolutionary Computations. – 1997. – Vol. 1, № 3. – P. 165–178.

[4] Stauffer A., Sipper M. On the relationship between cellular automata and L-systems: the self-replication case // Physica D. – 1998. – Vol. 116. – P. 71–80.

[5] Stauffer A., Sipper M. Modeling cellular development using L-systems // Lect. Notes in Comput. Sci. – 1998. – Vol. 1478. – P. 196–205.

[6] Achasova S.M., Bandman O.L., Markova V.P., Piskunov S.V. Parallel Substitution Algorithm. Theory and Application. – Singapore: World Scientific, 1994.

[7] Mange D., Stauffer A., Tempesti G. Embryonics: a macroscopic view of the cellular architecture // Lect. Notes in Comput. Sci. – 1998. – Vol. 1478. – P. 174–184.

[8] Mange D., Stauffer A., Tempesti G. Embryonics: a microscopic view of the molecular architecture // Lect. Notes in Comput. Sci. – 1998. – Vol. 1478. – P. 185–195.

[9] Prodan L., Tempesti G., Mange D., Stauffer A. Biology meets electronics: the path to a bio–inspired FPGA // Lect. Notes in Comput. Sci. – 2000. – Vol. 1801. – P. 187–196.

[10] Tempesti G. A new self-reproducing cellular automaton capable of construction and computation // Lect. Notes in Comput. Sci. – 1995. – Vol. 929. – P. 555–563.

[11] Chou H.-H., Reggia J.A. Problem solving during artificial selection of self-replicating loops // Physica D. – 1998. – Vol. 115. – P. 293–312.

[12] Codd E.F. Cellular Automata. – New York: Academic Press, 1968.

[13] Reggia J.A., Armentrout S.L., Chou H.-H., Peng Y. Simple systems that exhibit self-directed replication // Science. – 1993. – Vol. 259. – P. 1282–1287.