Bull. Nov. Comp. Center, Comp. Science, 21 (2004), 1–8

PSAs for self-replication of a cellular rectangular loop

S.M. Achasova

Abstract. The Parallel Substitution Algorithm (PSA) is a spatial dynamical system used to represent processes in cellular space. The PSA is applied to designing programs of self-replication in cellular space. The programs become more laconic (by the number of transition rules or substitutions) in comparison with the ones based on the classic cellular automata model. In this paper, there are presented two PSAs for modelling processes of self-replication of a rectangular loop in a cellular array. The first PSA simulates the self-replication process of the loop in one direction – from left to right. The second one simulates it in two directions – from left to right and from bottom upwards.

1. Introduction

There were two basic events in the research into self-replicating structures. The first one is von Neumann's development of a universal constructor – a cellular automaton (CA) capable of building any other automaton its description given, and therefore a copy of itself, when its own description is given [1]. The second one is Langton's development of a self-replicating loop. Langton's loop is a cellular automaton incapable of universal construction, but capable exclusively of self-replication [2]. There was stated a criterion for the true self-replication which required that an information embedded in the cellular automaton should be both used as instructions to be executed and copied as uninterpreted data. Application of the criterion makes it possible to rule out trivial cases of self-replication, when "self" is not the key notion. That is, constructing a copy is not directed by the initial structure itself but merely by a consequence of the transition rules.

Although Langton's loop is a very simple structure in comparison with von Neumann's universal constructor, it nevertheless has several hundreds of transition rules. There were developed more simple variants of the loop, but they had some tens of transition rules [3–5]. In this paper, we use an extended paradigm of the CA – Parallel Substitution Algorithm (PSA) [6] to simulate a self-replicating loop. The PSA has the following extra capabilities as compared to the classic CA: an arbitrary template of substitution, a change in the states of several cells in the same substitution, functional substitutions. These enable us to develop laconic descriptions of self-replication in cellular space.

At the present time, there are some lines of investigations into self-replication. Among them is creating of the novel massively parallel structures capable of solving some calculating problems while self-replicating [7–9]. Such structures are referred to as "useful replicators". In this case, self-description, i.e., a program of self-replication, is supplemented with a calculating program. And as long as a program of self-replication is endowed with additional functions, it is convenient to represent a program of self-replication in a compact form. The PSA notation allows us to make it.

2. The Parallel Substitution Algorithm

The PSA is a model for spatially distributed computations, in which many simple components locally interact to produce complex patterns of global behavior. Time and space in the PSA are discrete. Here, space is two-dimensional (in theory, N-dimensional). It is divided into a lattice of cells, representing an automaton or a processor. Each cell can be in one of n possible states, synchronously updated according to a local transition rule or a substitution. The left-hand side of a substitution defines an applicability condition and consists of the two parts – the base and the context. The right-hand side of a substitution gives the new states of the base cells. A substitution does not change states of the cells of its own context. New states of the base cells can be either merely states from a set of the possible ones (in such case, substitution is called symbolic) or functions of states (then substitution is called functional). Substitutions can have an arbitrary template (a geometrical figure) in a cellular array.

All substitutions, which are applicable to a cellular array in a clock cycle are simultaneously executed. The same cell can at once belong to the bases of two substitutions. If its new state is different in the right-hand sides of the substitutions then contradiction arises. A parallel substitution algorithm must contain a non-contradictory set of substitutions. Conditions for non-contradiction are determined in [6].

3. The 1D PSA for a self-replicating loop

With a PSA a simple typical self-replicating loop [4] is simulated. The loop being 3×3 square is embedded into two-dimensional cellular automata space. The cells require nine states (o is a building component; e, n, w, 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 quiescent state). In Figure 1, the replication process is partially shown. The whole replication process consists of 44 derivation steps. The replication time is defined as the number of steps it takes for both a 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***
Step 0	Step 13	Step 32	Step 39	Step 44

Figure 1

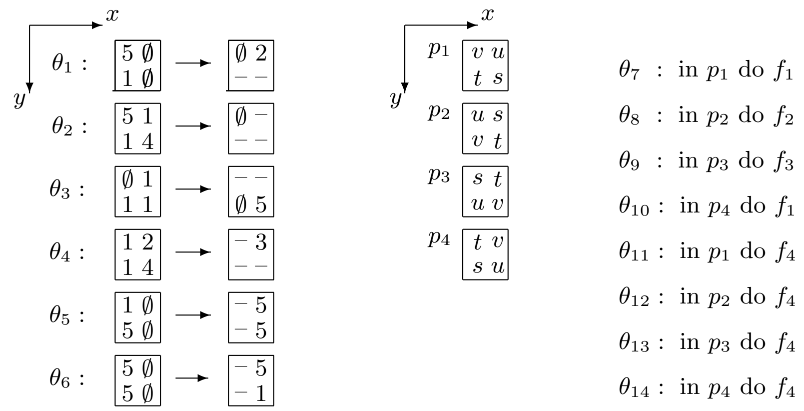

Figure 2

The PSA for a self-replicating loop consists of 14 parallel substitutions, six symbolic $\theta_1, \ldots, \theta_6$, and eight functional $\theta_7, \ldots, \theta_{14}$ (Figure 2). The cells require six states.

State 1 is a building component. Instead of four growth signals in the prototype loop there is only one State 4 in the PSA. For signal propagation through the loop, the internal sheath is used. With given proportions of the loop it is formed of one cell and the cell has State 3. The sheath is named the regular internal sheath. For the direction of a new loop growth and signal propagation through a new fragment of the loop, an auxiliary internal sheath being formed of the cell State 2 is used. When constructing a daughter loop comes to the end, and the auxiliary sheath is changed for the regular one, i.e., State 2 is changed for 3. The different states of the internal sheath are used to exclude contradiction while executing substitutions. State 5 is a growth signal for a constructing arm and, moreover, the same state forms a small fragment of an external sheath placed above the constructing arm and used for the direction of signal propagation through the constructing arm. And, finally, $\emptyset$ is the quiescent state.

The template of every substitution is 2×2 square. The symbolic substitutions have dashes in the right–hand sides of the substitutions corresponding to the context cells, which are to say their states do not change during execution of a proper substitution. Either of eight 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 four functions f_1, …, f_4 in its right-hand side:

$$
\begin{array}{lll}
f_1: & v = u, & \text{if } [(s = 2 \vee s = 3 \vee t = 2 \vee t = 3) \wedge v \neq \emptyset \wedge u \neq 5] \\
f_2: & v = u, & \text{if } (s = 3 \vee t = 3) \\
f_3: & v = u, & \text{if } [(s = 2 \vee s = 3 \vee s = 5 \vee t = 2 \vee t = 3 \vee t = 5) \wedge v \neq \emptyset] \\
f_4: & v = 1, & \text{if } [(s = 2 \vee t = 2) \wedge v = \emptyset \wedge u = 4]
\end{array}
$$

The whole set of the substitutions is divided into five subsets according to its purposes. The substitutions $\theta_7, \ldots, \theta_{10}$ answer for the signal propagation through the loop. The substitutions $\theta_{11}, \ldots, \theta_{14}$ answer for the growth of a new loop. The substitutions θ_1, θ_4 are responsible for the construction of the internal sheath. The first one builds an auxiliary internal sheath of the loop to be, that is the first step of construction of a daughter loop. The second substitution replaces the auxiliary internal sheath with the regular one. The substitutions θ_2, θ_3 remove the bridge between the old and the new loops and, moreover, θ_3 inserts the growth signal for the constructing arm into a loop under construction. The substitutions θ_5, θ_6 build a new constructing arm.

The self-replication process to be governed by the 1D PSA is shown in Figure 3, the quiescent state $\emptyset$ being represented by a blank space in this figure.

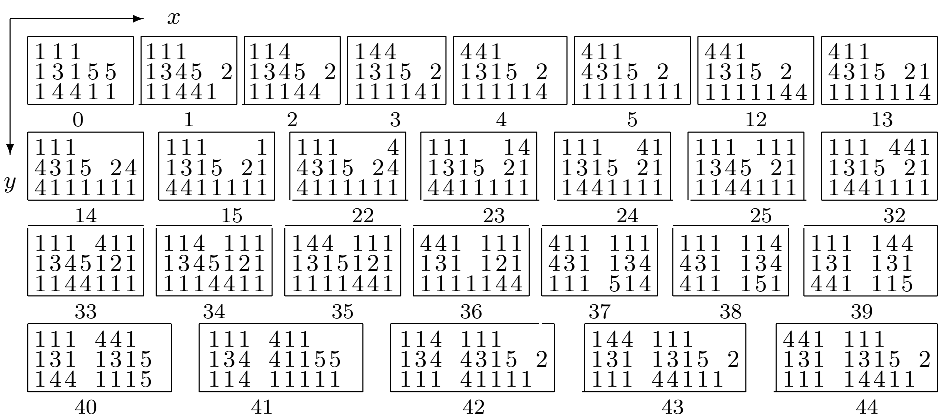

Figure 3

Note, that introducing the sheaths into the self–replicating loop does not bring about an increased complexity of the algorithm. Moreover, the expressive capabilities of the PSA enable one to reduce the number of states from nine to six and the number of the transition rules from 52 to 14 as compared to the initial cellular automaton [4].

One additional remark should be done. It is possible to combine the substitutions θ_3, θ_4 into a new substitution $\theta_{3,4}$ and θ_5, θ_6 into $\theta_{5,6}$. The templates of the new substitutions are 2×3 rectangles. The modified algorithm has only four symbolic substitutions and the same functional sub-

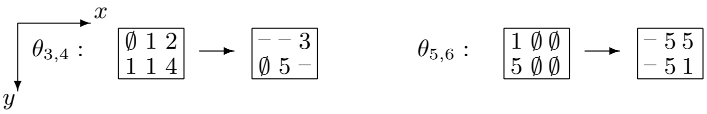

Figure 4

stitutions, and in all 12 substitutions. The substitutions $\theta_{3,4}$ and $\theta_{5,6}$ are shown in Figure 4.

4. The 2D PSA for a self-replicating loop

A PSA, governing the self-replication process of a loop in two directions (from left to right and from bottom upwards) is carried out on the basis of a 1D PSA. Two new cell states are added to six states. Namely, state *6* forms an auxiliary internal sheath which is used for the growth of the loop from bottom upwards, and State 7 is a a growth signal for a vertical constructing arm and, moreover, the same state forms a small fragment of an external sheath which is used for the construction of the loop from bottom upwards. 2D PSA has 20 parallel substitutions: twelve symbolic $\theta_1, \ldots, \theta_{12}$, and eight functional $\theta_{13}, \ldots, \theta_{20}$ (Figure 5).

The symbolic substitutions from 1D PSA, θ_1, θ_2, θ_3, θ_5, θ_6, appear in 2D PSA in the same form. The substitution θ_4 of 2D PSA introduces the growth signal for a vertical constructing arm in the loops of the ground row. The substitutions $\theta_7, \ldots, \theta_{12}$ of 2D PSA execute the same operations for replication of a loop from bottom upwards which are executed by the substitutions $\theta_1, \ldots, \theta_6$ of 1D PSA for the growth of a loop from left to right. The substitutions θ_8, θ_9 of 2D PSA remove the vertical bridge and,

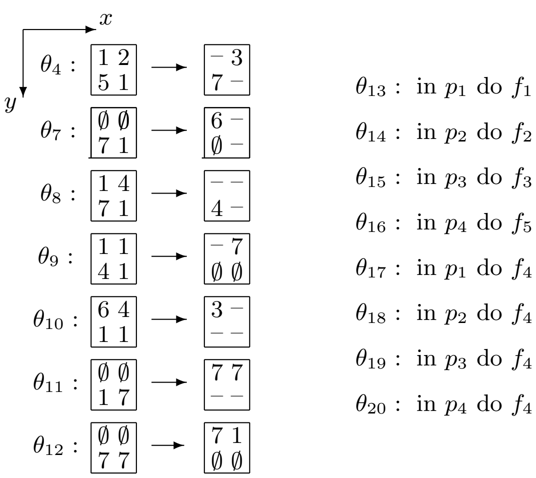

Figure 5

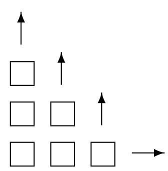

Figure 6

moreover, θ_9 inserts the growth signal for the vertical constructing arm into the loop under construction. It should be noted that the loops of the only ground row are replicated in two directions. The loops of the rest rows are replicated from bottom upwards. So, the growth front moves along the square diagonal, as is shown in Figure 6.

The functional substitutions of 2D PSA are responsible for the signal propagation through the loop and for the growth of a new loop. Either of eight functional substitutions takes one of the same templates $p1$, $p2$, $p3$, $p4$ as its left-hand side and uses one of five functions $f_1, \ldots, f_5$ in its right-hand side:

$$\begin{aligned}
f_1 : v = u, \text{ if } & [(s = 2 \vee s = 3 \vee s = 6 \vee t = 2 \vee t = 3 \vee t = 6) \wedge \\
& v \neq \emptyset \wedge u \neq 7] \\
f_2 : v = u, \text{ if } & [(s = 3 \vee s = 6 \vee t = 3 \vee t = 6) \wedge v \neq \emptyset] \\
f_3 : v = u, \text{ if } & [v \neq \emptyset \wedge (((s = 5 \vee t = 5) \wedge u \neq 7) \vee \\
& (s = 2 \vee s = 3 \vee t = 2 \vee s = 3))] \\
f_4 : v = 1, \text{ if } & [(s = 2 \vee s = 6 \vee t = 2 \vee t = 6) \wedge v = \emptyset \wedge u = 4] \\
f_5 : v = u, \text{ if } & [(s = 2 \vee s = 3 \vee s = 6 \vee s = 7 \vee t = 2 \vee t = 3 \vee t = 6 \vee \\
& t = 7) \wedge v \neq \emptyset \wedge u \neq 5]
\end{aligned}$$

The self-replication process to be governed by the 2D PSA is shown in Figure 7.

The 2D PSA, like the 1D PSA, can be made more laconic. For this purpose, each of the three pairs of substitutions θ_5, θ_6, θ_9, θ_{10} and θ_{11}, θ_{12} combines into one, $\theta_{5,6}$, $\theta_{9,10}$, $\theta_{11,12}$, respectively. The substitutions θ_3, θ_4 of the 2D PSA cannot be combined because of the evident contradiction of such a union. The new substitutions are shown in Figure 8. So, the 2D PSA can have 17 parallel substitutions,among which are nine symbolic and eight functional.

5. Conclusion

It is demonstrated that using the PSA the programs of self-replication of a rectangular loop in cellular space can be very simple in comparison with the classic cellular automata [2–5]. The program of self-replication of the loop in one direction (from left to right) has 14 or, in another variant, 12 parallel substitutions against 52 in the prototype loop [4]. The program of self-replication of the loop in two directions (from left to right and from bottom upwards) has 20 or, in another variant, 17 parallel substitutions. These programs can be further used for the construction of a useful replicator.

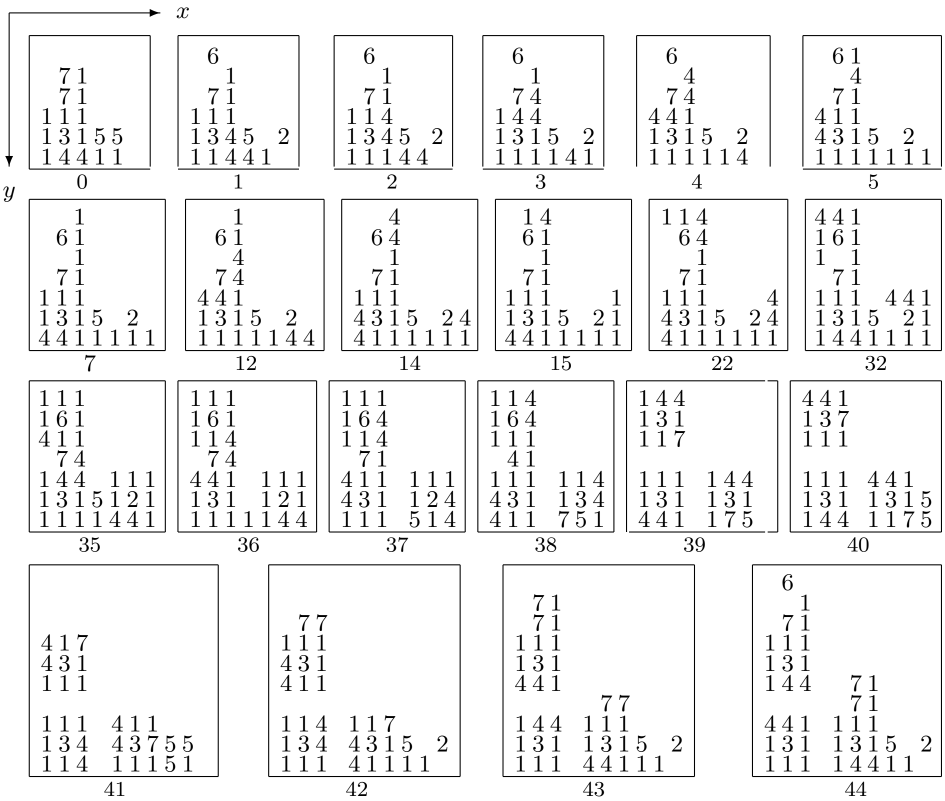

Figure 7

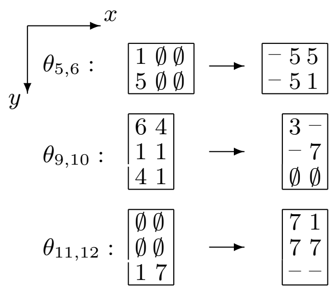

Figure 8

References

[1] von 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] Byl J. Self-reproduction in small cellular automata // Physica D. – 1989. – Vol. 34. – P. 295–300.

[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] 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.

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

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

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

[9] Petraglio A., Henry J.-M., Tempesti G. Arithmetic operations on self-replicating cellular automata // Lect. Notes in Artif. Intell. – 1999. – Vol. 1674. – P. 447–456.