Self–reproducing self–assembling evolutionary automata*

Jiří Wiedermann
Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
e–mail jiri.wiedermann@cs.cas.cz

September 13, 2004

Abstract

We introduce a computational model of a so–called globular universe which represents generalization of both classical cellular automata and contemporary models of self–assembly. Similarly as the latter mentioned model our model utilizes a multiset of globules which are endowed by self–organizing ability controlled by a finite state mechanism; these computational units are not fixed in a predetermined structure. The environment abounds in these units which are available at places where needed for a self–assembly of various objects. Within a globular universe we define the notion of self–reproducing evolutionary automaton. This notion refers to an automaton being at the beginning of a lineage of self–reproducing automata which leads to self–reproducing automata with arbitrary complex finite state control mechanisms via a series of mutations of intermediate automata. The ideas presented in this paper complement von Neumann's results on self–reproducing automata in a static universe by offering a precise definition of what is meant by "evolutionary self–reproduction" and by designing a dynamic nondeterministic universe with a self–reproducing self–assembling evolutionary automaton.

1 Introduction

It seems that all experience of mankind points to the fact that machines can produce only simpler machines than the original ones. The only exception have been living beings if considered as machines: not only can an organism produce an almost genuine copy of itself but, as Darwin has shown, in a long run, in an evolutionary process these "machines" keep improving. It was von Neumann who in late 1940 started a systematic quest for a logical, rather than material,

*This research was partially supported by grant No. 1ET100300419 within the National Research Program "Information Society".

basis of biological self–reproduction. He first proposed a mechanistic model. It consisted of a "robot" operating in a sea of its own spare parts. The robot had some elementary functions for moving around, identifying and collecting the required parts and assembling them together and possessed a tape with instructions for building a copy of itself by making use of these elementary functions. After constructing a replica of itself, the robot finally copied its instruction tape and inserted it into the replicated robot which could then start the same activities. By this design, it is generally agreed that von Neumann discovered basic principles for the process of biological self–reproduction. Namely, there had to be a program, instruction sequence to be used in two different ways: (1) to be interpreted as instructions for constructing an offspring, and (2) to be copied passively, without being interpreted. Quite understandingly, von Neumann was not able to construct a working model of his mechanistic self–reproducing automaton which would represent a convincing proof of soundness of his design idea. However, in 1953, following Stanislaw Ulam's vision of cellular automata, he invented a cellular automaton implementation of his mechanistic model. His cellular "robot" made use of a cellular automaton with 29 states per cell, consisted of approximatively 200 000 cells and its description took more than 200 pages [6]. Thus the topic of self–reproduction entered the field of the automata theory and since then cellular automata have been used in numerous applications and variations. Interestingly, in the contemporary automata theory within the computer science, only a limited attention has been paid to the self–reproducing issue. Nevertheless, this issue has migrated into the field of artificial life where it continues to be a subject of a vivid development. In this context, especially two questions are of interest. First, what exactly was the problem to which von Neumann gave his answer? And second, how good this answer was?

As far as the first question is concerned, von Neumann did not formulate the problem of self–reproduction for cellular automata in such a way that it would show that his solution is a satisfactory and complete answer to that problem. This is because constructing an automaton, which is merely able to produce its own copy, is interesting especially in a real world but less in the general framework of cellular automata. Namely, as pointed out by several researchers (cf. [3] for an extensive discussion of this problem), each cell of a cellular automaton can "reproduce itself" by properly initializing a cell in its immediate neighborhood. This cell can even posses universal computational properties, i.e., it can act as a motile processor of a single–tape universal Turing automaton working over a linear array of cells holding the input data. Thus, universality alone is not a good enough reason for devising a self–reproducing cellular automaton of an immense complexity as von Neumann did. Perhaps, von Neumann was aware of this since in his *Theory of Self–Reproducing Automata* ([6], p. 92) he asked: *"Can the construction of automata by automata progress from simpler types to increasingly complicated types? Also, assuming some suitable definition of "efficiency", can this evolution go from less efficient to more efficient automata?"* Nowadays it appears that the final aim of von Neumann's design was not the universality but the "self–improvement" issue of a self–reproducing automaton which, however, was not settled at all by von Neumann. As far as

the second question is concerned, in the past there were several trials to refine the von Neumann design of a self–reproducing automaton (cf. [3]). Nevertheless, in the absence of a precise task definition it has been difficult to compare the alternative solutions w.r.t. the given task and to judge in what sense the alternative solutions were better. In these efforts an evolution has still not been the main issue and that is why several authors, inclusively e.g. Herman [1] and recently also McMullin [3] called for a mathematical definition of the "evolutionary growth of complexity" (as it is called in [3]) of self–reproducing automata.

In this paper we formulate a new mathematical model of self–assemblage which is a generalization of both classical cellular automata and contemporary models of self–assemblage. This model allows to introduce nondeterminism into the self–assemblage process. Within this model we give a mathematical definition of a self–reproducing evolutionary automaton which captures essential aimes of von Neumann's design: the requirement of self–reproduction and the possibility of evolutionary changes leading *"from simpler types to increasingly complicated types",* as von Neumann put it. In our definition, the notion of "efficiency" of an automaton is measured by the minimal number of states of an equivalent finite control mechanism. Finally we describe nondeterministic globular universe and a self–reproducing self–assembling evolutionary automaton existing within this universe. This automaton is conceptually much simpler than von Neumann's proposal and includes a "built–in" nondeterministic evolutionary mechanism which leads directly to the infinite evolutionary lineages of automata with increasingly complex finite state control.

The structure of the paper is as follows: in Section 2 the globular universe is defined. Section 3 shows the computational equivalency of the model with cellular automata and Turing machines. Next Section 4 defines the notion of a self–reproducing evolutionary machine. Finally, Section 5 describes the design of a nondeterministic globular universe in which a self–reproducing evolutionary globular automaton exists. Conclusions in Section 6 mention possible avenues for a further research.

2 Globular universe

The basic particles possessing latent self–assembly properties in which our universe abounds and from which all our ensembles will be constructed are computational units called *globules*. As their name suggests globules have a shape of tiny balls. Each globule can be seen as an "embodied" finite automaton that can find itself in one of the finite number of states. These states determine the self–assembly properties of the globules, i.e., their abilities to bind with other globules. The globules move freely in an empty 3D space — except of the globules and their complexes there is nothing else in the space. They are subject to a random motion, occasionally colliding and possibly binding one with each other. The behavior of globules on this occasion is determined by their state at that moment and is described by an *interaction function*. This function de-

termines new states of the globules and their behavior after their interaction. Prior to giving a formal definition which abstracts the previously given informal view of a globular universe we will introduce few preliminaries.

Let $\mathcal{S}$ be a geometrical sphere of a unit radius in a 3D Euclidean space. A *contact domain*, or simply a *contact* on the surface of $\mathcal{S}$ is defined as the interior of any closed convex curve on its surface; the curve itself is a part of the contact domain. Each contact domain has a uniquely determined position on the surface w.r.t. the given coordinate system. We will consider only such contact domains whose boundaries are computable — i.e. there is a Turing machine algorithm that will "draw" the domain at the required position on the surface of $\mathcal{S}$. An example of a contact domain is a single point on a sphere, or a "belt" drawn around the "equator" of a sphere, a spherical circle drawn at a sphere's pole, etc. We will consider spheres with a finite number of contact domains on their surface; domains can overlap. Each domain is endowed by certain properties which are characterized by the color of the domain, and by the affinity of the domain. We will see that the domains represent the mechanism underlying the self–assembly properties of globules. We define the respective notions more formally.

Definition 2.1 *A* globular configuration space *is the set* $C = \{D_1, \ldots, D_k, \Sigma, \Gamma, \pi\}$, *where*

- $D_1, \ldots, D_k$, *for* $k > 0$, *are* contact domains, *or* contacts *defined on the surface of a unit geometrical sphere;*
- Σ *is the finite alphabet of* contact domain colors*;*
- $\Gamma = \{neutral, join\}$ *is the set of* affinities [1]*;*
- $\pi : \{D_1, \ldots, D_k\} \rightarrow 2^{\Sigma \times \Gamma}$ *is a function that for each contact domain defines the admissible set of* contact properties.

Any element of $(\Sigma \times \Gamma)^k = \Omega$ *is called a* configuration *(or a* state*, respectively) of a globule from space* C.

The state of a globule g can change when g interacts with another globule h. We say that g interacts with h if and only if both globules come into a contact, or if they are already in a contact — one with the other and one of the two changes its state. If more than two globules interact simultaneously this parallel multiple interaction is broken into a randomly ordered sequential series of pairwise interactions. The configuration changes, during an interaction, are governed by a so–called *interaction function* $\Phi : \Omega^2 \rightarrow \Omega^2$. Let $c_g, c_h \in \Omega$ be the states of g and h at the time of their interaction. Then $\Phi(c_g, c_h) = (c'_g, c'_h)$ is to be read as *"after the interaction of* g *with* h*, the state of* g *changes from* c_g *to* c'_g *whereas the state of* h *changes from* c_h *to* c'_h*"*.

[1] We can also consider a richer set of affinities, e.g. of form $\Gamma = \{neutral, attract, join, repel\}$ or affinities of a variable strength, given by an integer number; then the priorities among the affinities must be stated similarly as in the case of a self–assembly model introduced in [4].

At the interaction times the behavior of the globules is governed by the domain colors and the affinities. Consider two contact domains D_i and D_j with $\pi(D_i), \pi(D_j) \subseteq \Sigma \times \Gamma$ and two globules $g, h \in C$. Let $(\sigma_g, \gamma_g) \in \pi(D_i)$ and $(\sigma_h, \gamma_h) \in \pi(D_j)$ be the contact properties of the respective globules at these contacts immediately *after* the interaction, i.e., after the application of Φ. When both colors and affinities match, i.e., $\sigma_g = \sigma_h$ and $\gamma_h = \gamma_h = \gamma$ then the globules will behave as follows:

- if $\gamma = \mathit{neutral}$, then g and h are neither attracted to nor repelled from each other via contacts D_i and D_j; that is, both globules will not be joined via the respective contacts;
- if $\gamma = \mathit{join}$, then g and h will join one with each other via a *bond* established between the contacts D_i and D_j.

In all other cases the globules will behave as if both had a neutral affinity.

A bond between two globules can only be established when the globules have not been already bonded to other globules in a position which prevents both globules to enter a position allowing touching the contact domains needed for a new bond. If a new bond can be established then both globules stabilize in a relative position satisfying the needs of all existing bonds. If there are more possibilities for such a positioning, one of them is chosen randomly.

Note that if contact domains are not point domains (i.e., if they have a positive area), then they can make a "movable" bond between any points within their contact areas. This gives some freedom to globules which can for instance self–assemble into an elastic structure with a variable curvature. For instance, globules with an attracting contact domain in form of a belt around their equators can self–organize into a ball–shaped structure of a certain minimal and maximal radius.

Summarizing formally our previous notions, we arrive at the following definition of a globular universe:

Definition 2.2 A globular universe $\mathcal{U} = (C, \Phi)$ *is a multiset of globules and their ensembles with globules from the globular configuration space C interacting via the interaction function Φ.*

Note that for some arguments Φ need not be defined. That is, in fact Φ determines what kind of globules may interact in $\mathcal{U}$ and whether and how the properties of globules will change on this occasion. For instance, it need not be the case that the state of a globule can be changed into any other state or that an originally neutral globule will ever bind to some other globule via some sequence of interactions.

Next we will aim towards a definition of an evolution within a globular universe. The idea is to see the evolution as a series of "snapshots" taken at interaction times and documenting in this way the interactions of objects within the universe. In the sequel we will consider only globular universes with a finite number of finite globular objects, with a potentially infinite supply of

globules. The objects can consist of globules from the entire configuration space. However, the configuration space of the supplied globules may be restricted. Such globules will be said to arrive from the *environment* while the objects under consideration will be said to find themselves in an *observable part* of the universe. Once arriving from the environment and interacting with an object within the observable universe a globule will become a part of this universe unless stated otherwise. The globules arriving from the environment into the observable universe will also be seen as an *input* into the observable universe.

A *universe configuration* reflects the static aspects of a situation in the observable universe at an interaction time. These aspects include the description of all objects within the universe and their spatial relations. The dynamic aspects, i.e., the movements of objects leading to their collisions and state changes of globules are captured through the interactions among the objects.

A globule represents a basic object. All other globular objects will essentially be self–assembled objects formed by multisets of globules. In general, a *description of an object* O is given by an adjacency graph where its nodes correspond to the globules and its edges to the bonds among the globules. Of course, the mapping between the nodes and the globules and that between the edges and the bonds must be included in this definition. However, in many cases we will deal with objects having a regular structure leading to their simpler representation. In such a case an object O will be defined by

- its *size*, i.e., No. of elements in the underlying multiset from which it is composed;
- the subset $S \subseteq \Omega$ of admissible states of globules potentially creating that multiset;
- its spatial organization which is described as a computable invariant (predicate) that holds for all globules in O and captures their adjacency relations with their neighbors, i.e. it captures in fact bonds among the respective globules (see examples of globular objects in the sequel).

The *spatial relations* among the globular objects are described with the help of predicates which hold for all globules within the objects. Such a predicate could be e.g. a binary predicate TOUCHES(A,B) with the meaning *"object A touches object B"*. Other predicates could be e.g. OUTSIDE, INSIDE, NEXT_TO and the like. In fact, all our predicates that will be used in the sequel will be computable in a polynomial time w.r.t. the size of the objects involved.

Let $\mathcal{E} \subseteq \mathcal{U}$ be the multiset of globules in the environment, O_t be the set of all objects in the observable part of $\mathcal{U}$ at an interaction time t, and let R_t be the set of spatial relations holding among the objects in O_t at that time. The *(observable) universe configuration* $C_t = \{O_t, R_t\}$ at time t is described by the description of all globular objects in the universe and relations among them at time t. Let INTERACT_t be the set of all interactions to be realized over objects at time t. This set is given by the list of pairs of interacting globules

at that time. To each globule its state and membership in $\mathcal{E}$ or to an object in O_t must be stated. Thus, an interaction can occur either between two globules from the environment or between a globule from the environment and another one from an object, or between two globules within the same object or within different objects. The evolution starts at time $t = 0$ in a certain initial universe configuration C_0. The set of interactions $\mathrm{INTERACT}_0$ is then applied to the objects in O_0 to get C_1. In this way the evolution proceeds by applying interactions to objects within configurations at interaction times t_i, for $i = 1, 2, \ldots$ Note that due to the fact that dynamic information about globular objects is not captured in a configuration and the states of globules arriving from $\mathcal{E}$ are in general unknown beforehand, the interactions among the objects cannot be computed (e.g. by using Newton laws) and thus the entire evolution cannot be determined from knowing the initial configuration only (as it is the case with a classical cellular automaton). Under such a scenario an evolution is seen as an on–line interaction between the objects and elements from $\mathcal{E}$ (if any) representing the input into the observable part at that time. As a result, applying $\mathrm{INTERACT}_t$ onto objects in C_t we get C_{t+1}. Also note that by transiting from C_t to C_{t+1} not only objects but also their spatial relations may change.

Next we will describe basic globular objects which we will need in the sequel.

A globule in a given state presents the simplest object. Interactions among globules lead to emergence of more complex objects. For instance, after a collision of two globules, g and h, respectively, a pair $g.h$ can emerge if the interaction function is such that it results in joining g to h.

A simple globular object with a regular rigid structure is a *grid*. It is a two–dimensional rectangular array with globules in identical initial states residing in the array's cells. Each globule has 4 neighbors (in the north, south, east, and west direction) to which it is bonded once for all times and with which it can interact and change their state.

A useful globular object is a *strand*. It is a linear string of globules concatenated via bonds. For each non–empty strand its first globule is defined and for each globule in a non–empty strand, except the last one, its successor is defined. The important operation over strands is an operation of extending a strand by a globule; this globule is added behind the last globule of the strand. Another operation is a copy operation; this operation will be described in Section 4. A strand with both ends joined together is called a *ring*.

So far we considered in essence a deterministic globular universe: thanks to the deterministic definition of the interaction function, from a given configuration and the set of interactions to be performed over the objects described by that configuration at that moment, the next configuration is computable in a unique way (the notable exception could be "random serialization" of multiple collisions). It is obvious that similarly as in the case of classical cellular automata probabilism could also be introduced into the model (e.g. in order to model "mutations"), and also non–determinism. As we will see later, the latter option is particularly convenient because it allows considering situations, which are in principle possible without bothering much about their probabilities. In this case, instead of an interaction function, we will consider a *nondeterministic*

interaction relation giving a finite number of possible outcomes for each interacting pair of globules. In our considerations we will then always say explicitly which "branch" of a nondeterministic development is to be used. In analogy with the standard nondeterminism known from the automata theory, we will consider that branch which will lead "where we need", i.e., in the case of self–reproducing automata to the self–reproduction of the automaton at hand, in the case of evolving automata their evolution, etc.

Similarly, we will also consider nondeterministic universes in the following sense: if the spatial constellation of existing objects leaves an access path free for a globule to come from the environment, a globule in a required state will "come flying" from the environment if available in $\mathcal{E}$. That is, from among all globules, which can in principle arrive, the one "we need" will arrive, indeed. This is a similar condition as considered, e.g., in the tile assembly model where tiles are available when and where needed. A *nondeterministic universe* will be a universe with a nondeterministic interaction relation and nondeterministic arrival of globules. Considering such universes enables concentrating on self–assembly aspects of interactions.

Within any globular universe, by interaction of globules various complexes of self–assembled globules can arise. We will be particularly interested in self–replicating assemblies arising from certain initial universe configurations in certain universes. Prior to submerging into the related problems we will briefly study the power of our model.

3 Globular Universe, Cellular Automata and Self–assemblage

First of all, we show how one–dimensional cellular automata can be simulated within a globular universe. For such a purpose we use the grid structure described in the previous section. The respective globules will have four bonding contacts spread equidistantly along their equators and each globule will simulate one cell of our cellular automaton. It is obvious that starting from the "seed" globules containing the input to the cellular automaton, any cellular automaton can be simulated by such a grid.

The question of the reverse simulation of a globular universe on a cellular automaton is a little more complicated. W.l.o.g. we can consider a Turing machine in place of a cellular automaton. For a Turing machine it is possible to keep on its tapes the representation of observable universe configurations and to realize the respective operations needed for updating these configurations. To do this in the finite control of a Turing machine at hand the complete table describing the interaction function of the simulated cellular automaton must be stored. The list of interactions to be performed over the objects represented on the machine's tapes will appear at the machine's input after processing the previous list of such operations. In this way the simulation can proceed as needed.

As far as simulations of models of self–assembly are concerned, the situation is as follows. There seems to be no "referential" model of self–assemblage. Therefore we will concentrate onto the elementary model, so–called tile assembly model considered, e.g., in [4]. This model consists of tiles which possess pre–defined self–assembly properties. These properties are described with the help of interaction strength assigned to the sides of rectangular sides; the sides bind when the total interaction strength exceeds a given parameter. This parameter corresponds to the affinity strength in our model (see the footnote in the previous chapter) and therefore can be simulated by our model when using globules with four contact points equidistantly placed along the globule's equator instead of the square–shaped tiles. It is clear that the globular universe presents generalization of the tile assembly model.

4 Self–reproducing evolving automata

For the definiteness of our subsequent discussion we will first define the notion of a self–reproducing automaton in a globular nondeterministic universe. In what follows we will always consider universe $\mathcal{U}$ with an observable part $\mathcal{P} \subseteq \mathcal{U}$ with the environment $\mathcal{E} \subseteq \mathcal{U}$.

Definition 4.1 *A globular object M in $\mathcal{P}$ is* self–reproducible *if and only if there exists an evolution, starting in a configuration C_M containing M as a single object which, after carrying a finite number of interactions among the globules from M and $\mathcal{E}$ gives rise to a configuration with at least two occurrences of M in $\mathcal{P}$.*

The notion of an offspring of a self–reproducing automaton is defined in an obvious way:

Definition 4.2 *We say that a self–reproducing object M_1 is an* ancestor *of a self–reproducing object M_2 (or that M_2 is an* offspring *of M_2) if and only if there is an evolution from a configuration C_{M_1} containing exactly M_1 into a configuration C_{M_2} containing M_2.*

Obviously, thanks to the nondeterminism an offspring of a self–reproducing automaton M can be either an exact copy of M or a different automaton (if different pathes in the evolution have been taken). It may happen that the different offspring is still a self–reproducing machine. If this new automaton is in a sense better than the old automaton we have embarked on a specific "positive" evolution. In order to capture in what sense the new automaton could be better we turn our attention towards the information processing ability of the underlying machine. In the sequel we will consider automata controlled by a finite state mechanism which is "made up" from globules. We will call such class of automata *globular automata*. Obviously, the processing power of a finite state mechanism is directly related to the number of its states. In principle, automata with more states are able to distinguish among a greater

number of different situations and thus are able to generate a richer repertoire of actions which can lead to a more sophisticated behavior and/or to a different machine's architecture. Yet in concrete cases it may happen that an automaton has more states than needed in order to generate a given behavior. Thus, the number of states alone is not an adequate indicator of an automaton's power. Therefore the complexity of a globular automaton should be defined by the number of states of a minimal finite state mechanism controlling an automaton in an identical way. Fortunately, in the sequel we will not need to know the minimal automaton producing a given behavior; instead, we will show that in principle such an automaton can be constructed in the process of evolution.

Definition 4.3 *We say that a self–reproducing globular automaton M is a* self–reproducing evolving automaton *(or that M* has an evolutionary potential*) in $\mathcal{U}$ if and only if for any given finite state automaton* A *among the offsprings of M in $\mathcal{U}$ there is a self–reproducing evolutionary globular automaton with a control mechanism realizing* A.

Obviously, a similar definition of a self–reproducing evolving automaton can be given also for the case of classical cellular automata; however, a realization of such an automaton need not be as simple as is the realization of such an automaton in a nondeterministic globular universe.

5 Constructing a self–assembly evolutionary self–reproducing globular automaton

The existence of a self–reproducing globular automaton within a given universe depends much on the properties of the globules in $\mathcal{U}$. For instance, in a universe with single–state globules, no self–reproducing automata can exist. Similarly, in universes with only neutral globules (i.e., no self–assembly is possible) no complex objects can be built and therefore only single–globule self–replication objects can exist in it, with no evolutionary potential.

The case where self–assembly works and the environment supplies globules "as we like it" is more interesting.

Theorem 5.1 *There exists a nondeterministic globular universe with a self–reproducing evolving globular automaton.*

A sketch of the proof: Let $\mathcal{U} = (C, \Phi)$ be a nondeterministic globular universe. We will define explicitly neither its configuration space nor the respective interaction function; rather this will become clear from the course of the proof. Let M be the globular automaton we are after. M will behave as follows: it will react to the sequence of environmental changes which follow a certain regular pattern described by a regular language $R \subseteq \Omega^*$ recognized by M. In such a case, i.e., if and only if M recognizes a word $w \in R$, M will either self–reproduce or generate a self–reproducing offspring with a different behavior, possibly with more states in its control mechanism than M had.

Let $w = \sigma_1\sigma_2\ldots\sigma_k \in \Omega^*$ be a word. This word will be "presented" to M as a series of "waves", the first wave consisting of globules from environment $\mathcal{E} \subseteq \mathcal{U}$ in state σ_1, etc. A "wave" means a situation when all globules arriving from the environment are in state σ_i, for $i = 1, 2, \ldots, k$. In this case, we can imagine that as though has M floated in a sea of globules in state σ_1, then σ_2, etc. and these globules interact with all globules on M's surface. We say that w is an input to M and that M *accepts* w if and only if M on input w will generate either an equivalent offspring or a self–reproducible offspring with a modified finite control mechanism (M must be able to generate either of the two, not always an equivalent offspring).

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic finite state automaton recognizing R. First we show how the transition function of this automaton will be represented in M. For simplicity of explanation, assume first that the cardinality of both Σ and Q is substantially less than the cardinality of C. Under such assumption there is a 1 to 1 correspondence between Q and a subset of C and between Σ and a subset of C and there are sufficiently many globules which can be used for other purposes rather than for representing sets Q and Σ. Now we can represent each element of these sets by a corresponding globule from C. In the sequel we will not distinguish between the elements of these sets and their globular representation. That is, we in fact assume that $\Sigma, Q \subset C$. Moreover, instead of saying "a globule in state q" we will often say "a globule q".

Especially note that there is a distinguished globule in state $q_0 \in C$. Then $\delta : Q \times \Sigma \to 2^Q$ can be represented as a finite sequence of the segments of globules of form $\$p\$\sigma\$q$ with $p, q \in Q$ and $\sigma \in \Sigma$. Each segment represents a transition of A of form $\delta(p, \sigma) = q$. Of course, for such a representation we need a further globule representing the separator \$. Obviously, since δ is a nondeterministic relation in its representation as stated before, there can be transitions with the same left–hand side. Assuming that the respective globules have strong bonds on their poles the entire δ can be represented as a linear strand of globules with the given syntax. W.l.o.g. we can join the ends of this strand to form a ring. To simplify the explanation we will call the first occurrence of Q in the segment $\$Q\$\Sigma\$Q\$$ as the "current state" whereas the second occurrence as the "new state". For technical reasons we will add to our representation of δ a transition of form $\delta(q_0, b) = q_0$, with $b \notin \Sigma$ and $b \notin \mathcal{E}$ (i.e., a globule in state b will never appear at the input to M).

In order to enable a smooth working of M we will further assume that the basic structure of M is created by a double–ring: in parallel, and bonded to the just described ring there exists a second auxiliary ring. This ring consists of segments of form \$a\$b\$a\$, with $a \notin Q, b \notin \Sigma$. The \$ symbols match in both rings and thus the globules in state a match the states from Q and the globules in state b match elements from Σ in the segment representation of δ in the original ring. This second ring is bonded to the first ring via bonds between the corresponding globules. The original ring will be called the first track whereas the auxiliary ring the second track.

Next M has to remember the current state $q \in Q$ of A. To represent the current state of A we will mark the right–hand side of the transition rule whose

application has caused A to enter state q. The marking will be realized in the second track (under the new state in the respective segment in the first track) by changing the state of the corresponding globule to state $s \notin Q$.

Now we need to define the input mechanism to our globular representation of A. In accordance with the assumption that the globules will appear at places when we need and where we need we will simply assume that the input globules in state $\sigma_i \in \Sigma$ will interact with all globules in the ring but only globules b in track two will react to the input and change their states to σ_i.

Now we are in a position to describe one move of a globular automaton.

The situation before the move is as follows: under each globule from Σ in the first track there is a globule $x = \sigma_i \in \Sigma$ corresponding in the second track to the symbol read by A at that time, under each globule, representing a state from Q in each segment, there is an auxiliary globule in state a. The only exception is in exactly one segment where under the second occurrence of a state from Q there is globule s marking the current state of A. Initially, s is placed under the second occurrence of q_0 in the "artificially added" transition $\delta(q_0, b) = q_0$.

Next we must first distribute the information on the current state of A into all segments. To this end s will interact with q and change its state to q. Then it will interact with its left neighbor in order to propagate q to the next segment until the whole ring gets circumvented. After this action under the first two symbols in each segment there is a pair (q, x) denoting the current state and symbol read by A. Then again the whole ring is circumvented by a signal to see if there is a match between the pair (q, x) and the globules above it. All matches get marked by setting the states of globules under the new state in the auxiliary ring to a distinguished state $m \notin Q$.

If there is no match discovered, i.e., in the case that the computation of A has stuck in a configuration from which there is no continuation, the initial state q_0 is entered and the recognition of a new sequence of inputs will begin.

If there is a match, then we have identified the set of transitions that can be potentially applied in this configuration of A. To apply a transition we non-deterministically select one from among all marked transitions and mark it by setting the state under the new state globule to state s. Then we "reset" all globules in the second track, except the one in state s, to their initial values a or b, respectively. Resetting is done by sending a signal around the ring and changing the states of globules m to a and those of globules x to b. In the next move, we assume that the globules from the previous input wave (i.e., globules "read" by the globular automaton) have left the observable universe. If the state marked by s is not a final state from set F the globular automaton is ready for the next move.

Otherwise, M enters the initial state of A, stops recognition and starts the self–replication. The replication proceeds as follows. It starts in the segment with the initial state. Therefrom a special signal is sent moving around the double–ring and doubling its structure. The new ring is built from the incoming globules by transcribing the states of the globules from the original ring into the new ring. The new ring touches the old one only at the currently copied segment. Of course, the newly copied parts do not remain attached to the old

ring — the respective bonds get cancelled. The new ring "grows" by inserting further segments to it. The whole process looks as though the new ring "rolled" over the old one and grew on this occasion. If the copying is done faithfully, the result of this process will be an exact replica that is separated from the old ring. However, the copying process can also be carried out nondeterministically, with alternatives introducing changes into the segments of the replicated ring.

In the sequel we will describe the design of a *nondeterministic evolutionary mechanism* which will generate an offspring with a modified transition function of the underlying finite state automaton A. These changes will not be completely arbitrary, e.g. they will preserve the chosen syntax of the transition relation. The new transition relation can either have the same number of segments, or more segments than the original relation.

The case with a transition relation having the same number of segments is easy — it is enough to introduce nondeterministic alternatives into the process of transcribing the elements of δ from the original to its copy. The copying process can be even designed so as to nondeterministically decide to skip copying of certain segments. To remain consistent with the idea of two rings rolling one on each other, the simplest way to achieve the desired effect of skipping one segment is to copy a *junk segment* (i.e. one which could never be interpreted due to its parameters) into the place of the skipped segment. Note that such a process can lead to a "backward" evolution, in a sense that an off-spring can be generated with a transition relation which might have been already generated in a previous generation.

The case when the new transition relation will have more segments then the original one is realized by letting the copying mechanism nondeterministically decide to copy certain segments twice. This is done as follows: after copying a given segment, the copying site (a signal, in fact) can change its direction and instead of proceeding further and carrying out the copying process it will backtrack one segment without carrying the copying process, but freing the already copied segment and binding the previous one in the original back to its counterpart in the copy. As a result, the copied ring roles backwards on the original ring by one segment. Then again the direction of the copying signal is reversed and the copying process can be resumed. Note that by backtracking both in the original and in the copied segment, the newly copied segment (and indeed, all subsequent segments) will be copied *before* the already copied segment, but this makes no harm since our representation of the transition relation is insensitive to the order of segments in it. On the occasion of its second copying the copied segment can be modified. It is clear that the new individuum can recognize a different regular language and can, but need not be able to self–replicate.

Now it should be clear that *over a fixed set* Q *and* Σ the evolutionary mechanism just described can generate arbitrary transition functions for A. That is, we still cannot "enumerate" all finite state control mechanisms (over all sets of states and all alphabets). In order to do so we must relax our assumption on the size of cardinalities of Q, Σ and C. However, if the cardinalities of both Σ and Q are larger than the cardinality of C we cannot map elements of Q and Σ to elements of C as before. Then we must encode elements of Σ and Q in a unary

notation into strings of identical globules from C. This will complicate the design of the self–replicating automaton but the above mentioned ideas will work also in this case. Especially, the evolutionary mechanism will now be free to generate "new" states and "new" symbols for A and this will enable evolutions leading to arbitrary complex transition relations. In this case M will become a real self–replicating evolutionary globular automaton according to Definition 4.3. This ends the sketch of the proof of theorem 5.1.

After presenting the construction of the previous self–reproducing evolutionary automaton two remarks are in order.

The first remark concerns the properties of universes in which self–reproducing evolutionary globular automata cannot exist. Namely, it is clear that our construction cannot be realized in case the cardinality of C is too small to allow encodings needed for a globular automaton to work as envisaged. Thus it seems that there is a lower bound on the cardinality of globular configuration space below which no self–reproducing evolutionary automata can exist even if a self–assembly within the respective universe is possible. Moreover, even if a configuration space of a given universe is sufficiently large, there still need not exist transitions among the states of globules which would allow constructing a self–reproducing evolutionary automaton.

The second remark concerns further properties of self–reproducing evolutionary automata which we did not pay attention to in our design. It appears that our self-replicating automaton can easily be made *individuated* in the following sense: arbitrary offsprings of a self–reproducing evolutionary automaton will maintain their separate identities also in a direct contact. This can be achieved by designing the universe and constructing M in such a way that the globules from which the globular automata are made bind only with the globules from the environment. This condition for automata to be individuated has been suggested by McMullin in [2].

6 Conclusion

In the paper we presented three main achievements. The first one was the design of a globular universe which enabled a study of self–assembly in a more general setting than the previous models did. This has been mainly due to getting rid of the rigid structure of cellular space (as in the case of classical cellular automata) or of the necessity to deal with explicit dynamic aspects of particle motion (as in the case of lattice gas automata). The introduction of nondeterminism both into the self–assembly process and into the dynamic input appearance in the observable part of the globular universe enabled concentration to principal existential questions related to the self–assembly of globular objects. A view of a basically continuous evolutionary process as that of a finite series of configurations taken at interaction times and related via sets of on–line interactions enabled a formal treatment of evolution much in the spirit of the computational theory. An application of ideas from the computational complexity theory has led to the second achievement, viz the formal definition

of a self–reproducing evolutionary automaton. Last but not least, the third achievement was the design of a specific nondeterministic globular universe and a constructive proof of the existence of a self–reproducing evolutionary automaton within this universe. The automaton itself is substantially simpler than the automata designed by other authors. Moreover, our automaton includes a nondeterministic evolutionary mechanism which guarantees the existence of evolutionary paths towards more complex automata thus answering positively von Neumann's question from the introduction in a constructive way.

We believe that the framework of a globular universe, or a similar one, will enable a further, more detailed study of questions related to self–reproduction, evolution and to the self–emergence of self–reproducing evolutionary machines. As far as the latter machines are concerned, machines with "more complicated" evolving bodies would be of interest. A possible avenue would be to consider the ideas from theoretical biology related to artificial life synthesis (cf.[5]). Here cell-like systems controlled by a genom and embedded in a membrane are considered. The first ideas along these lines in the spirit of our modelling have been sketched in [7]; in fact, the current paper emerged as a result of an effort to bring more formalism into the respective research.

References

[1] Herman, G. T.: On universal computer constructor. Information Processing Letters, Vol 2, pp. 61–64, 1973

[2] McMullin, B.: Some remarks on autocatalysis and autopoiesis. Annals of the New York Academy of Sciences, Vol. 901, pp. 163–174, 2000

[3] McMullin, B.: John von Neumann and the Evolutionary Growth of Complexity: Looking Backwards, Looking Forwards... Artificial Life, Vol 6. Issue 4, Fall 2000, pp. 347-361

[4] Rothemund, P., Winfree, E.: The program-size complexity of self-assembled squares (extended abstract). In Proceedings of the thirty-second annual ACM symposium on Theory of computing, pages 459-468. ACM Press, 2000.

[5] Szostak, J. W., Bartel, D. P., Luisi, P. L.: Synthesizing Life. Nature 409 (2001), pp. 389-390.

[6] von Neumann, J.: Theory of Selfreproducing Automata. A. Burks (Ed.), University of Illinois Press, Urbana and London, 1966

[7] Wiedermann, J.: Coupling computational and non–computational processes: minimal artificial life. Pre–proceedings of the Fifth Workshop on Membrane Computing (WMC5), G. Mauri, Gh. Paun, C. Zandroni (Eds.), Dept. of Comp. Sci., University of Milan — Bicocca, Italy, June 16–19, 2004, 444 p.