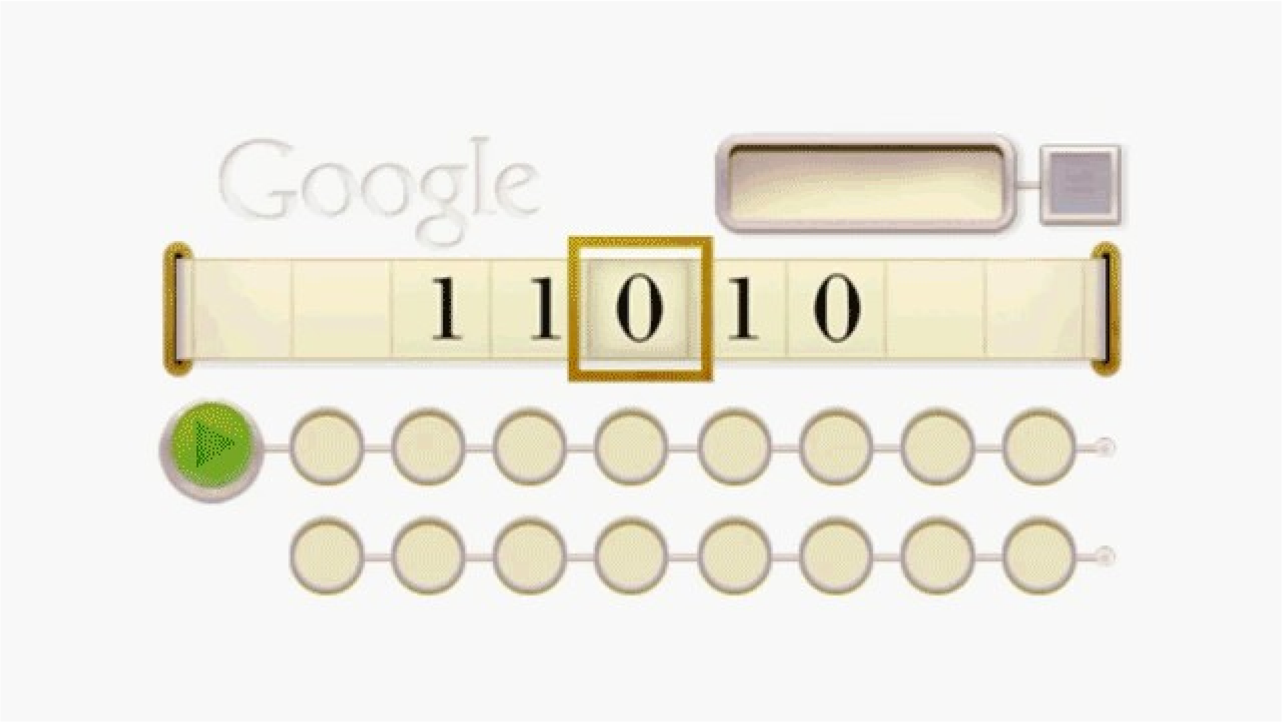
A Formal Framework for Static (Tissue) P Systems

Rudolf Freund and Sergey Verlan

Presented by: Ren Tristan A. de la Cruz

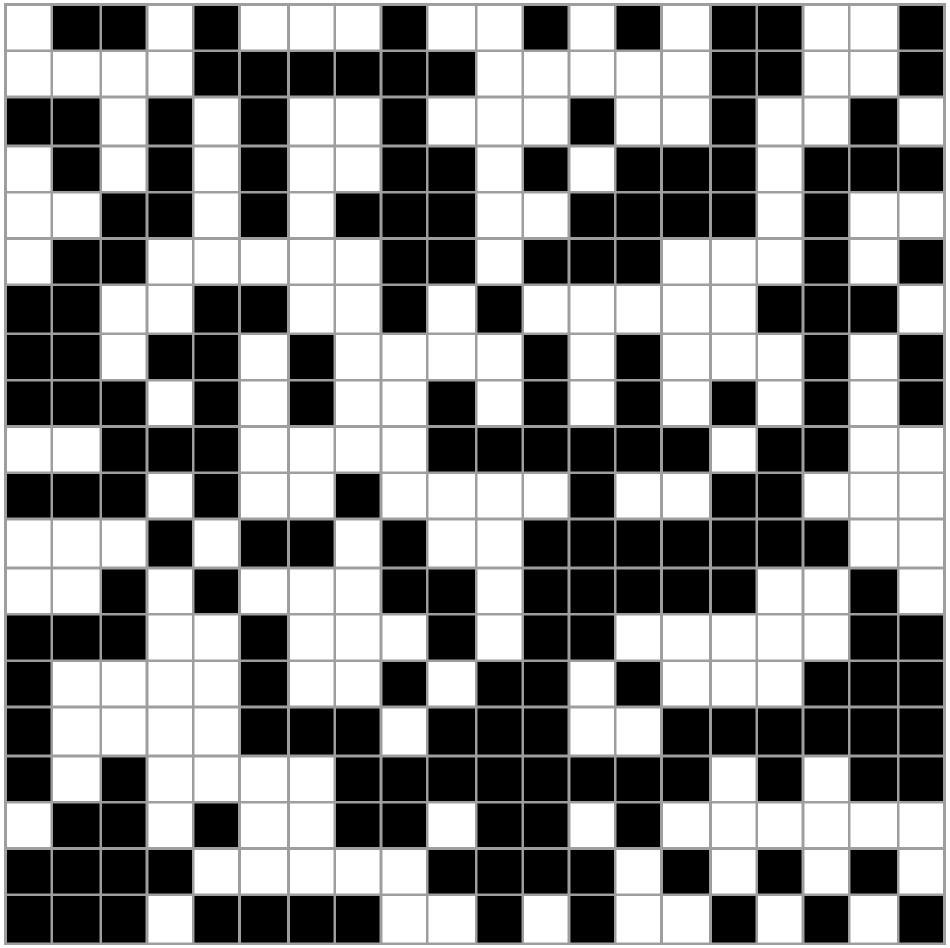
Models of Computation

Turing Machine (1936)



https://te.xel.io/posts/2017-02-27-the-turing-network-a-thought-experiment-for-quantum-computing.html

Cellular Automata (1940s)



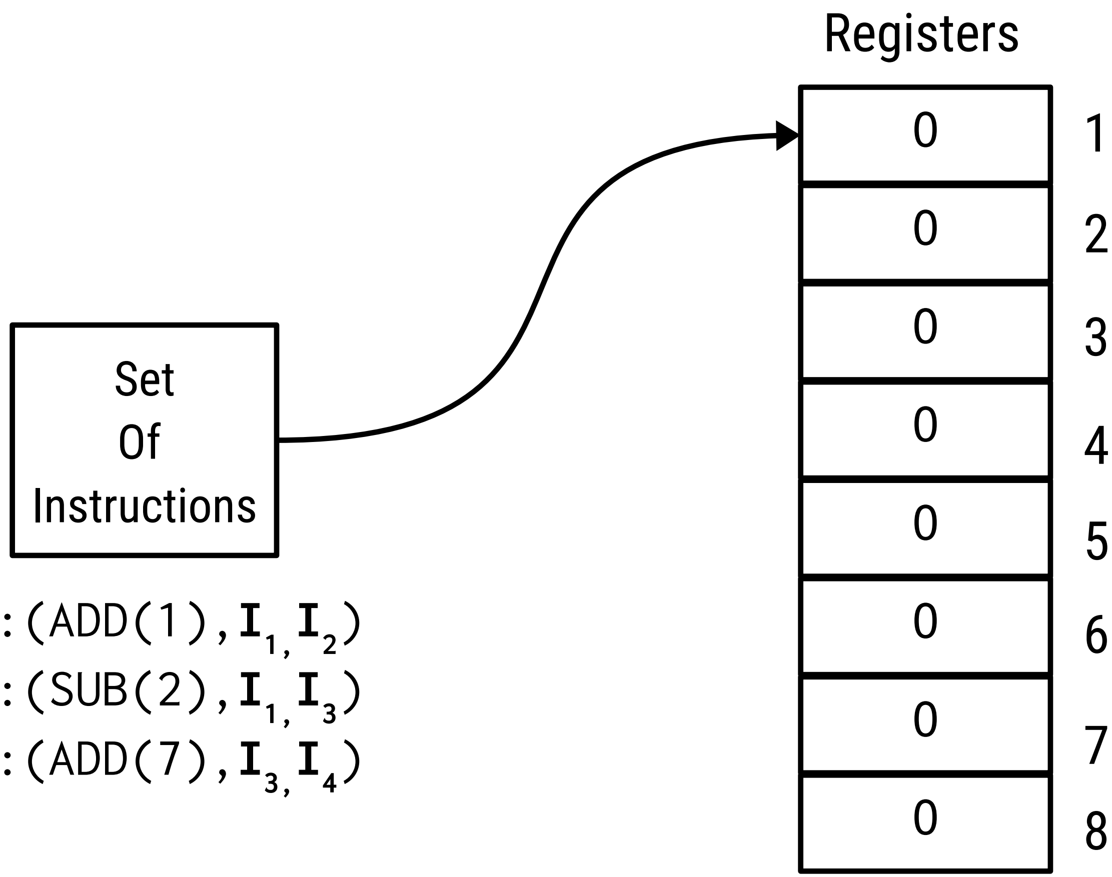
https://giphy.com/gifs/programming-automata-cellular-s1WbnYEoHbbmE

Lambda Calculus (1930s)

$$
\begin{aligned}
& (\underline{\lambda f}.\lambda g.\lambda h.fg(h\,h))\underline{(\lambda x.\lambda y.x)}h(\lambda x.x\,x) & \\
\to_\beta\ & (\lambda g.\underline{\lambda h}.(\lambda x.\lambda y.x)g(\underline{h\,h}))h(\lambda x.x\,x) & (1) \\
\to_\alpha\ & (\underline{\lambda g}.\lambda k.(\lambda x.\lambda y.x)g(k\,k))\underline{h}(\lambda x.x\,x) & (2) \\
\to_\beta\ & (\underline{\lambda k}.(\lambda x.\lambda y.x)h(k\,k))\underline{(\lambda x.x\,x)} & (3) \\
\to_\beta\ & (\underline{\lambda x}.\lambda y.x)\underline{h}((\lambda x.x\,x)\ (\lambda x.x\,x)) & (4) \\
\to_\beta\ & (\underline{\lambda y}.h)\underline{((\lambda x.x\,x)\ (\lambda x.x\,x))} & (5) \\
\to_\beta\ & h & (6)
\end{aligned}
$$

https://cs.stackexchange.com/questions/102732/beta-reduction-order-in-lambda-calulus

Register Machine (1960s)



$\mathbf{I}_1$:(ADD(1),$\mathbf{I}_1$,$\mathbf{I}_2$)
$\mathbf{I}_2$:(SUB(2),$\mathbf{I}_1$,$\mathbf{I}_3$)
$\mathbf{I}_3$:(ADD(7),$\mathbf{I}_3$,$\mathbf{I}_4$)
….

Membrane Computing &

P Systems

Cell-Like P Systems

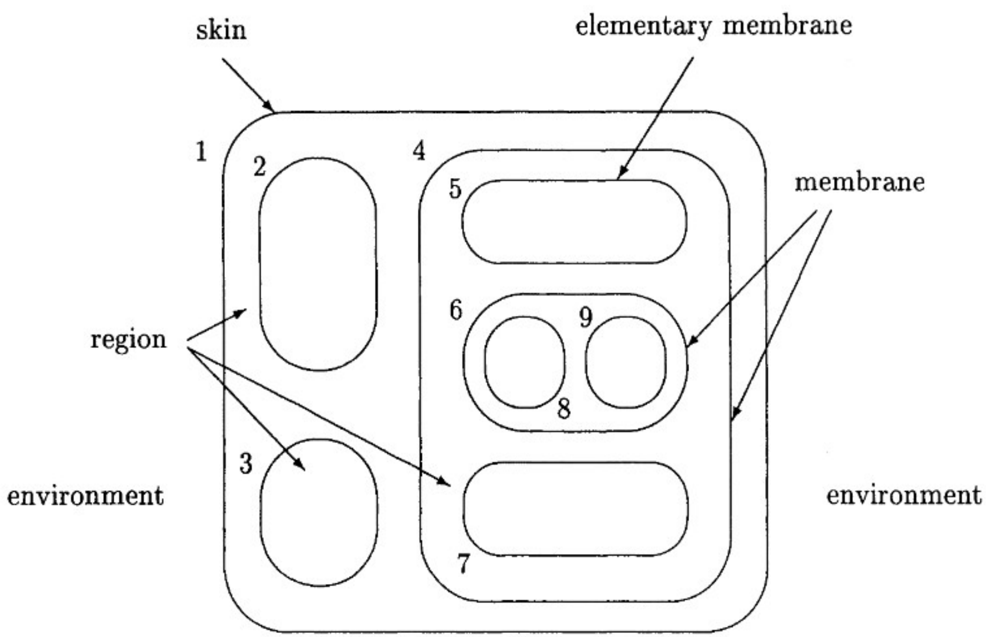


Fig. 1.2. A membrane structure

$$\Pi_2 = (O, \mu, \lambda, a^n c^k d, \lambda, (R_1, \rho_1), (R_2, \rho_2), (\emptyset, \emptyset), 3),$$
$$O = \{a, c, c', d, yes, no\},$$
$$\mu = [_1 [_2 \ ]_2 [_3 \ ]_3]_1,$$
$$R_1 = \{r_1 : dcc' \to (no, in_3),\ r_2 : d \to (yes, in_3)\},$$
$$\rho_2 = \{(r_1, r_2)\},$$
$$R_2 = \{r_3 : ac \to c',\ r_4 : ac' \to c,\ r_5 : d \to d\delta\},$$
$$\rho_2 = \{(r_3, r_5), (r_4, r_5)\}.$$

The structure of Π_2 is better seen in Figure 3.4.

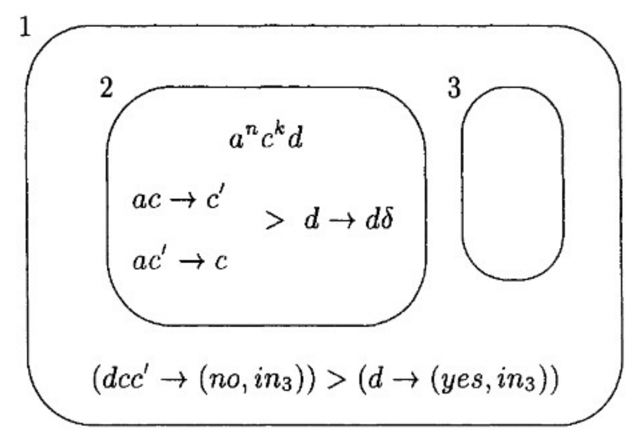


Fig. 3.4. A system deciding whether k divides n

Tissue-Like P Systems

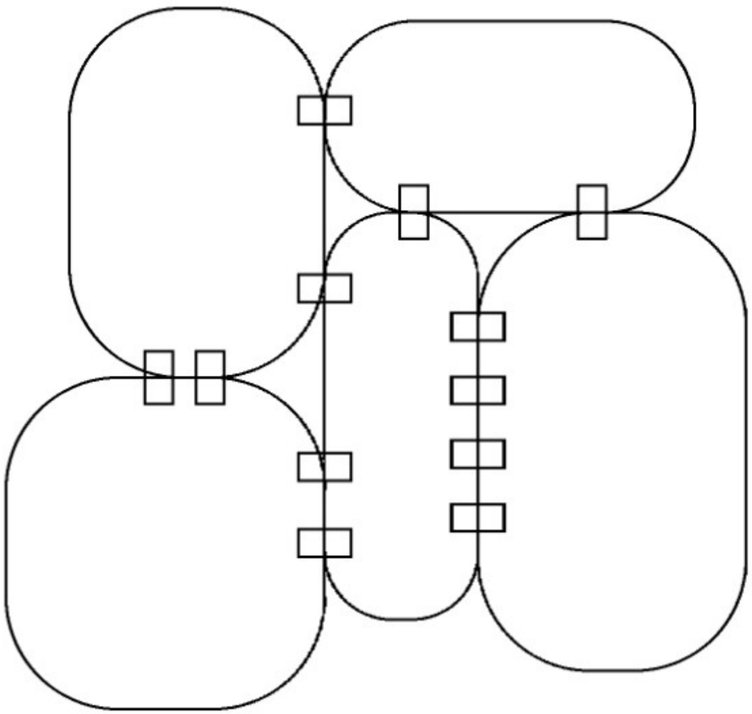


Fig. 1. Inter-cellular communication.

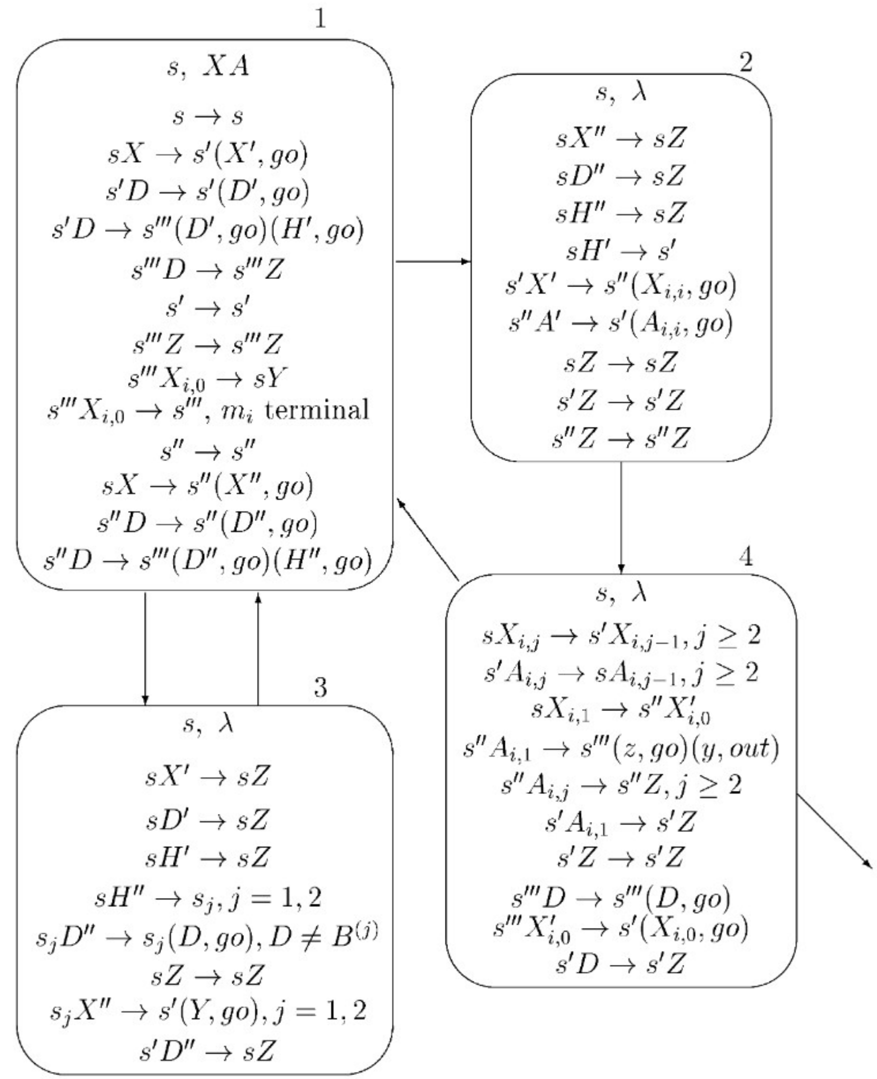
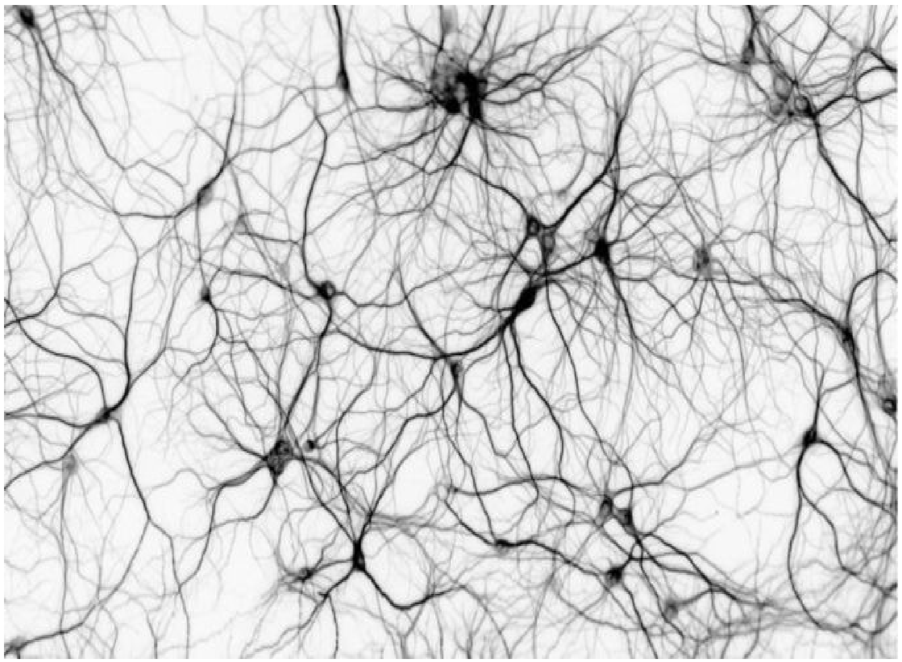


Fig. 3. The tP system from the proof of Theorem 4.

Neural-Like P Systems



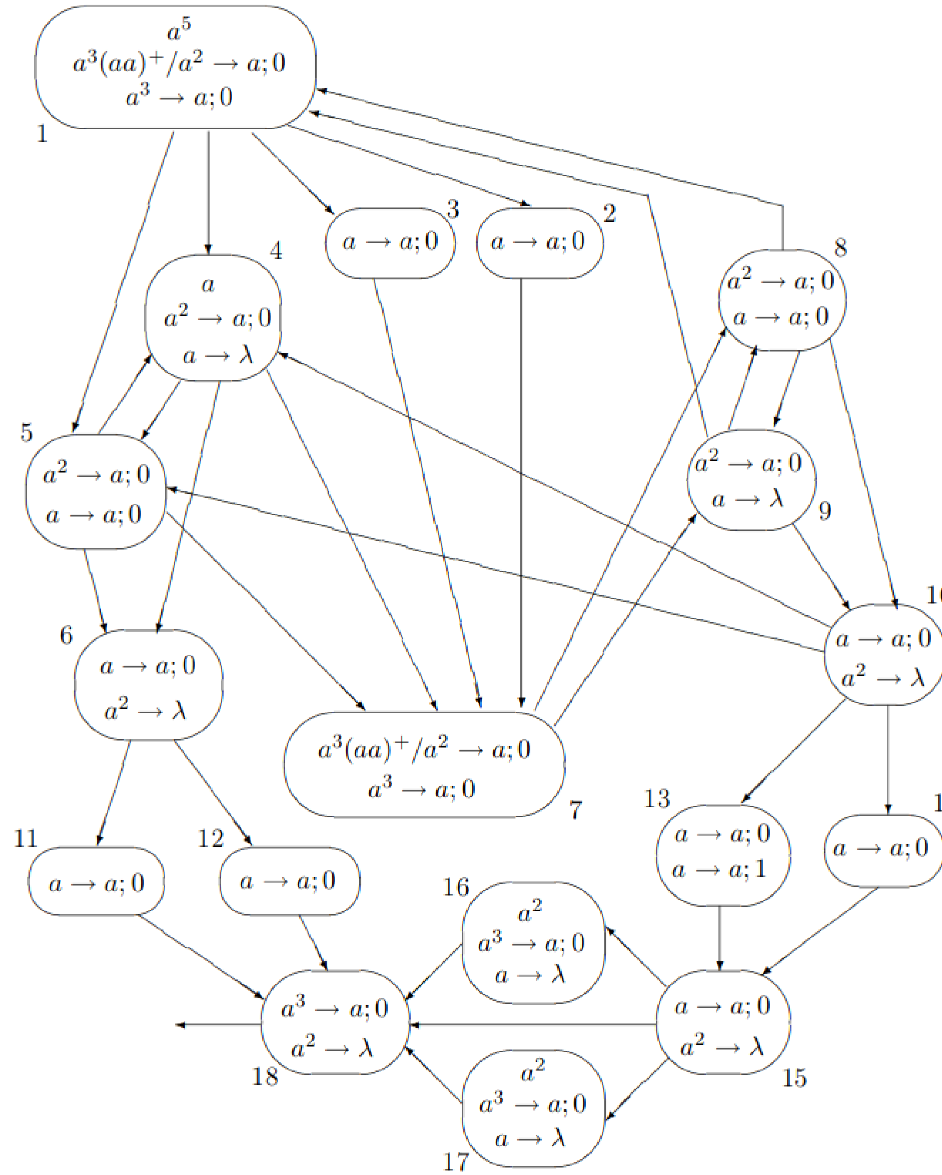


Figure 14: A "small" SN P system computing a non-semilinear set of numbers

Formal Framework

Network of Cells/Membranes

Definition 3.1. *A* $\boxed{\text{network of cells}}$ of degree $n \geq 1$ *is a construct*

$$\Pi = (n, V, w, Inf, R), \text{ where:}$$

1. *n is the number of cells;*
2. *V a* finite alphabet*;*
3. *$w = (w_1, \ldots, w_n)$ where $w_i \in \langle V, \mathbb{N} \rangle$, for all $1 \leq i \leq n$, is the* finite multiset initially associated to cell *i;*
4. *$Inf = (Inf_1, \ldots, Inf_n)$ where $Inf_i \subseteq V$, for all $1 \leq i \leq n$, is the* set of symbols occurring infinitely often in cell *i (in most of the cases, only one cell, called* the environment*, will contain symbols occurring with infinite multiplicity);*
5. *R is a finite set of* $\boxed{\text{interaction rules}}$ *of the form*

$$\boxed{(X \to Y; P, Q)}$$

where $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$, and $x_i, y_i \in \langle V, \mathbb{N} \rangle$, $1 \leq i \leq n$, are vectors of multisets over V as well as $P = (p_1, \ldots, p_n)$, $Q = (q_1, \ldots, q_n)$,

and p_i, q_i, $1 \leq i \leq n$, are finite sets of multisets over V. We will also use the notation

$$\boxed{((x_1, 1) \ldots (x_n, n) \to (y_1, 1) \ldots (y_n, n); (p_1, 1) \ldots (p_n, n), (q_1, 1) \ldots (q_n, n))}$$

for a rule $(X \to Y; P, Q)$; moreover, if some p_i or q_i is an empty set or some x_i or y_i is equal to the empty multiset, $1 \leq i \leq n$, then we may omit it from the specification of the rule.

Configuration

Definition 4.1. *Consider a network of cells* $\Pi = (n, V, w, Inf, R)$. *A configuration* C *of* Π *is an* n*-tuple of multisets over* V $(u'_1, \ldots, u'_n)$ *with* $u'_i \in \langle V, \mathbb{N}_\infty \rangle$, $1 \leq i \leq n$; *in the following,* C *will also be described by its finite part* C^f *only, i.e., by* $(u_1, \ldots, u_n)$ *satisfying* $u'_i = u_i \cup Inf_i{}^\infty$ *and* $u_i \cap Inf_i = \emptyset$, $1 \leq i \leq n$.

In the sense of the preceding definition, the *initial configuration* of Π, C_0, is described by w, i.e., $C_0{}^f = w = (w_1, \ldots, w_n)$, whereas $w'_i = w_i \cup Inf_i{}^\infty$, $1 \leq i \leq n$, is the initial contents of cell i, i.e., $C_0 = w \cup Inf^\infty$.

Eligible Rules

Definition 4.2. *We say that an* $\boxed{\textit{interaction rule } r = (X \to Y; P, Q) \textit{ is } \text{eligible}}$ *for the configuration* C *with* $C = (u_1, \ldots, u_n)$ *if and only if for all* i, $1 \leq i \leq n$, *the following conditions hold true:*

- *for all* $p \in p_i$, $p \subseteq u_i$ *(every* $p \in p_i$ *is a submultiset of* u_i*),*
- *for all* $q \in q_i$, $q \nsubseteq u_i$ *(no* $q \in q_i$ *is a submultiset of* u_i*), and*
- $x_i \subseteq u_i$ *(*x_i *is a submultiset of* u_i*).*

Moreover, we require that $x_j \cap (V - inf_j) \neq \emptyset$ *for at least one* j, $1 \leq j \leq n$. *This last condition ensures that at least one symbol appearing only in a finite number of copies is involved in the rule. The set of all rules eligible for* C *is denoted by* $Eligible(\Pi, C)$.

Derivation Modes

For the specific *derivation modes* to be defined in the following, the selection of multisets of rules applicable to a configuration C has to be a specific subset of $Appl(\Pi, C)$.

Definition 4.5. *For the derivation mode ϑ, the selection of multisets of rules applicable to a configuration C is denoted by $Appl(\Pi, C, \vartheta)$.*

Definition 4.6. *For the* asynchronous *derivation mode (asyn),*

$$Appl(\Pi, C, asyn) = Appl(\Pi, C),$$

i.e., there are no particular restrictions on the multisets of rules applicable to C.

Definition 4.7. *For the* sequential *derivation mode (sequ),*

$$Appl(\Pi, C, sequ) = \{R' \mid R' \in Appl(\Pi, C) \ \textit{ and } \ |R'| = 1\},$$

i.e., any multiset of rules $R' \in Appl(\Pi, C, sequ)$ has size 1.

The most important derivation mode considered in the area of P systems from the beginning is the *maximally parallel* derivation mode where we only select multisets of rules R' that are not extensible, i.e., there is no other multiset of rules $R'' \supsetneq R'$ applicable to C.

Definition 4.8. *For the* maximally parallel *derivation mode (max),*

$$\begin{aligned} Appl(\Pi, C, max) = \{R' \mid R' \in Appl(\Pi, C) \ &\textit{ and there is} \\ &\textit{no } R'' \in Appl(\Pi, C) \ \textit{ with } R'' \supsetneq R'\}. \end{aligned}$$

Halting Conditions

4.1 Halting Conditions

A halting condition is a predicate applied to an accessible configuration. The system halts according to the halting condition if this predicate is true for the current configuration. In such a general way, the notion halting with final state or signal halting can be defined as follows:

Definition 4.16. *An accessible configuration C is said to fulfill the* signal halting *condition or* final state halting *condition* (S) *if and only if $C \in S(\Pi, \vartheta)$ where*

$$S(\Pi, \vartheta) = \{C' \mid C' \in Acc(\Pi) \text{ and } State(\Pi, C', \vartheta) = \textbf{true}\}.$$

Here $State(\Pi, C', \vartheta)$ means a decidable feature of the underlying configuration C', e.g., the occurrence of a specific symbol (signal) in a specific cell.

The most important halting condition used from the beginning in the P systems area is the *total halting*, usually simply considered as *halting*:

Definition 4.17. *An accessible configuration C is said to fulfill the* total halting *condition* (H) *if and only if no multiset of rules can be applied to C with respect to the derivation mode anymore, i.e., if and only if $C \in H(\Pi, \vartheta)$ where*

$$H(\Pi, \vartheta) = \{C' \mid C' \in Acc(\Pi) \text{ and } Appl(\Pi, C', \vartheta) = \emptyset\}.$$

The adult halting condition guarantees that we still can apply a multiset of rules to the underlying configuration, yet without changing it anymore:

Definition 4.18. *An accessible configuration C is said to fulfill the* adult halting *condition* (A) *if and only if $C \in A(\Pi, \vartheta)$ where*

$$\begin{aligned} A(\Pi, \vartheta) = \{C' \mid\ & C' \in Acc(\Pi),\ Appl(\Pi, C', \vartheta) \neq \emptyset \text{ and} \\ & Apply(\Pi, C', R') = C' \text{ for every } R' \in Appl(\Pi, C', \vartheta)\}. \end{aligned}$$

Remarks