“Rule synchronization” in P systems

Péter Battyányi and **György Vaszil**

Faculty of Informatics
University of Debrecen
Hungary

Sevilla, February 6, 2020

UNIVERSITY of DEBRECEN

The idea of rule synchronization

$$a^n b^m u$$

$$b \to bd \otimes au \to u$$

1

Fig. 1. Multiplication by using synchronization

[Bogdan Aman, Gabriel Ciobanu: Synchronization of Rules in Membrane Computing. In: *Proc. CMC 20*, 257-268.]

UNIVERSITY *of* DEBRECEN

$$a^n b^m u$$

$$b \to bd \otimes au \to u$$

1

$a^n b^m u$

$a^{n-1} (bd)^m u$

$a^{n-2} (bd^2)^m u$

$\vdots$

$a (bd^{n-1})^m u$

$(bd^n)^m u \Leftrightarrow b^m \boxed{d^{n \cdot m}} u$

UNIVERSITY of DEBRECEN

Notice:

- Given rules $r_1 \otimes r_2$, both r_1 and r_2 have to be applied <u>at least once</u>
- Otherwise, application is maximal parallel

Convenient way to simulate register machines

- The value of register *r* corresponds to the multiplicity of a_r
- Labels l_i govern the functioning

$$l_1 : (ADD(r), l_2, l_3),$$

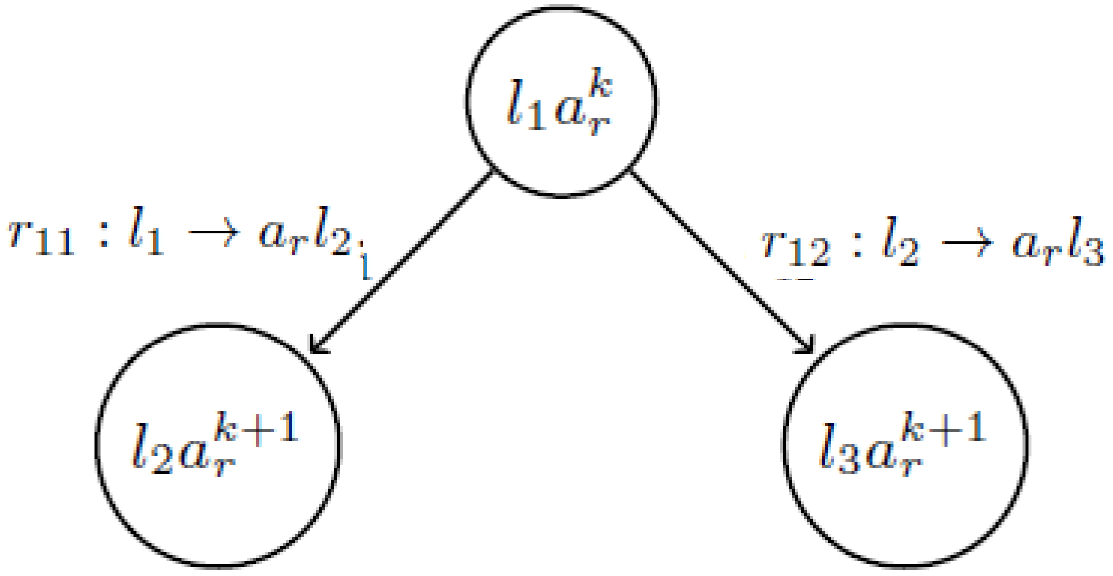


Fig. 3. Simulating ADD instruction.

UNIVERSITY of DEBRECEN

[Bogdan Aman, Gabriel Ciobanu: Synchronization of Rules in Membrane Computing. In: *Proc. CMC 20*, 257-268.]

Convenient way to simulate register machines

$l_1 : (SUB(r), l_2, l_3)$

$$r_{21} : l_1 \to l_1',$$
$$r_{22} : a_r u \to v,$$
$$r_{23} : l_1' \to l_2,$$
$$r_{24} : v \to u,$$
$$r_{25} : l_1' \to l_3 u,$$
$$r_{26} : u \to \varepsilon,$$

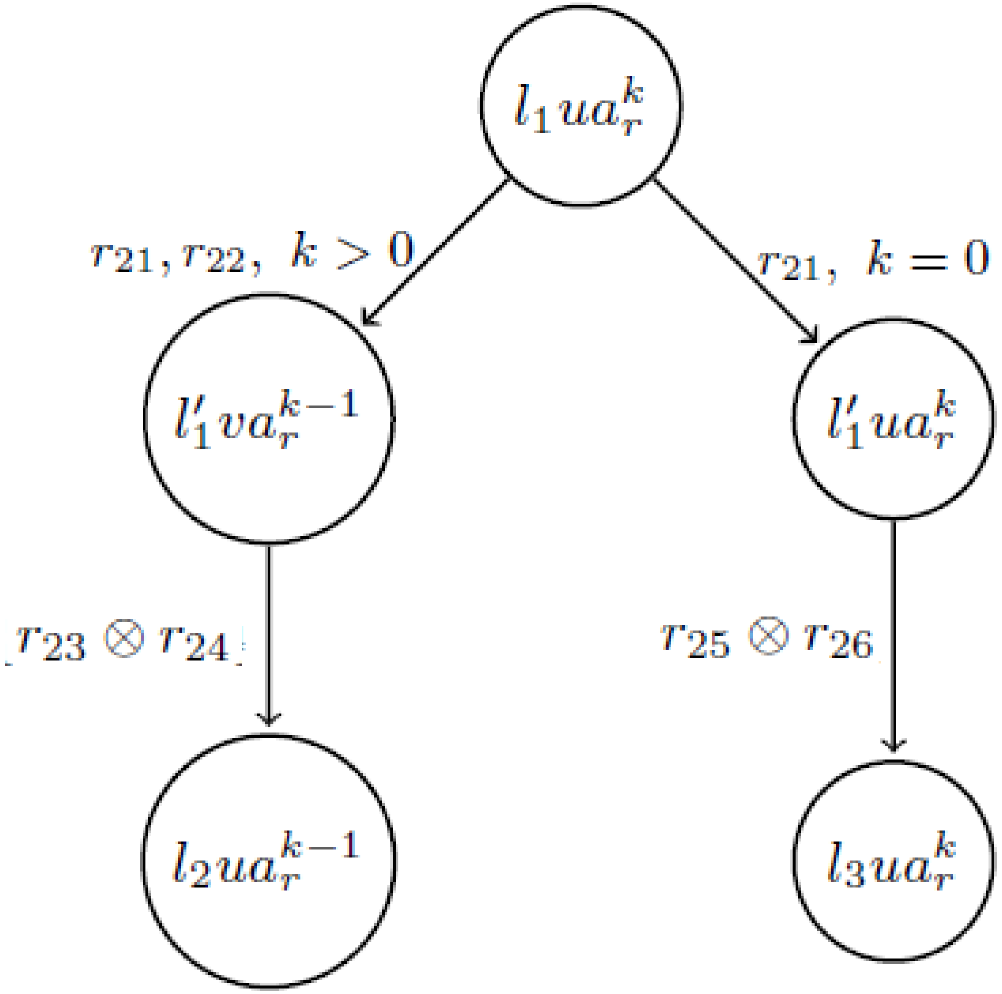


Fig. 4. Simulating SUB instruction.

Our first question

- **Rule synchronization** and **maximal parallelism** can simulate Turing machines
- **Without rule synchronization** we also get the maximal **Turing machine** equivalent power

- What happens if we have rule synchronization **without the maximal parallelism – with “free” rule application**?

UNIVERSITY of DEBRECEN

Our answer

- **Petri nets** can be viewed as computing models
- Simple place-transition are **less powerful** than Turing machines

- **Rule synchronized** systems with **free rule application** can be **simulated** by simple place-transition nets

UNIVERSITY of DEBRECEN

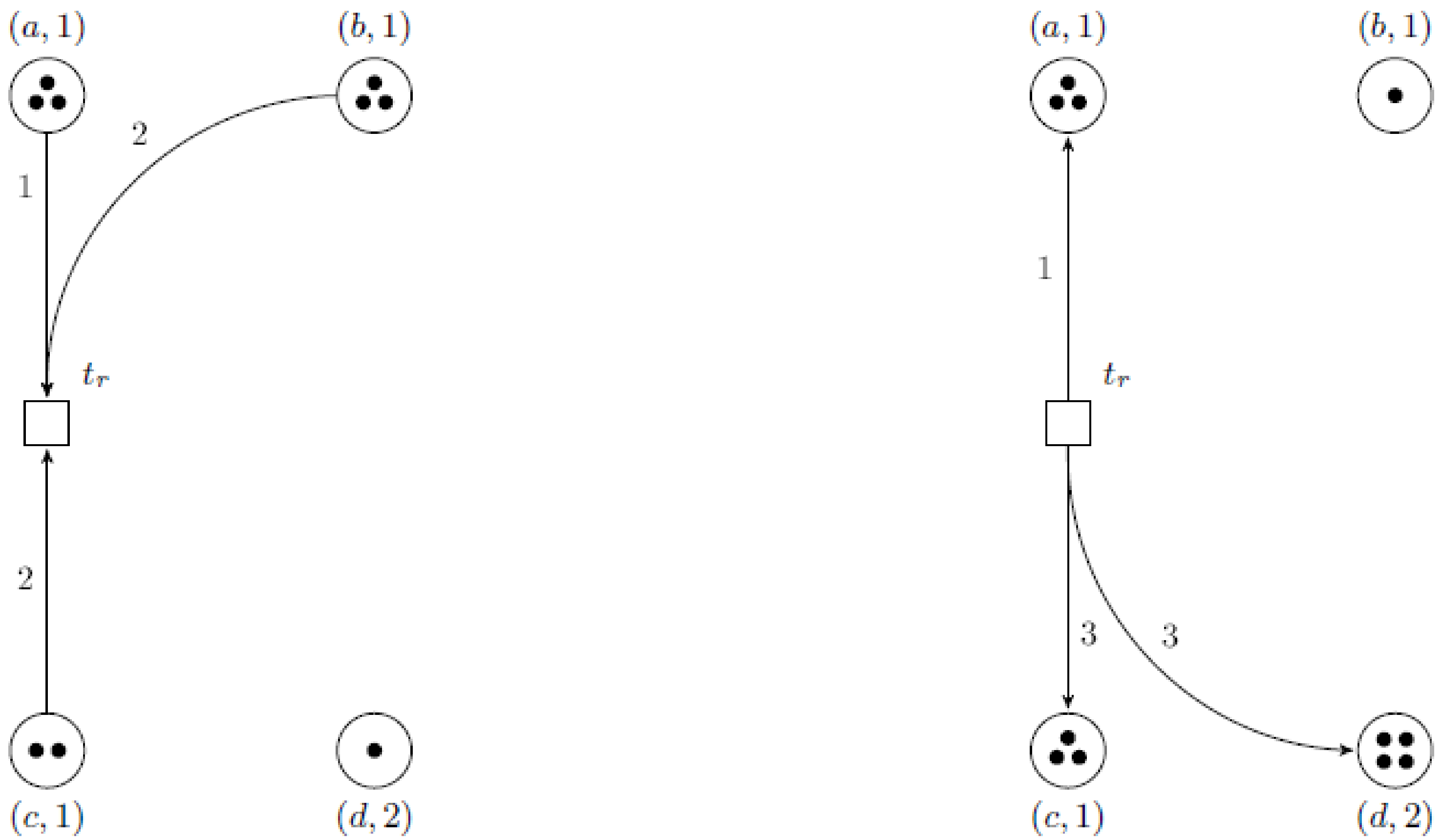


Fig. 1. Assume that $w_1^0 = a^3b^3c^2$, $w_2^0 = d$ and $r = ab^2c^2 \to ac^3(d, in_2)^3$. The figure on the left shows the arcs pointing to transition t_r together with their weights, the figure on the right shows the arcs going out from transition t_r together with their weights.

Our second question/research suggestion

- With rule synchronization we can “do **arithmetic**” conveniently
- Can we use this to **convert** ordinary P systems to **reversible systems**?

UNIVERSITY of DEBRECEN

Why is reversibility interesting?

Landauer's principle – **theoretical lower bound** on the energy consumption of computation

- The **logically irreversible** operations necessarily **dissipate energy**
- Landauer limit: the **minimum possible amount** of energy required to **erase one bit** of information is

  $$k \cdot T \cdot \ln 2$$

  where k is the Boltzmann constant, T is the temperature of the circuit in Kelvins

The idea

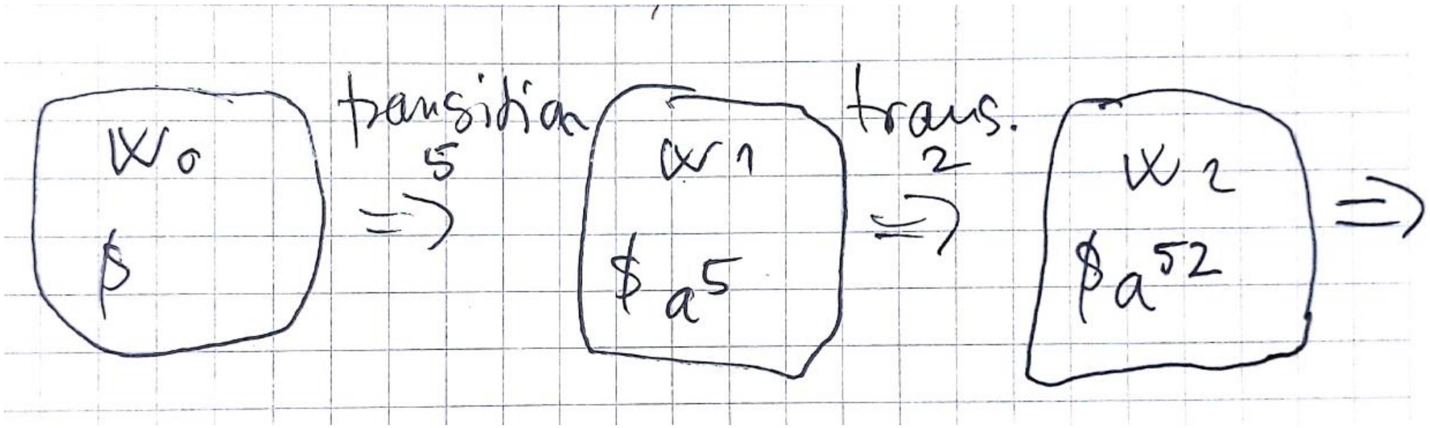
- If the **number** of possible different **transitions is bounded**,
- we can **record the "computation history"** inside the system, and
- use this history to make the machine "**backwards deterministic**".

UNIVERSITY of DEBRECEN

[Charles H. Bennett, Logical reversibility of computation. *IBM journal of Research and Development* 17.6 (1973): 525-532.]

Stage		Quadruples	*Contents of tape*		
			Working tape	*History tape*	*Output tape*
			_INPUT	_	_
Compute[a]	1)	$\begin{cases} A_1[b\ /\ b] \to [b + b]A_1' \\ A_1'[/\ b\ /] \to [+\ 1\ 0]A_2 \end{cases}$			
		$\vdots$			
	m)	$\begin{cases} A_j[T\ /\ b] \to [T' + b]A_m' \\ A_m'[/\ b\ /] \to [\sigma\ m\ 0]A_k \end{cases}$			
		$\vdots$			
	N)	$\begin{cases} A_{f-1}[b\ /\ b] \to [b + b]A_N' \\ A_N'[/\ b\ /] \to [0\ N\ 0]A_f \end{cases}$			
			_OUTPUT	HISTOR$\underline{\text{Y}}$	_
Copy output[b]		$A_f[b\ N\ b] \to [b\ N\ b]B_1'$			
		$B_1'[/\ /\ /] \to [+\ 0\ +]B_1$			
	$x \neq b$:	$\{\ B_1[x\ N\ b] \to [x\ N\ x]B_1'\ \}$			
		$B_1[b\ N\ b] \to [b\ N\ b]B_2'$			
		$B_2'[/\ /\ /] \to [-\ 0\ -]B_2$			
	$x \neq b$:	$\{\ B_2[x\ N\ x] \to [x\ N\ x]B_2'\ \}$			
		$B_2[b\ N\ b] \to [b\ N\ b]C_f$			
			_OUTPUT	HISTOR$\underline{\text{Y}}$	_OUTPUT
Retrace	N)	$\begin{cases} C_f[/\ N\ /] \to [0\ b\ 0]C_N' \\ C_N'[b\ /\ b] \to [b - b]C_{f-1} \end{cases}$			
		$\vdots$			
	m)	$\begin{cases} C_k[/\ m\ /] \to [-\sigma\ b\ 0]C_m' \\ C_m'[T'\ /\ b] \to [T - b]C_j \end{cases}$			
		$\vdots$			
	1)	$\begin{cases} C_2[/\ 1\ /] \to [-\ b\ 0]C_1' \\ C_1'[b\ /\ b] \to [b - b]C_1 \end{cases}$			
			_INPUT	_	_OUTPUT

The idea



…

The arithmetic we need

$$\underbrace{a \ldots a}_{5\,\text{db}} \longleftrightarrow \underbrace{a \ldots a}_{52\,\text{db}}$$

52 = 5 * 10 + 2

5 = (52 – 2) / 10

UNIVERSITY *of* DEBRECEN

Division

$$a^n b^m u_1$$

$$b \to bd \otimes u_1 \to u_2$$

$$ab \to \varepsilon \otimes u_2 \to u_3 u_0 \otimes d \to r$$

$$u_0 \to u_0' \quad au_3 \to au_{3a} \quad bu_3 \to bu_{3b}$$

$$u_{3a} \to u_1 c \otimes r \to b \otimes u_0' \to \varepsilon$$

$$u_{3b} \to \varepsilon \otimes br \to \varepsilon \otimes u_0' \to \varepsilon$$

$$u_3 \to c \otimes r \to \varepsilon \otimes u_0' \to \varepsilon$$

1

Fig. 2. Division by using synchronization

$a^n b^m u_1$

$a^n (bd)^m u_2$

$b^{m-n} r^m u_3 u_0$

$a^{n-m} r^m u_3 u_0$

$a^{n-m} r^m u_{3a} u_0'$ | $u_{3b} b^{m-n} r^m u_0'$

$a^{n-m} b^m u_1 c$ | r^n

$a^{52} b^{10} u_1 \Rightarrow r^2 c^5$

[Bogdan Aman, Gabriel Ciobanu: Synchronization of Rules in Membrane Computing. In: *Proc. CMC 20*, 257-268.]

How can rule synchronization be useful?

rules for $x \Rightarrow 10 \cdot x + 2$ $\otimes$ rules for transition 2

rules for $y \Rightarrow (y-2)/10$ $\otimes$ rules for the „inverse" of transition 2

UNIVERSITY of DEBRECEN

A problem

- **What do we mean** by “transition 2”?
- Do the **rule combinations** identify the **transitions** in a way that is necessary?

No. A simple example

$a \to aa \otimes aa \to aa$

aaaaa => *aaaaaaaa* $\boldsymbol{a^5} \Rightarrow a^8$

aaaaa => *aaaaaa* $\boldsymbol{a^5} \Rightarrow a^6$

aaaaa => ***aaaaaa*** $a^5 \Rightarrow \boldsymbol{a^6}$

aaaa => ***aaaaaa*** $a^4 \Rightarrow \boldsymbol{a^6}$

UNIVERSITY *of* DEBRECEN

Related question

- How to **avoid “ambiguous**” situations?
- Avoid maximal **parallelism** – let **all rules** in the rule “tuple” be executed **exactly once**?
- Are there **special cases** where maximal **parallel application** makes **no trouble**?
- What is the **power of** these possibly “**reversible**” variants?

UNIVERSITY of DEBRECEN

Acknowledgments

Research supported in part by project no. **K 120558**, implemented with the support provided from the **National Research, Development and Innovation Fund** of Hungary, financed under the **K 16** funding scheme, and

UNIVERSITY *of*
DEBRECEN