Using Inhibitors to Achieve Universality of Sequential P Systems

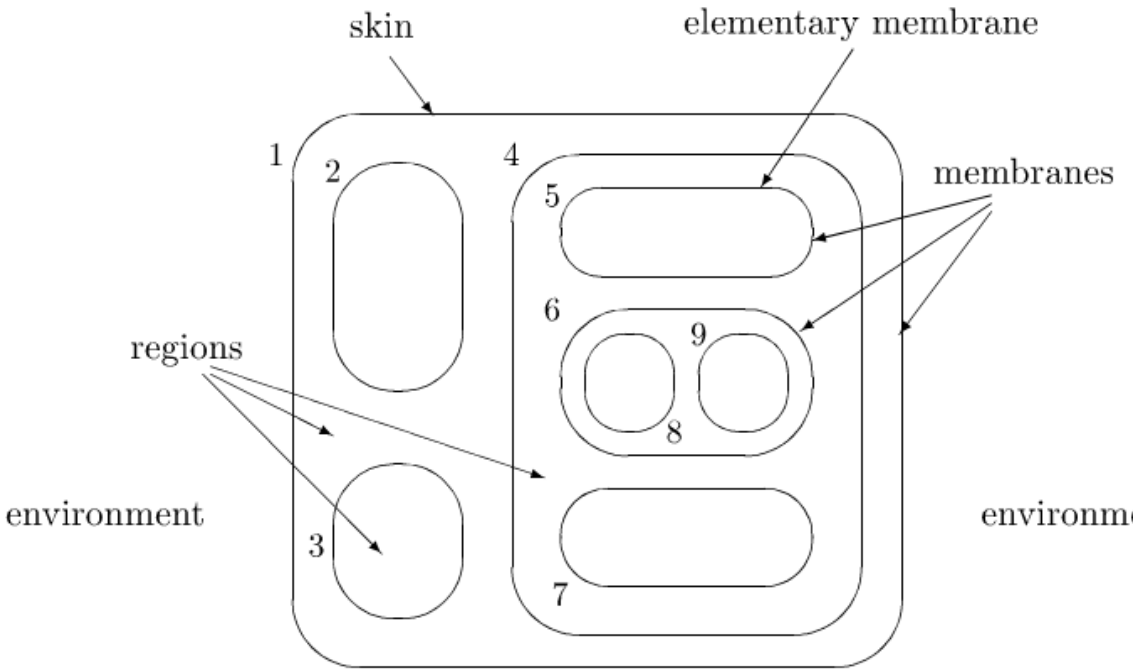
Michal Kováč

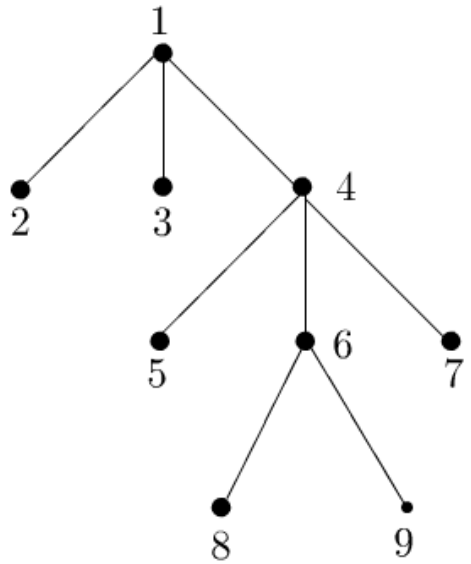
FMFI UK, Slovakia

24.6.2014

1. Overview of P systems
   - P systems
   - Variants
2. Sequential P systems with inhibitors
   - Accepting case
   - Generating case

Membrane structure





Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules

Sequential vs. maximal parallel rule application

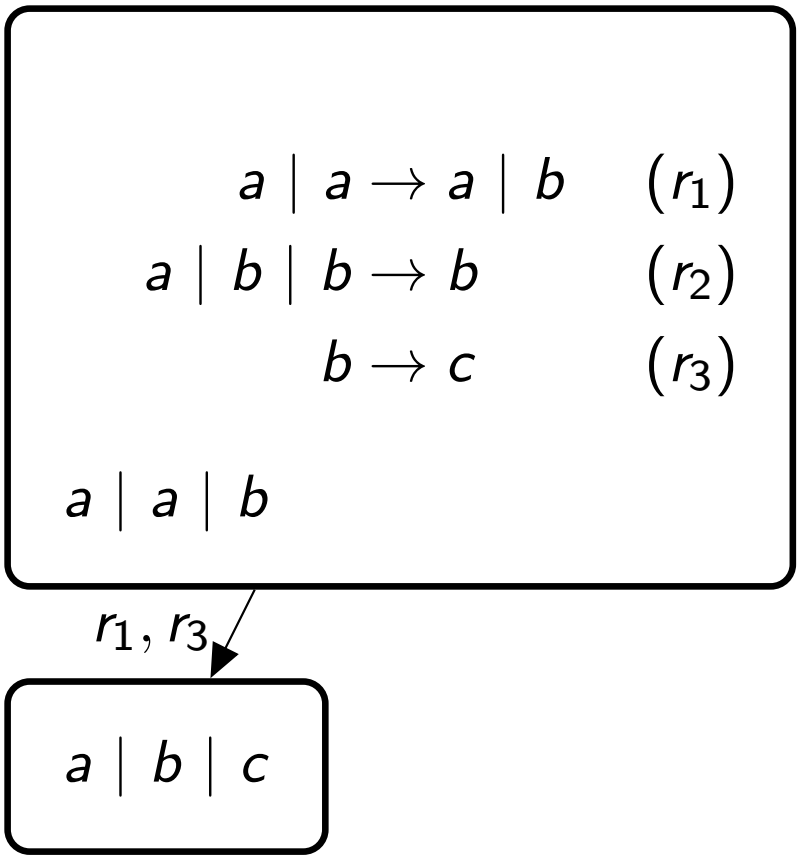
- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules

$$a \mid a \to a \mid b \quad (r_1)$$
$$a \mid b \mid b \to b \quad (r_2)$$
$$b \to c \quad (r_3)$$

$$a \mid a \mid b$$

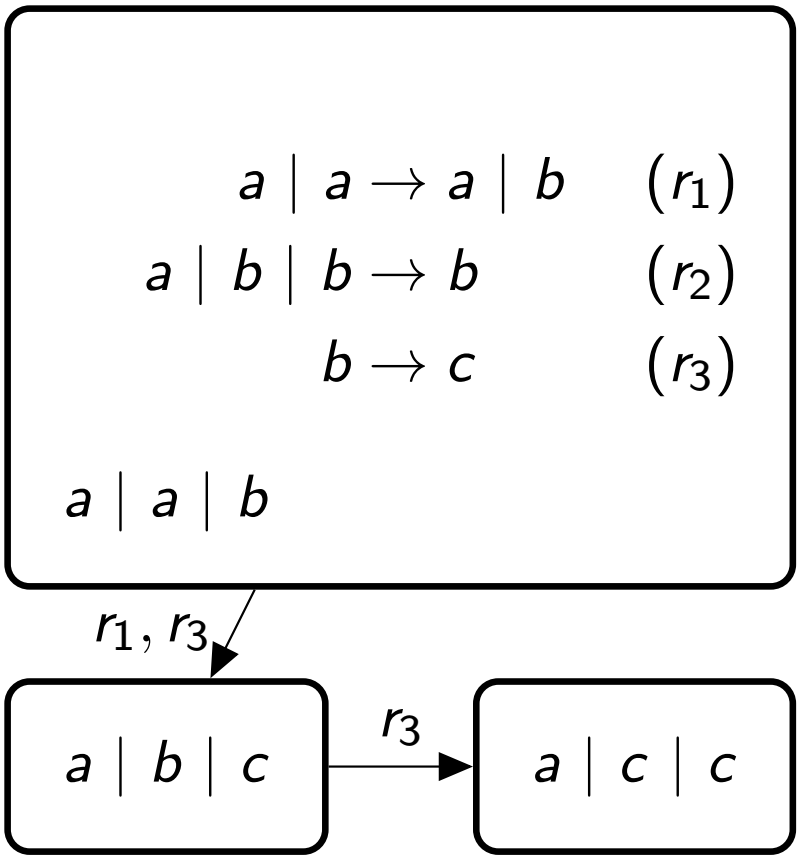
Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



Sequential vs. maximal parallel rule application

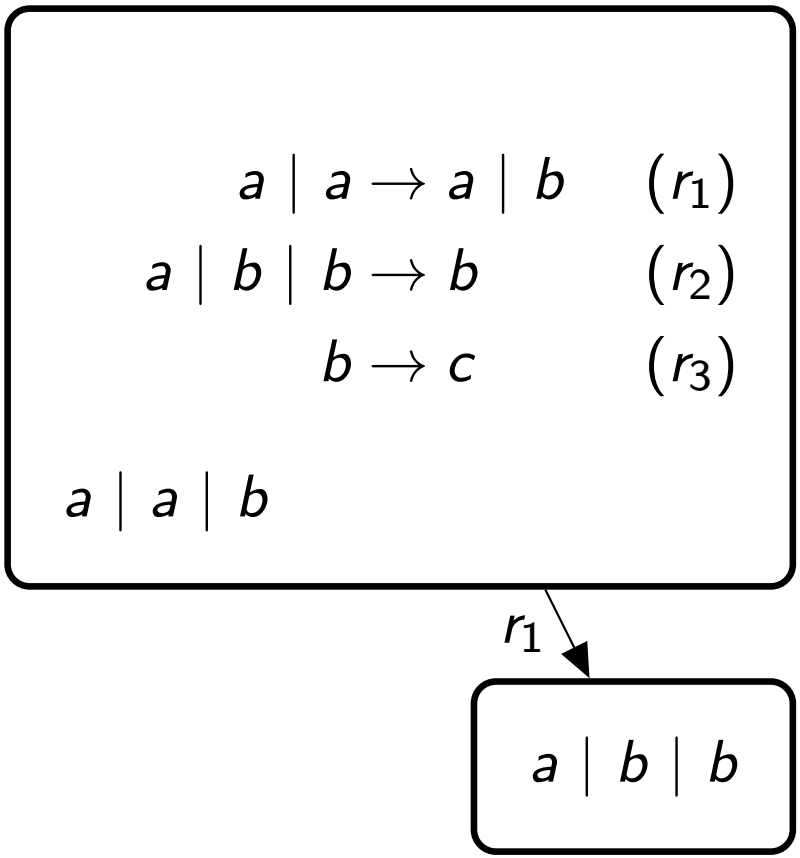
- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules

$$a \mid a \to a \mid b \quad (r_1)$$
$$a \mid b \mid b \to b \quad (r_2)$$
$$b \to c \quad (r_3)$$

$$a \mid a \mid b$$

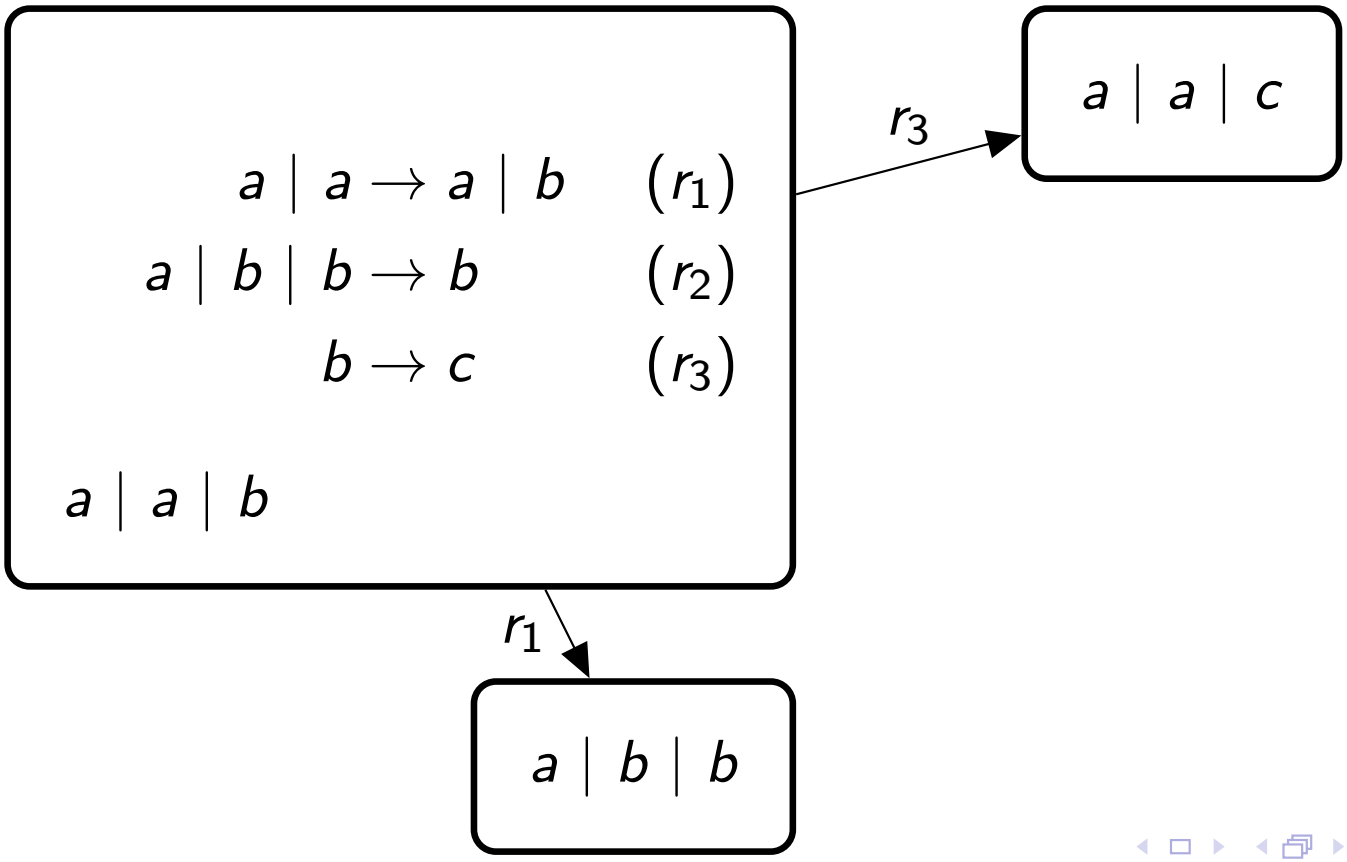
Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



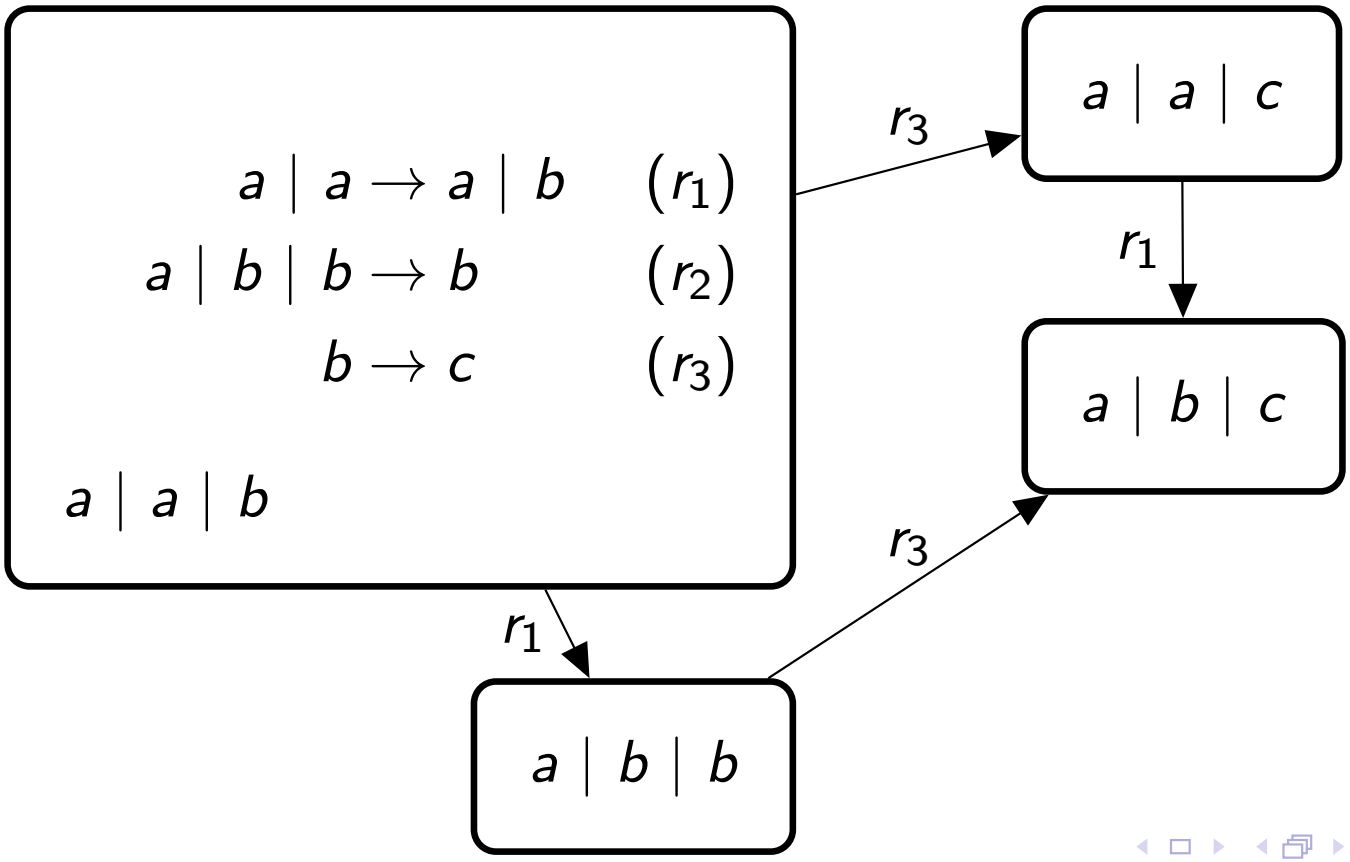
Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



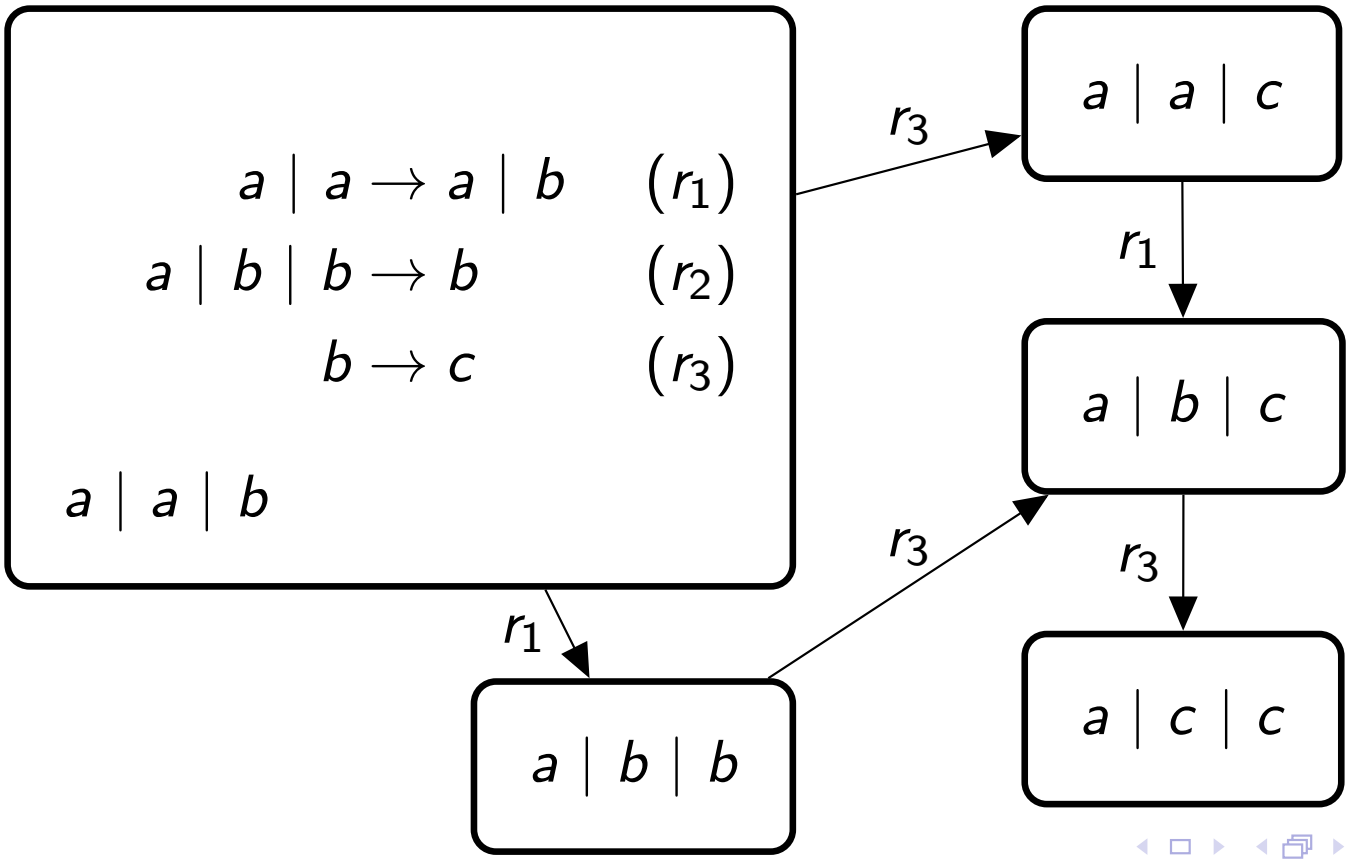
Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



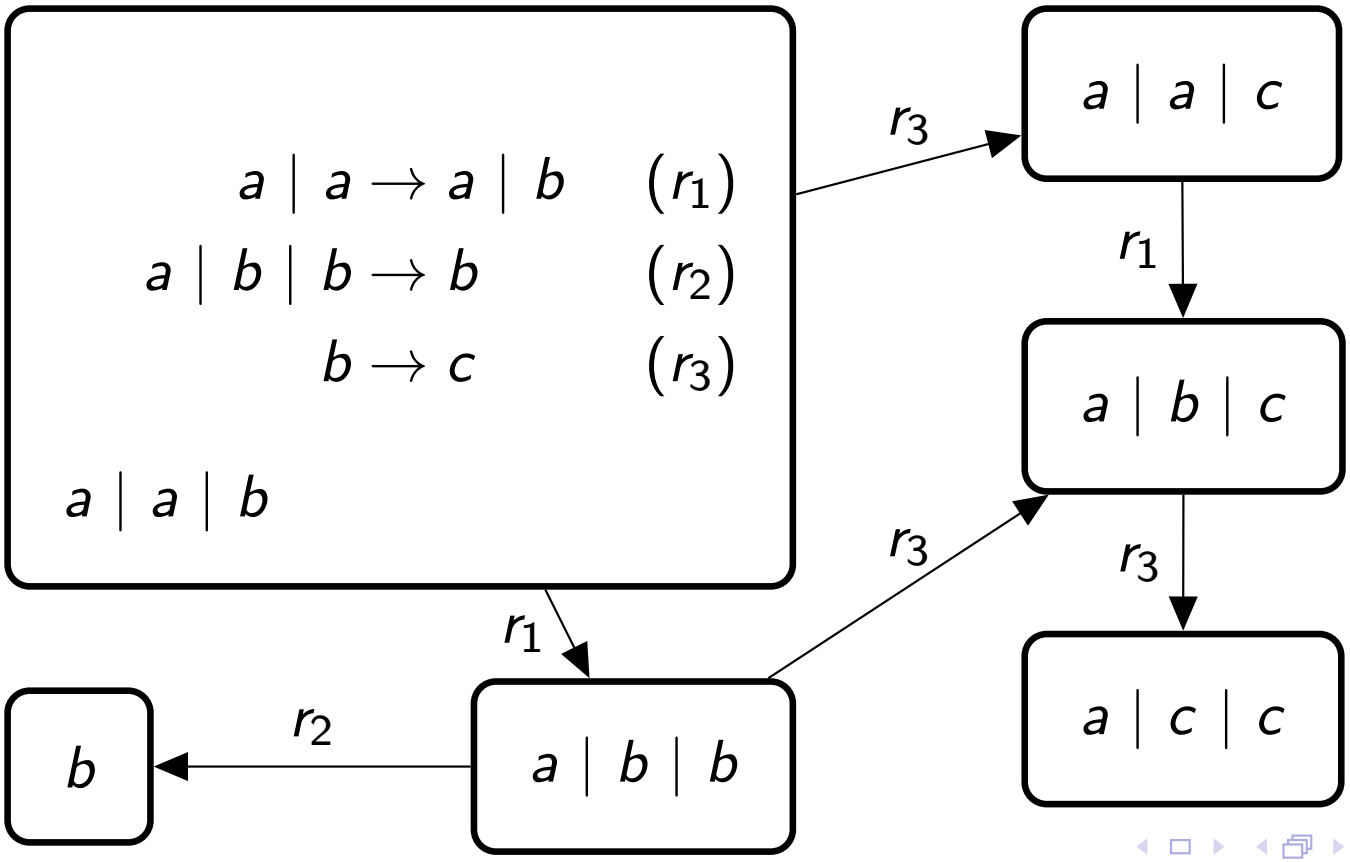
Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



Sequential vs. maximal parallel rule application

- Sequential: in each step apply 1 rule
- Maximal parallel: in each step apply a maximal multiset of rules



Language of a P system

- The result of the computation is a multiset of objects, which is present in a specific membrane at a halting configuration
- The language generated by a P system is a set of results of all possible conputations.

Variants of rules

- cooperative ($a \mid b \mid b \rightarrow b$) (universal [Păun, 1998])

Variants of rules

- cooperative ($a \mid b \mid b \rightarrow b$) (universal [Păun, 1998])
- non-cooperative ($b \rightarrow c$) (PsCF [Sburlan, 2005])

Variants of rules

- cooperative ($a \mid b \mid b \to b$) (universal [Păun, 1998])
- non-cooperative ($b \to c$) (PsCF [Sburlan, 2005])
- non-cooperative with inhibitors ($a \to b \mid_{\neg c,d}$) (PsET0L [Ionescu and Sburlan, 2004])

Variants of rules

- cooperative ($a \mid b \mid b \to b$) (universal [Păun, 1998])
- non-cooperative ($b \to c$) (PsCF [Sburlan, 2005])
- non-cooperative with inhibitors ($a \to b \mid_{\neg c,d}$) (PsET0L [Ionescu and Sburlan, 2004])
- catalytic ($a \mid b \to a \mid c \mid d$)

Variants of rules

- cooperative ($a \mid b \mid b \to b$) (universal [Păun, 1998])
- non-cooperative ($b \to c$) (PsCF [Sburlan, 2005])
- non-cooperative with inhibitors ($a \to b \mid_{\neg c,d}$) (PsET0L [Ionescu and Sburlan, 2004])
- catalytic ($a \mid b \to a \mid c \mid d$)
  - catalytic with 2 catalysts (universal [Freund et al., 2005])
  - with 1 catalyst (open problem)
  - with 1 catalyst and inhibitors (universal [Ionescu and Sburlan, 2004])

Sequential P systems

Sequential P systems with cooperative rules

Sequential P systems

Sequential P systems with cooperative rules

- are equal to VASS $\Rightarrow$ not universal [Ibarra et al., 2005]
- with priorities are universal [Ibarra et al., 2005]
- with unbounded membrane creation are universal [Ibarra et al., 2005]

Sequential P systems

Sequential P systems with cooperative rules

- are equal to VASS $\Rightarrow$ not universal [Ibarra et al., 2005]
- with priorities are universal [Ibarra et al., 2005]
- with unbounded membrane creation are universal [Ibarra et al., 2005]
- **with inhibitors [Kováč, 2014]**

Overview of the simulation for the accepting case

- Simulation of a register machine
- Contents of register j is represented by the multiplicity of the object a_j
- SUB instruction is simulated by inhibitors

Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules

Overview of the simulation for the generating case

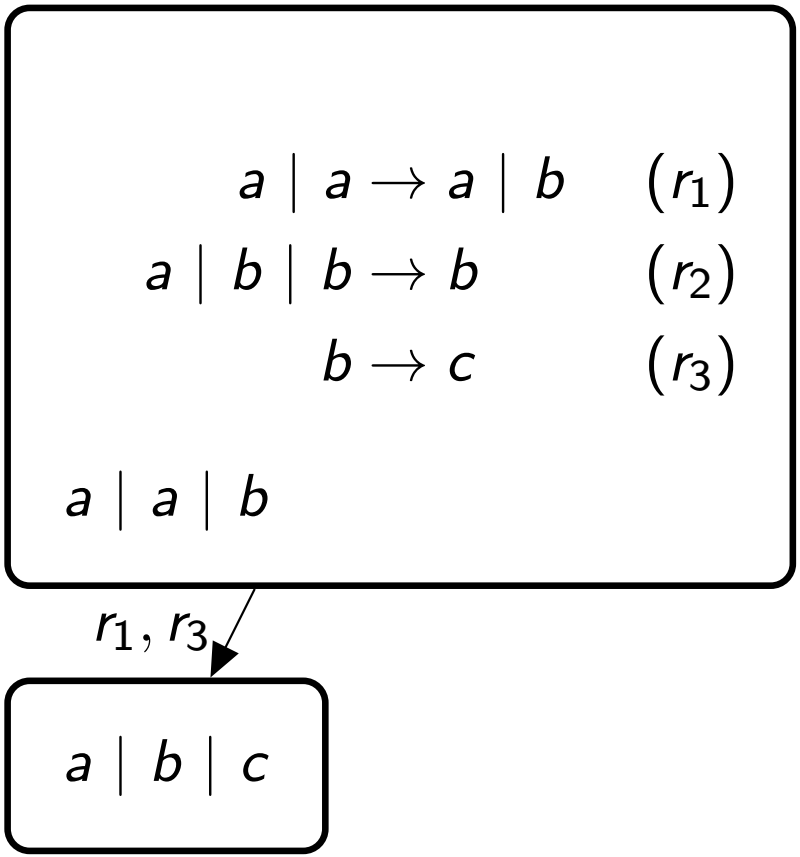
- Simulation of a maximal parallel P system
- Start with the same rules

$$a \mid a \to a \mid b \qquad (r_1)$$
$$a \mid b \mid b \to b \qquad (r_2)$$
$$b \to c \qquad (r_3)$$

$$a \mid a \mid b$$

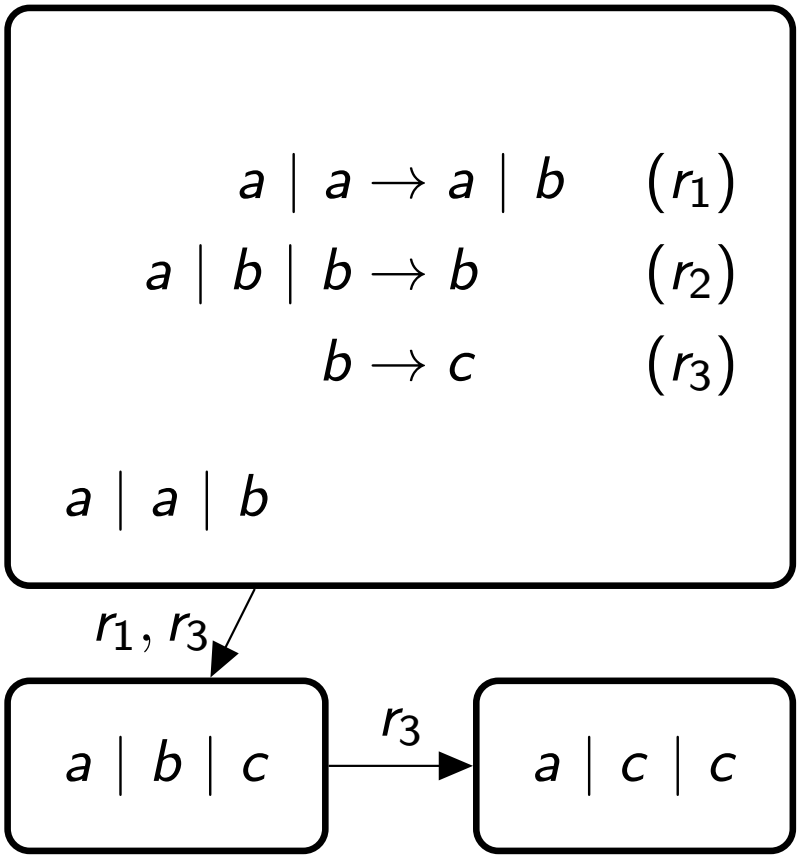
Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



Overview of the simulation for the generating case

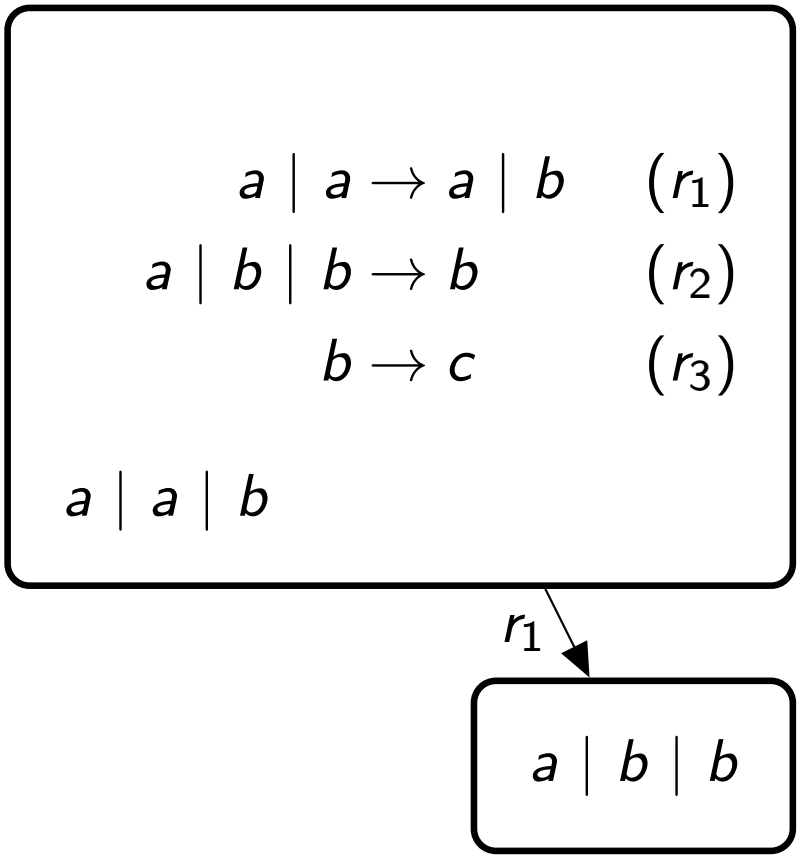
- Simulation of a maximal parallel P system
- Start with the same rules

$$a \mid a \to a \mid b \qquad (r_1)$$
$$a \mid b \mid b \to b \qquad (r_2)$$
$$b \to c \qquad (r_3)$$

$$a \mid a \mid b$$

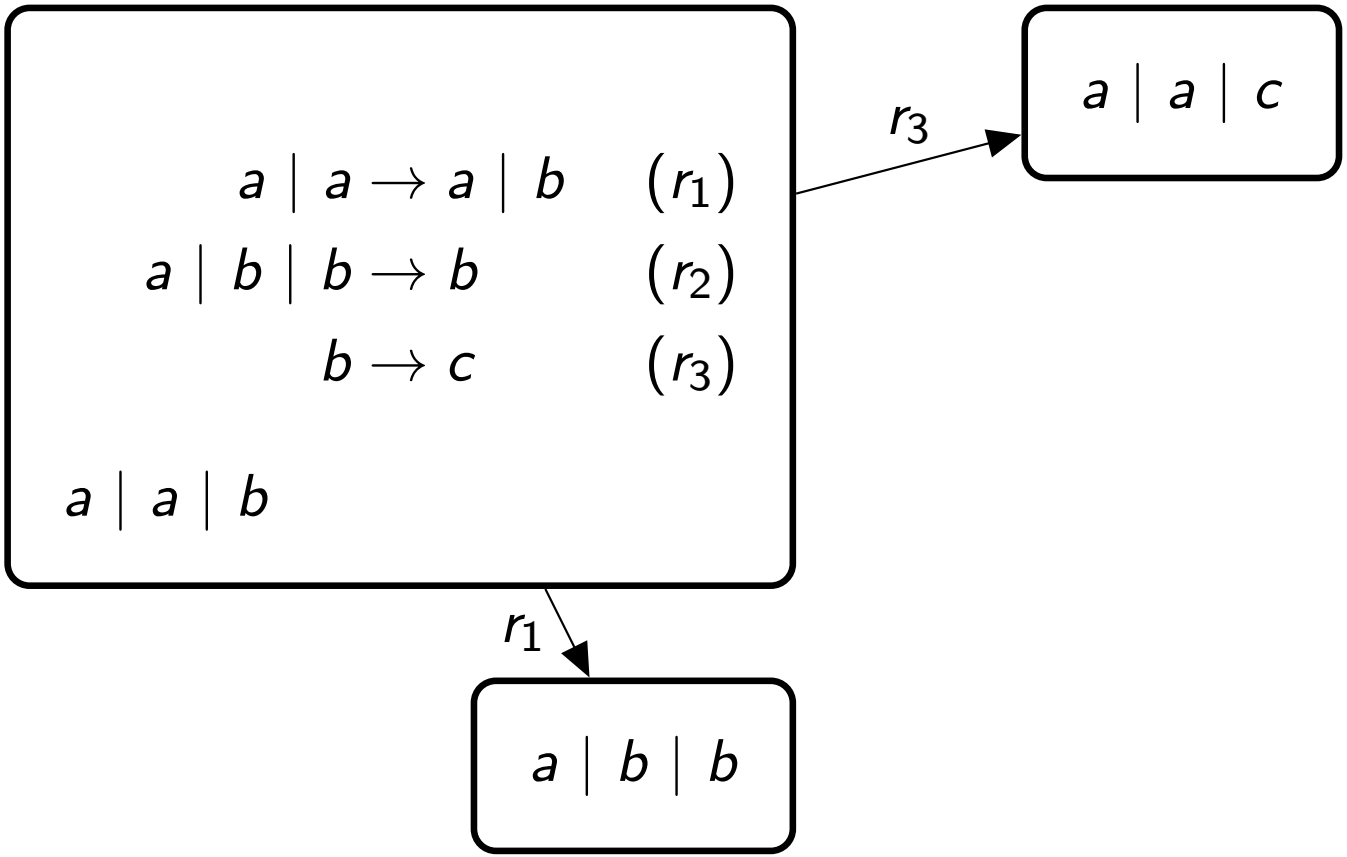
Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



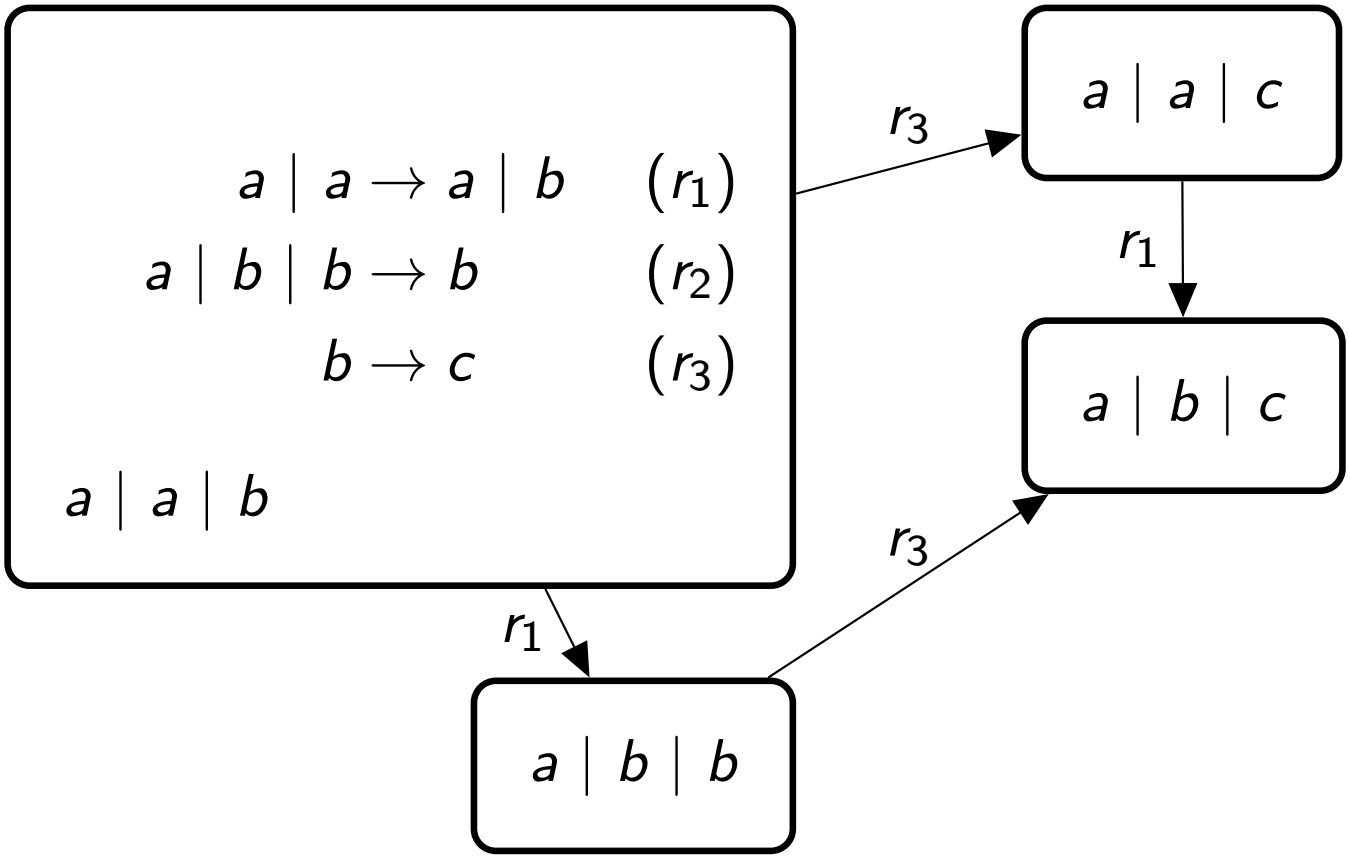
Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



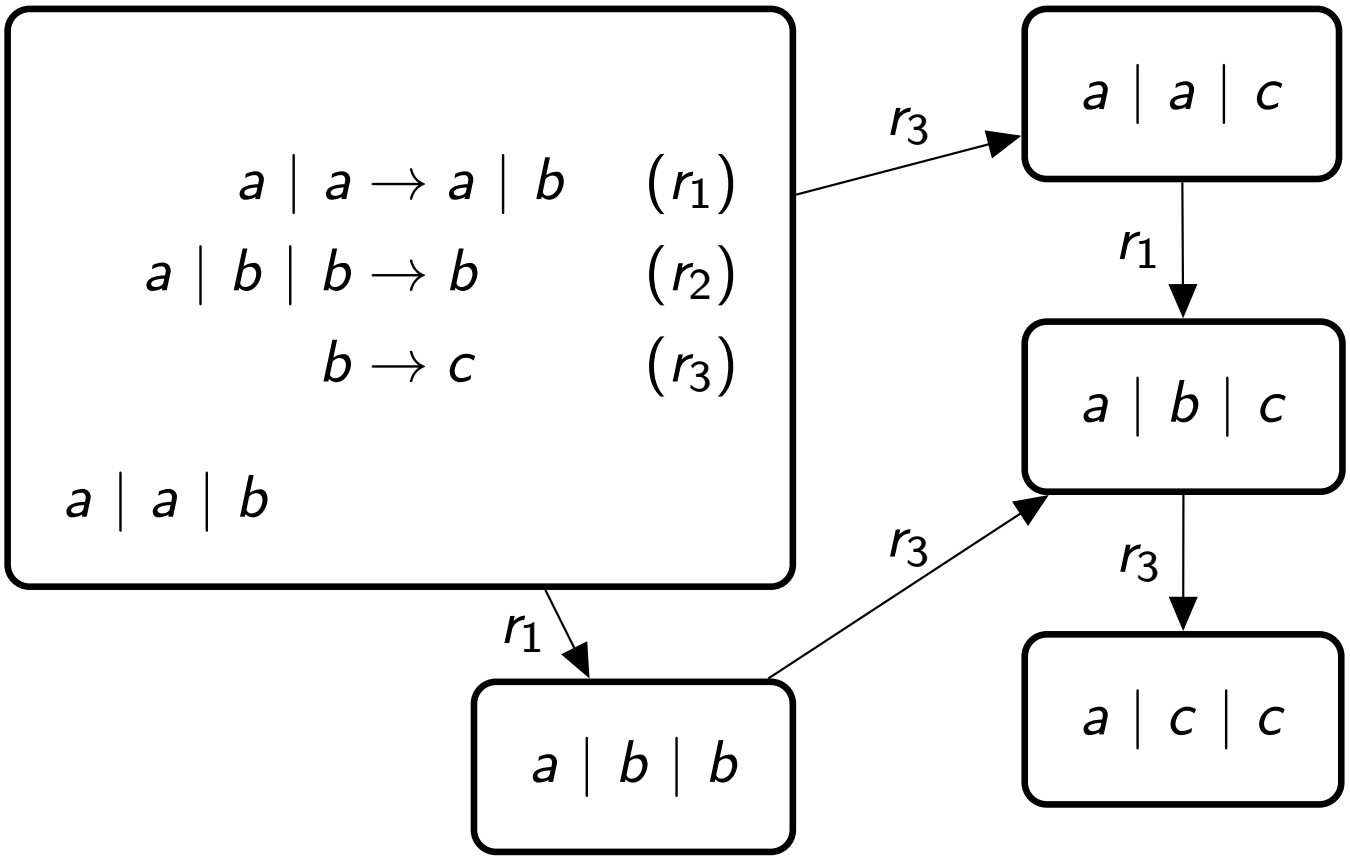
Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



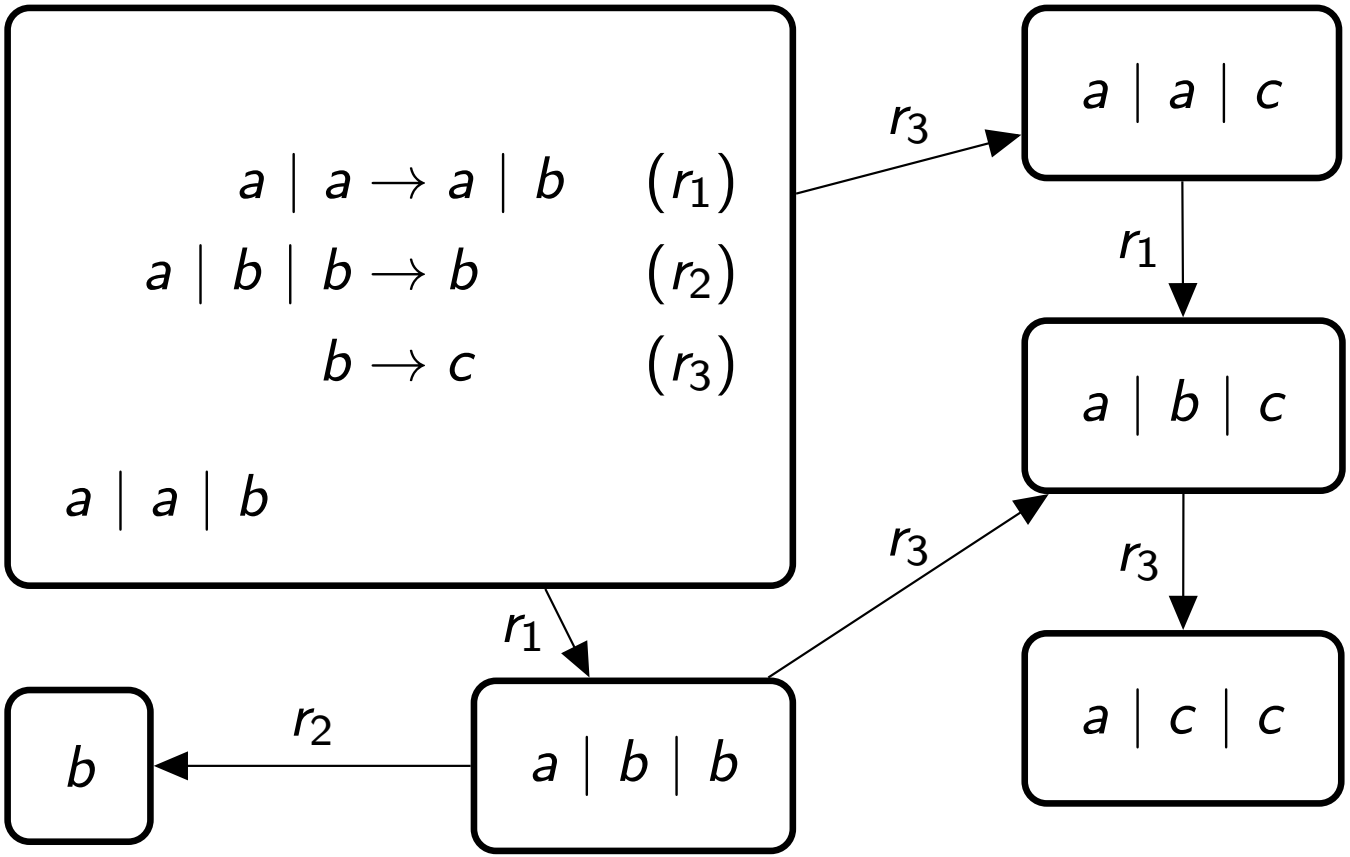
Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



Overview of the simulation for the generating case

- Simulation of a maximal parallel P system
- Start with the same rules



Two phases

- Prevent the rule application on already rewritten objects

Two phases

- Prevent the rule application on already rewritten objects
  - replace objects on the right side a with a'
  - add *RESTORE* phase

Two phases

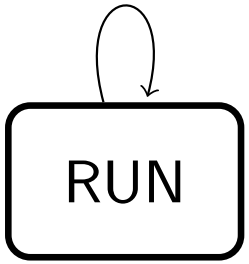
- Prevent the rule application on already rewritten objects
  - replace objects on the right side a with a'
  - add *RESTORE* phase
- $RUN \mid a \mid a \rightarrow RUN \mid a' \mid b'$
- $RUN \mid a \mid b \mid b \rightarrow RUN \mid b'$
- $RUN \mid b \rightarrow RUN \mid c'$

Two phases

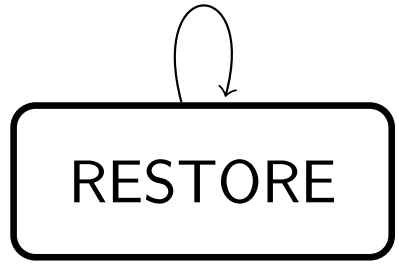
- Prevent the rule application on already rewritten objects
  - replace objects on the right side a with a'
  - add *RESTORE* phase
- $RUN \mid a \mid a \rightarrow RUN \mid a' \mid b'$
- $RUN \mid a \mid b \mid b \rightarrow RUN \mid b'$
- $RUN \mid b \rightarrow RUN \mid c'$
- $RESTORE \mid a' \rightarrow RESTORE \mid a$
- $RESTORE \mid b' \rightarrow RESTORE \mid b$
- $RESTORE \mid c' \rightarrow RESTORE \mid c$

Switching the phases

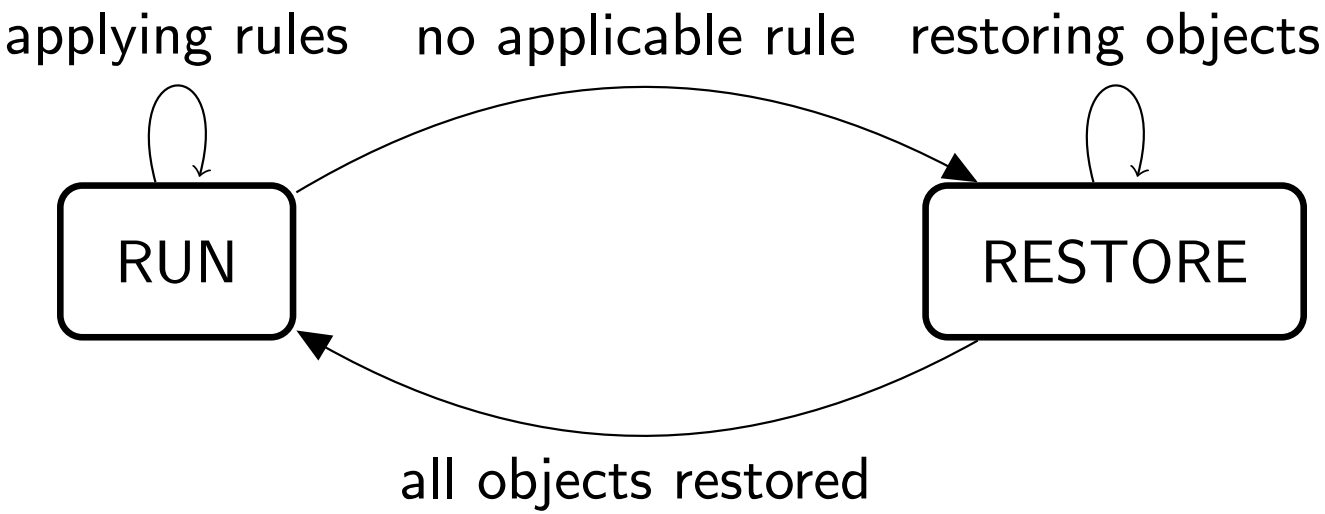
applying rules



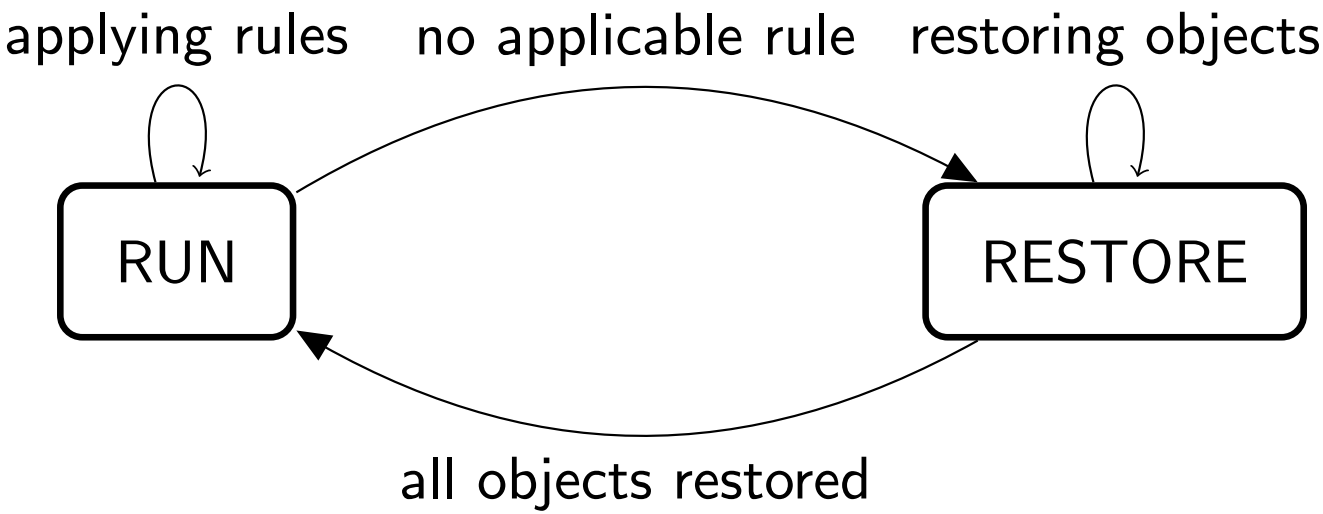
restoring objects



Switching the phases

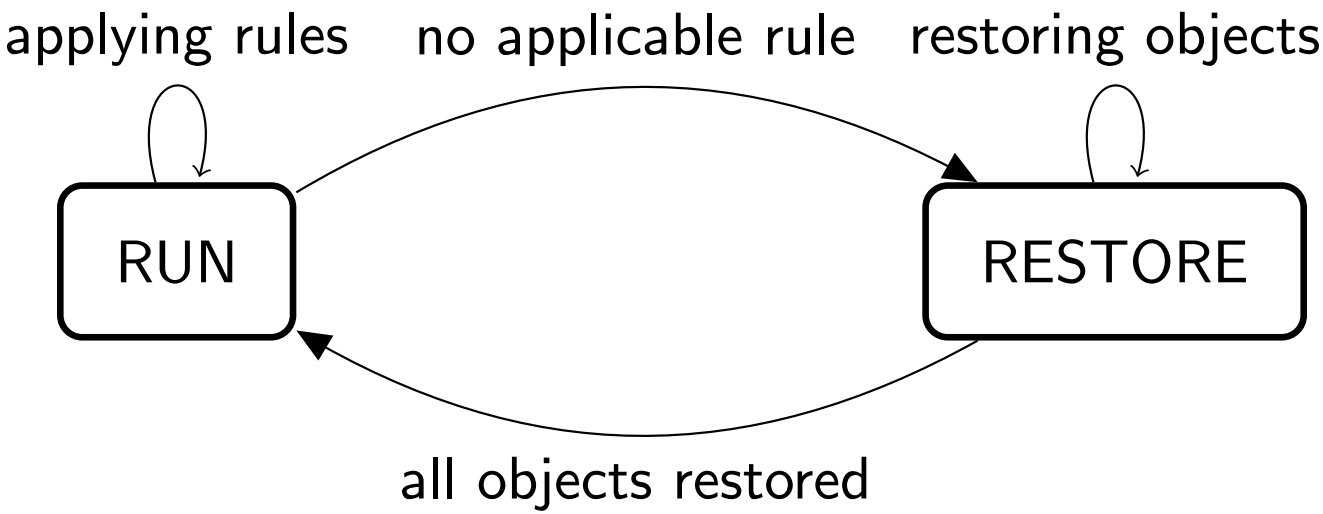


Switching the phases



- $RUN \mid UNUSABLE_1 \mid UNUSABLE_2 \mid UNUSABLE_3 \rightarrow RESTORE$

Switching the phases



- $RUN \mid UNUSABLE_1 \mid UNUSABLE_2 \mid UNUSABLE_3 \rightarrow RESTORE$
- $RESTORE \rightarrow RUN \mid_{\neg a' b' c'}$

Creating UNUSABLE objects (simple case)

- $(3) : b \rightarrow c$
- $RUN \rightarrow RUN \mid UNUSABLE_3 \mid_{\neg b, UNUSABLE_3}$

Creating UNUSABLE objects (complicated case)

- $(1) : a \mid a \rightarrow a \mid b$
- $RUN \rightarrow RUN \mid UNUSABLE_1 \mid_{\neg a, UNUSABLE_1}$
- Wrong for exactly 1 occurence of a

Promoting objects

- $RUN \mid a \rightarrow RUN \mid \dot{a} \mid_{\neg \dot{a}}$
- $RUN \mid b \rightarrow RUN \mid \dot{b} \mid_{\neg \dot{b}}$
- $RUN \mid c \rightarrow RUN \mid \dot{c} \mid_{\neg \dot{c}}$
- At most 1 object can be promoted.

Using promoted objects

- $RUN \mid a \mid \dot{a} \rightarrow a' \mid b'$
- $RUN \mid \dot{a} \mid b \mid b \rightarrow b'$
- $RUN \mid \dot{a} \mid \dot{b} \mid b \rightarrow b'$
- $RUN \mid a \mid \dot{b} \mid b \rightarrow b'$
- $RUN \mid \dot{b} \rightarrow c'$

Using promoted objects

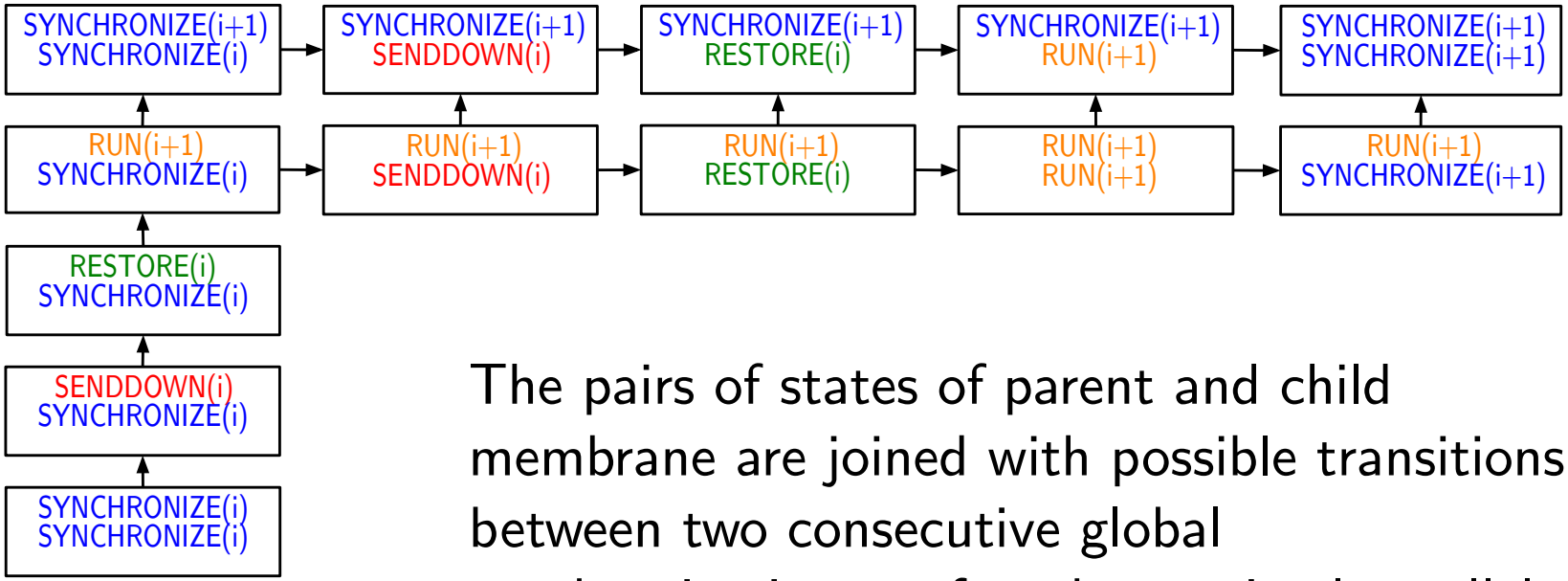
- $RUN \mid a \mid \dot{a} \rightarrow a' \mid b'$
- $RUN \mid \dot{a} \mid b \mid b \rightarrow b'$
- $RUN \mid \dot{a} \mid \dot{b} \mid b \rightarrow b'$
- $RUN \mid a \mid \dot{b} \mid b \rightarrow b'$
- $RUN \mid \dot{b} \rightarrow c'$
- $RUN \rightarrow RUN \mid UNUSABLE_3 \mid_{\neg b, \dot{b}, UNUSABLE_3}$
- $RUN \rightarrow RUN \mid UNUSABLE_1 \mid_{\neg a, UNUSABLE_1}$

Multiple different objects on the left side

- $(2) : a \mid b \mid b \rightarrow b$
- $RUN \rightarrow RUN \mid UNUSABLE_2 \mid_{\neg a, \dot{a}, UNUSABLE_2}$
- $RUN \rightarrow RUN \mid UNUSABLE_2 \mid_{\neg b, UNUSABLE_2}$

Thanks for your attention!

Parent and child membrane phases



The pairs of states of parent and child membrane are joined with possible transitions between two consecutive global synchronizations - after the maximal parallel steps i and i+1

Obr. : Possible pairs of states of parent and child membrane

Snapshot of all membrane states