P Systems: A formalism for computing and for systems biology

Andrea Maggiolo Schettini

Dipartimento di Informatica, Università di Pisa, Italy

Torino – November 16, 2009

Introduction (1)

Natural Computing is the study of

- models of computation
- inspired by the functioning of biological systems

Natural Computing is not Bioinformatics

- Bioinformatics is the development and application of Computer Science means to Biology (and Medicine)

A relevant part of Natural Computing exploits methods and means of formal language theory

- Biological mechanisms can be often suitably described by rewrite rules

Introduction (2)

The main aims of bio-inspired models of computation are:

- to propose new unconventional computing architectures
- (in the context of formal language theory) to study new classes of formal languages
- to propose new programming paradigms

Typical results on bio-inspired models of computation are:

- Universality (that is Turing completeness)
- Existence of polynomial solutions to NP-complete problems (with exponential workspace)
- Connections with other models of computation (Petri Nets, Lambda calculus, etc...)

Outline of the talk

P Systems

P Systems are (a class of) formalisms studied in Membrane Computing

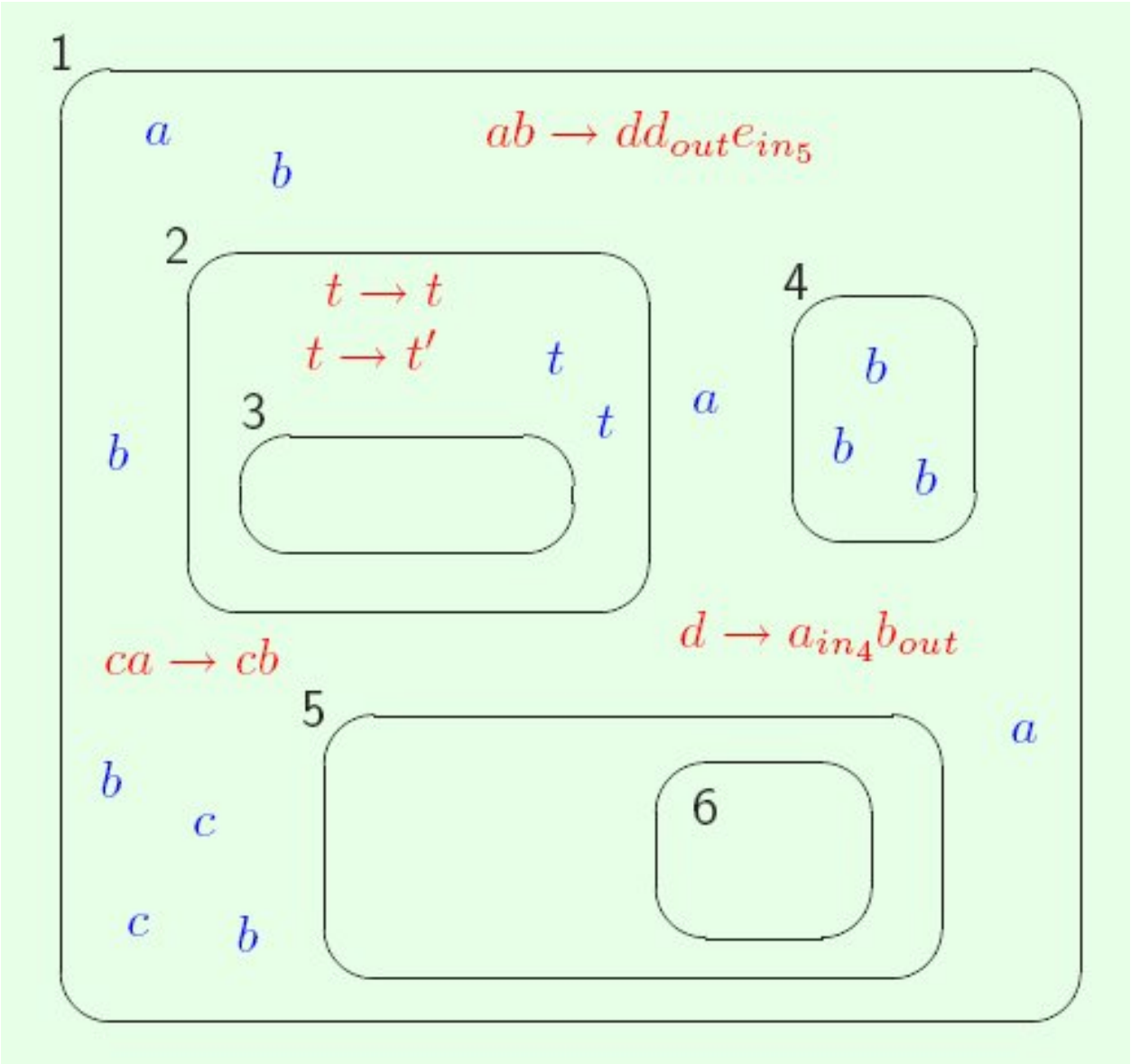
The P Systems web page: `http://ppage.psystems.eu/`

P Systems are distributed computing devices inspired by the structure and the functioning of a living cells.

The key elements of P Systems are:

- Membranes (that create compartments used to distribute computations)
- Multisets (abstractions of chemical solutions that are used as data)
- Evolution (rewriting) rules (abstractions of chemical reactions that are used as programs)

The simplest P Systems



Formal definition of P Systems

A *P System* Π is given by

$$\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$$

where:

- V is an *alphabet* whose elements are called *objects*;
- $\mu \subset \mathbb{N} \times \mathbb{N}$ is a *membrane structure*, such that $(i, j) \in \mu$ denotes that the membrane labeled by j is contained in the membrane labeled by i;
- w_i with $1 \leq i \leq n$ are strings from V^* representing multisets over V associated with the membranes $1, 2, \ldots, n$ of μ;
- R_i with $1 \leq i \leq n$ are finite sets of *evolution rules* associated with the membranes $1, 2, \ldots, n$ of μ.

Evolution rules

An evolution rule $u \longrightarrow v$ consists of a multiset of objects u (representing reactants) and a multiset of messages v (representing products). A message may have one of the following forms:

- a_{here}, meaning that object a remains in the same membrane;
- a_{out}, meaning that object a is sent out of the membrane;
- a_{in_l}, meaning that object a is sent into the child membrane l.

The subscript *here* is often omitted.

Evolution rules can be classified into:

- non-cooperative rules: the left-hand side consists of a single object (e.g. $a \longrightarrow b^2 d_{out}$)
- cooperative rules: the left-hand side can be any multiset of objects (e.g. $a^2 b \longrightarrow b^2 d_{out}$)
  - a particular case of cooperative rules are catalytic rules, namely rules of the form $ca \longrightarrow cb^2$ where c belongs to a special set of objects called catalysts.

Maximal parallelism

Evolution rules are applied with *m*aximal parallelism:

- More than one rule can be applied (on different objects) in the same step
- Each rule can be applied more than once in the same step (on different objects)
- Maximality means that:

  > *A multiset of instances of evolution rules is chosen non–deterministically such that no other rule can be applied to the system obtained by removing all the objects necessary to apply the chosen instances of rules.*

Computation of a P System

A sequence of transitions between configurations of a given P System is called a *computation*.

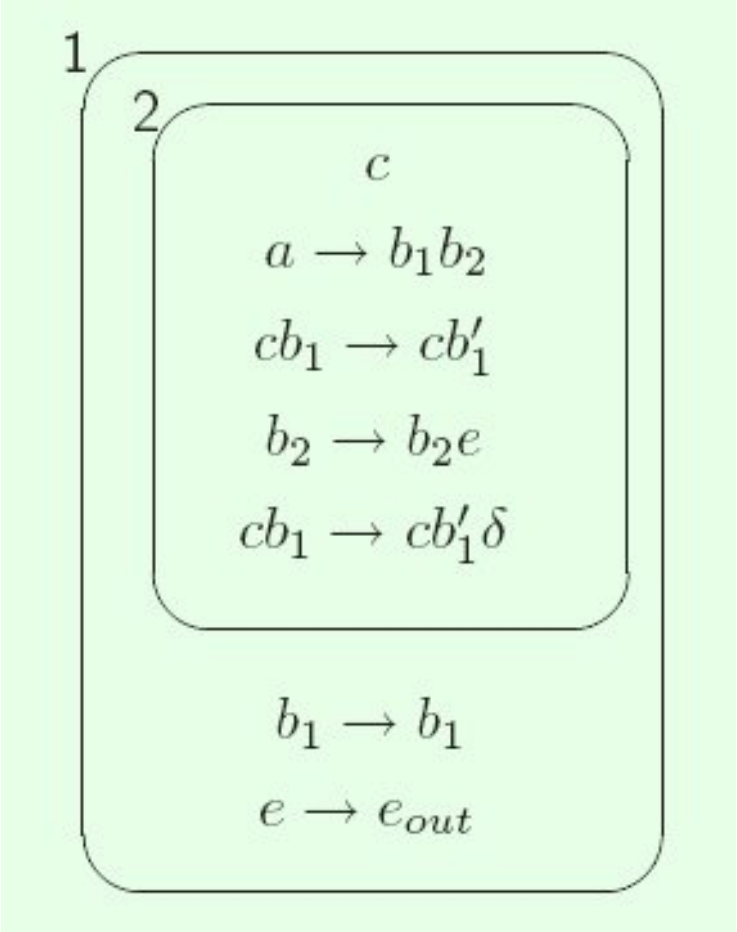
A computation is *successful* if and only if it reaches a configuration in which no rule is applicable.

The result of a successful computation is the multiset of objects sent, during the computation, either out of the skin membrane or into a chosen membrane.

Unsuccessful computations (computations which never halt) yield no result.

Example of P System

A P System computing n^2 (with a dissolving rule)



INPUT: a^n inside membrane 2 OUTPUT: e^{n^2} sent out of membrane 1

Universality of P Systems

P Systems with cooperative rules are universal,

- actually, catalytic rules are enough

P Systems with non-cooperative rules only are not universal

Some variants of P Systems

Variants obtained by considering different types of evolution rules:

- extended with priorities;
- with promoters and inhibitors;
- with dissolution of membranes;
- symport/antiport rules;
- with active membranes;
- .....

Variants obtained by considering graphs rather than trees as membrane structures:

- tissue-like systems;
- spiking neural systems.

P Systems with active membranes

Active membrane means that evolution rules can change the membrane structure of a P System

- In particular, there can be membrane division rules
- $[u]_i \rightarrow [v]_j[z]_k$

It has been proved that with membrane division it is possible to solve NP-complete problems in polynomial time (but exponential space).

Roughly, the idea is the following:

- Every possible solution is encoded inside a different membrane in a linear number of steps by means of membrane division rules
- Each membrane checks whether the solution it contains is correct (in polynomial time, by def.)

Formal semantics of P Systems

Some formal semantics of P Systems have been defined with different aims;

- to develop interpreters proved to be correct (Ciobanu et al.);
- to study causality aspects or P Systems (Busi);

The main difficulty in the definition of a formal semantics is the handling of the maximal parallelism.

Observable Behavior

In order to define reasonable semantics and behavioral equivalences we have to characterize what it is reasonable to observe of the behavior of a P System

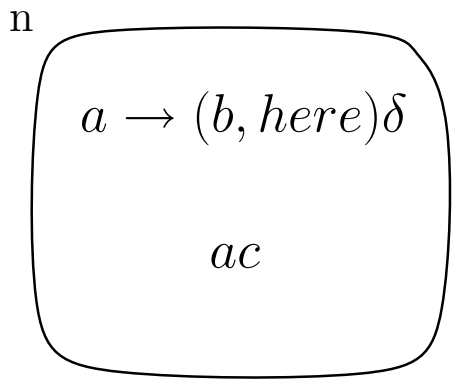
We choose to observe the input/output behavior of membranes:

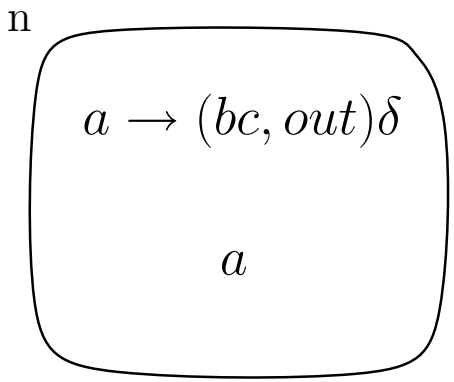
Two membranes are equivalent if, at each step, they can:

- *receive the same objects from outer and inner membranes*
- *send the same objects to the outer membrane (or to the external environment)*
- *send the same objects to the same inner membranes*

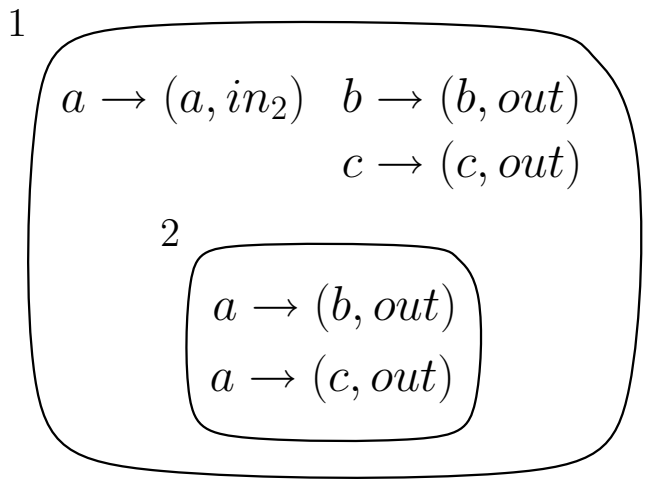
Examples of Equivalent Membranes

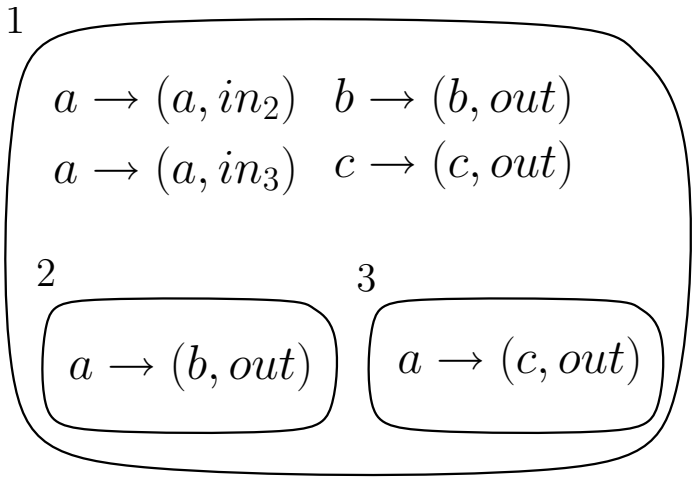
The following membranes could be considered as equivalent:





and also the following two:





Examples of Equivalent Membrane Contents

A *membrane content* is a pair $(\mathcal{R}, u)$ where

- $\mathcal{R}$ is a set of evolution rules
- u is a multiset of objects

that can be (a part of) the content of a membrane

These membrane contents should be considered pairwise equivalent:

$$(\mathcal{R}_1\,,\,\varnothing) \qquad \text{and} \qquad (\mathcal{R}_2\,,\,\varnothing)$$

$$\text{where } \mathcal{R}_1 = \{\quad a \to (b, here)\ ,\ \ a \to (c, here)\quad\} \text{ and}$$
$$\mathcal{R}_2 = \mathcal{R}_1 \cup \{\quad aa \to (bc, here)\quad\}.$$

Outline of the talk

The (simplest) P Algebra

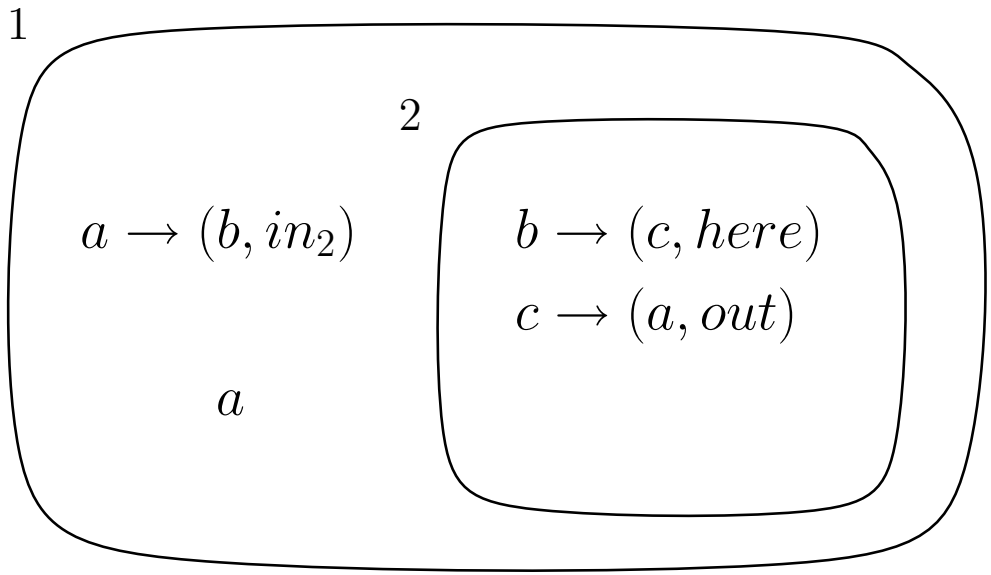
Def. (P Algebra) The syntax of *membrane contents* c, *membranes* m, and *membrane systems* ms is given by the following grammar:

$$c ::= (\varnothing, \varnothing) \mid (u \rightarrow v_h v_o \{v_{l_i}\}, \varnothing) \mid (\varnothing, a) \mid c \cup c$$
$$m ::= [_l c\,]_l \qquad\qquad ms ::= m \mid ms|ms \mid \mu(m, ms)$$

where l and l_i range over $\mathbb{N}$ and a ranges over V.

- $u \rightarrow v_h v_o \{v_{l_i}\}$ stands for $u \rightarrow (v_h, here)(h_o, out)(v_{l_1}, in_{l_1}) \ldots (v_{l_n}, in_{l_n})$
- $c_1 \cup c_2$ denotes the membrane content obtained by merging the rules and the objects of c_1 and c_2
- $[_l c]_l$ denotes a membrane whose content is c and whose label is l
- $ms|ms$ denotes *juxtaposition* of membranes
- $\mu(m, ms)$ denotes the containment of the membranes ms in m (hierarchical composition)

Example of Term of the P Algebra



corresponds to the term:

$$\mu\big(\ [_1\ (a \to (b, in_2), \varnothing)\ \cup\ (\varnothing, a)\ ]_1\ ,\\ [_2\ (b \to (c, here), \varnothing)\ \cup\ (c \to (a, out), \varnothing)\ ]_2\ \big)$$

that is (for short):

$$\mu\big(\ [_1\ a \to (b, in_2)\ ,\ a\ ]_1\ ,\ [_2\ b \to (c, here)\ ,\ c \to (a, out)\ ]_2\ \big)$$

Semantics of the P Algebra

The semantics of the P Algebra is a Labeled Transition System (LTS)

- states are terms
- transitions are labeled by information about the input/output behavior of the system (observation)

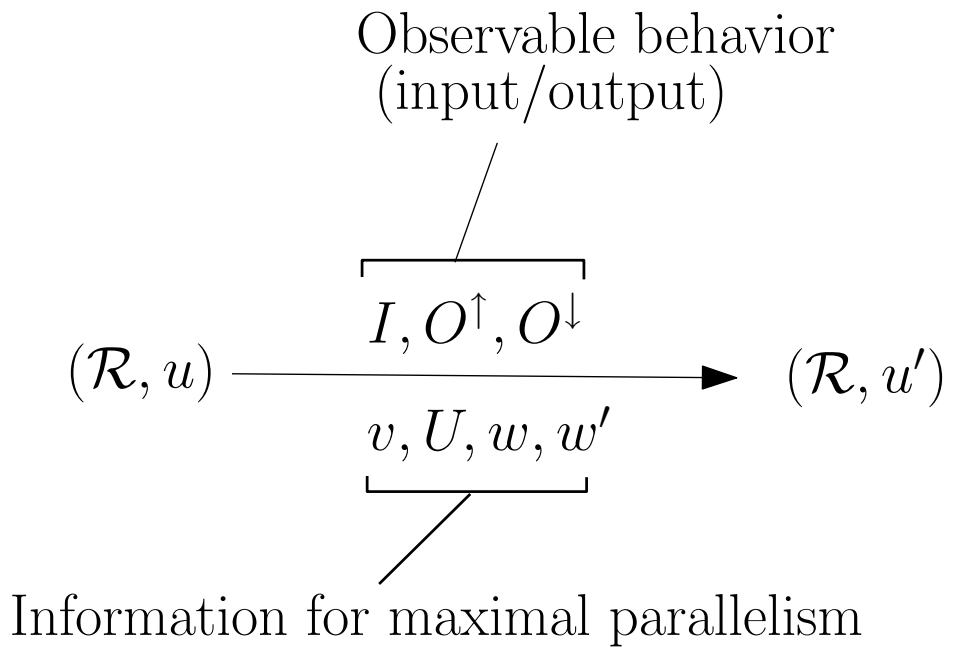
Let's start with membrane contents. We would like to

- define the behavior of individual evolution rules and objects
- infer the behavior of a membrane content from the behaviors of its rules and objects

Problem: it is hard to express the concept of maximal parallelism in a compositional way

Solution: we enrich transition labels with information concerning the (potential) application and non application of evolution rules

Transitions of Membrane Contents



In particular:

- v and w are objects consumed by some evolution rules, as it results from the composition of the transitions of single rules and of single objects, respectively
- w' are objects that are not consumed by any rule
- U contains all left hand sides of rules in $\mathcal{R}$

Inference Rules for Membrane Contents

The empty membrane content can only receive input objects:

$$\frac{I \in V^*}{(\varnothing, \varnothing) \xrightarrow[\varnothing,\varnothing,\varnothing,\varnothing]{\varnothing,I,\varnothing,\varnothing} (\varnothing, I)}$$

A single object can either be used by some rule (thus disappearing) or not:

$$\frac{I \in V^*}{(\varnothing, a) \xrightarrow[\varnothing,\varnothing,a,\varnothing]{I,\varnothing,\varnothing} (\varnothing, I)} \qquad \frac{I \in V^*}{(\varnothing, a) \xrightarrow[\varnothing,\varnothing,\varnothing,a]{I,\varnothing,\varnothing} (\varnothing, Ia)}$$

A single rule can be applied n times (possibly $n = 0$):

$$\frac{I \in V^* \quad n \in \mathbb{N}}{(u \to v_h v_o \{v_{l_i}\}, \varnothing) \xrightarrow[u^n,\{u\},\varnothing,\varnothing]{I,v_o^n,\{(l_i,v_{l_i}^n)\}} (u \to v_h v_o \{v_{l_i}\}, I v_h^n)}$$

Inference Rules for Unions of Membrane Contents

$$\frac{x_1 \xrightarrow[u_1, U_1, v_1, v_1']{I_1, O_1^{\uparrow}, O_1^{\downarrow}} y_1 \qquad x_2 \xrightarrow[u_2, U_2, v_2, v_2']{I_2, O_2^{\uparrow}, O_2^{\downarrow}} y_2 \qquad v_1' v_2' \not\vdash U_1 \oplus U_2}{x_1 \cup x_2 \xrightarrow[u_1 u_2, U_1 \oplus U_2, v_1 v_2, v_1' v_2']{I_1 I_2, O_1^{\uparrow} O_2^{\uparrow}, O_1^{\downarrow} \cup_{\mathbb{N}} O_2^{\downarrow}} y_1 \cup y_2}$$

This inference rule simply makes the union of each pair of label elements taken from transitions membrane contents x_1 and x_2.

- $u_1 u_2$ is the union of multisets u_1 and u_2, whereas $\cup_{\mathbb{N}}$ and $\oplus$ are other special union operations

Condition $v_1' v_2' \not\vdash U_1 \oplus U_2$ means that no rule could be applied to the objects left unchanged ($v_1' v_2'$) must not

- $v \vdash U$ means $\exists u.(u \subseteq v \wedge u \in U)$

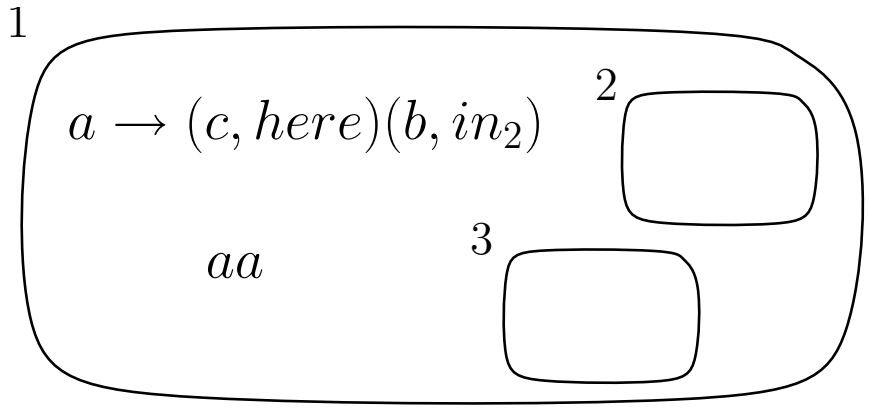
Inference Rules for Membranes and Membrane Systems

When the $[_l \_]_l$ is applied to a membrane content x

- the transition from x to y must be *acceptable*
  - the first and the third label under the arrow must be the same
- information under the arrow is no longer necessary

$$\frac{x \xrightarrow[u,U,u,v']{I,O^{\uparrow},O^{\downarrow}} y}{[_l x]_l \xrightarrow{\{(I,I)\},O^{\uparrow},O^{\downarrow}} [_l y]_l}$$

Inference rules for Membrane Systems (i.e. juxtaposition and hierarchical composition operations) are trivial.



$$(a \to c_{here}\, b_{in_2}\ ,\ aa) \xrightarrow[aa,\{a\},aa,\varnothing]{I,\varnothing,(2,bb)} (a \to c_{here}\, b_{in_2}\ ,\ Icc)$$

$$[_1\,(a \to c_{here}\, b_{in_2}\ ,\ aa)\,]_1 \xrightarrow{\{(1,I)\},\varnothing,(2,bb)} [_1\,(a \to c_{here}\, b_{in_2}\ ,\ Icc)\,]_1$$

$$[_2(\varnothing, \varnothing)]_2 \xrightarrow{\{(2,bb)\},\varnothing,\varnothing} [_2(\varnothing, bb)]_2 \qquad [_3(\varnothing, \varnothing)]_3 \xrightarrow{\{(3,\varnothing)\},\varnothing,\varnothing} [_3(\varnothing, \varnothing)]_3$$

$$[_2(\varnothing, \varnothing)]_2 \mid [_3(\varnothing, \varnothing)]_3 \xrightarrow{\{(2,bb),(3,\varnothing)\},\varnothing,\varnothing} [_2(\varnothing, bb)]_2 \mid [_3(\varnothing, \varnothing)]_3$$

$$\mu([_1 \ldots]_1, [_2 \ldots]_2 \mid [_3 \ldots]_3) \xrightarrow{\{(1,I)\},\varnothing,\varnothing} \mu([_1(\_, Icc)]_1, [_2(\varnothing, bb)]_2 \mid [_3(\varnothing, \varnothing)]_3)$$

Maximal Parallelism Theorem

The multiset of objects that are left unchanged in an acceptable transition of a membrane content cannot be used to perform an acceptable transition with the same rules

- i.e. there is no rule that can be applied to unused objects

Theorem (Maximality)

$$(\mathcal{R}, u) \xrightarrow[u', U, u', v']{I, O_1^{\uparrow}, O_1^{\downarrow}} x \qquad \text{implies} \qquad (\mathcal{R}, v') \not\xrightarrow[u'', U', u'', v'']{I', O_2^{\uparrow}, O_2^{\downarrow}}$$

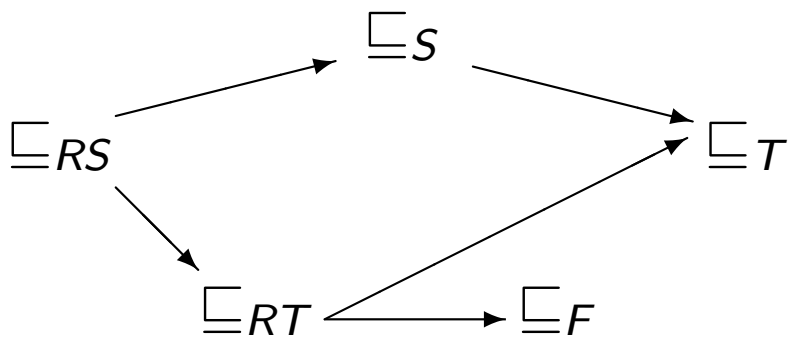
for any $u'' \neq \varnothing$

Well–known Behavioral Preorders

We have considered a number of behavioral preorders:

- *simulation* $\sqsubseteq_S$
- *ready simulation* $\sqsubseteq_{RS}$
- *ready trace preorder* $\sqsubseteq_{RT}$
- *failure preorder* $\sqsubseteq_F$
- *trace preorder* $\sqsubseteq_T$

It is well–known that the considered preorders are structured as follows (where $\longrightarrow$ is $\subseteq$)



In the case of the P Algebra all the inclusions are strict

Well–known Behavioral Equivalences

The kernels of the preorders (the largest equivalence each of them contains) are well–known behavioral equivalences

- bisimulation $\approx$ is the kernel of $\sqsubseteq_S$
- trace equivalence $\approx_T$ is the kernel of $\sqsubseteq_T$

The inference rules of the semantics of the P Algebra satisfy *de Simone* format.

Theorem All of the considered preorders are precongruences

Corollary All of the considered equivalences are congruences

Diagnosable properties

The presence of a desease in a biological system is usually deduced from observation of markers (e.g. the high/low expression of some proteins, the presence/absence of some substances, etc...)

We have defined a notion of *diagnosability* on the semantics of P systems that allows us to say on a model when a disease (expressed as a logical formula) can be deduced from the observation of specific markers

Diagnosability is defined by exploiting the general systems security concept of *opacity*

Opacity and Diagnosability

Let $\mathcal{T}$ be the set of all traces of the semantics of P systems.

Definition. Given a set of observables Θ, an *observation function* is any function $\mathcal{O} : \mathcal{T} \to \Theta^*$.

Definition. A predicate ϕ on system traces is *opaque* w.r.t. the observation function $\mathcal{O}$ and P system P if for every trace w of P such that $\phi(w)$ holds, there exists a trace w' such that $\neg\phi(w')$ holds and $\mathcal{O}(w) = \mathcal{O}(w')$.

Definition. A predicate ϕ is *diagnosable* w.r.t. the observation function $\mathcal{O}$ and P system P if the predicate ϕ is not opaque w.r.t. the observation function $\mathcal{O}$ and P system P.

Diagnosability is in general undecidable, hence the interest of particular classes for which it is decidable

Outline of the talk

Timed P systems

Defined by Cavaliere and Sburlan

Each evolution rule is enriched with a natural number representing the time (number of steps) needed by the rule to be enterely executed

When a rule with time n is applied

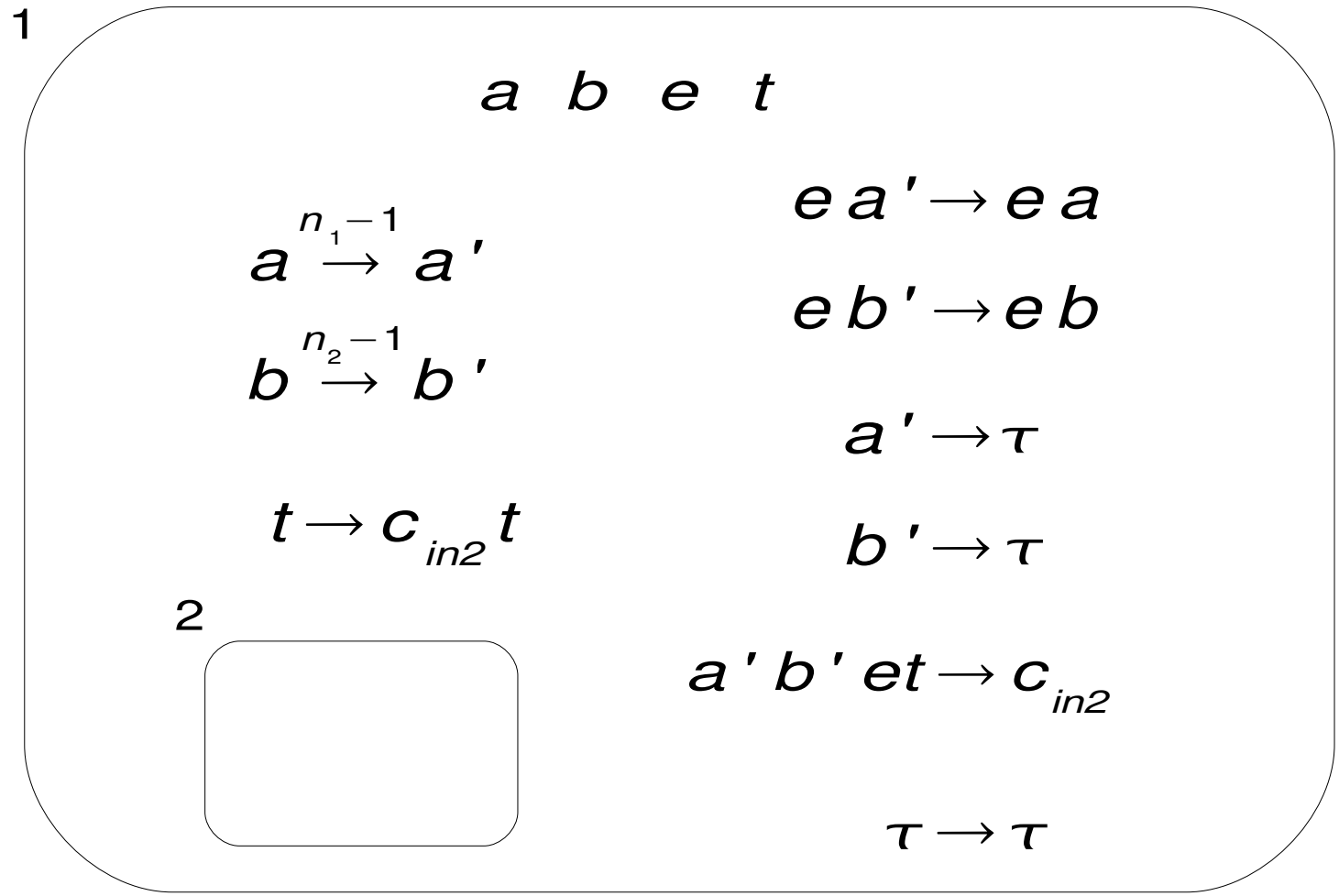
- consumed objects are immediately removed
- produced objects appear after n steps

Semantically, for each application of an evolution rule $u \xrightarrow{n} v$ in membrane i, a *pending rule* $\xrightarrow{n-1}_i v$ is created. At each step:

- every $\xrightarrow{k}_i v$ with $k > 1$ becomes $\xrightarrow{k-1}_i v$
- every $\xrightarrow{1}_i v$ is deleted and objects v are added to the content of membrane i

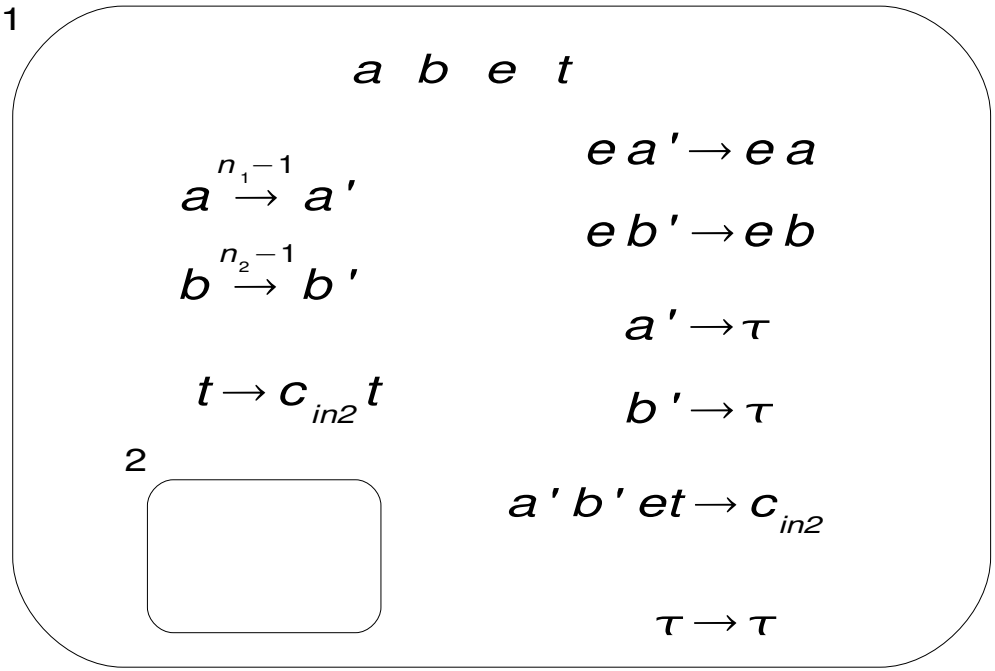
Example of timed P system

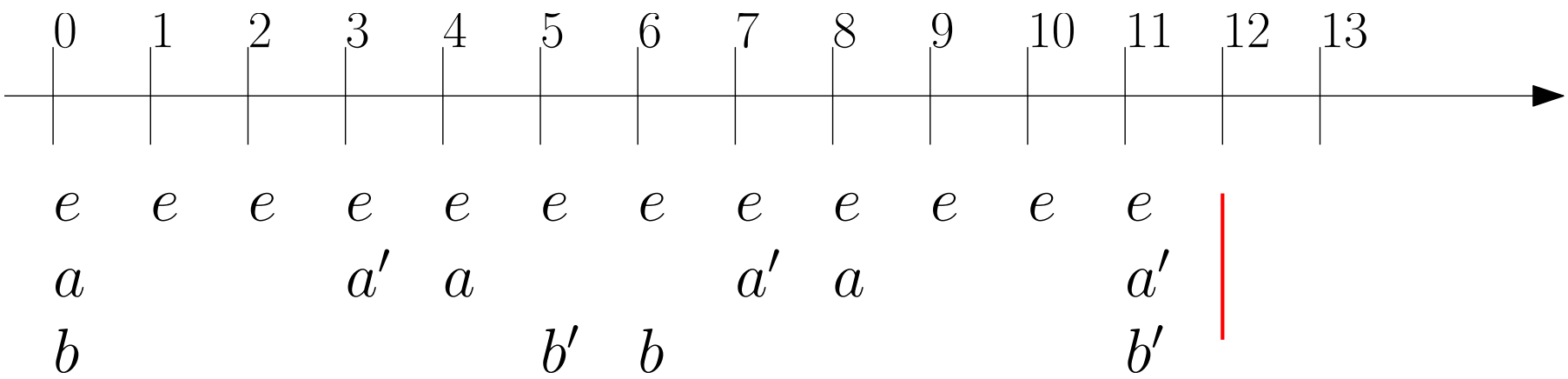
A timed P system computing the least common multiple of n_1 and n_2.



Running the example

Let $n_1 = 4$ and $n_2 = 6$...





Timed P systems: definition

Definition A *timed P system* Π is a tuple

$$\langle V, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m \rangle$$

where

- V is an alphabet whose elements are called *objects*.
- μ is a membrane structure consisting of a hierarchy of m membranes labelled by $1, 2, \ldots, m$. The skin membrane is labelled by 1.
- w_i $(i = 1, 2, \ldots, m)$ is a string of V^* representing a multisets of objects enclosed in membrane i.
- R_i $(i = 1, 2, \ldots, m)$ is a finite set of *timed evolution rules* associated with the membrane i. The rules are of the form $u \xrightarrow{n} v$, $n \in \mathbb{N}, u \in V^+$, and $v \in \{a_{here}, a_{out}, a_{in_j} \mid a \in V, 1 \leq j \leq m\}^*$.

Timed P systems: definition

Definition A multiset of *pending rules* $\mathcal{U}$ is a multiset of elements of the form $\xrightarrow{k}_i v$, with $k > 0$

Definition A *configuration* is a pair $(\Pi, \mathcal{U})$ where Π is a timed P system and $\mathcal{U}$ is a multiset of pending rules

A computation performed by a timed P system Π can be described as a sequence of steps between configurations

- the initial configuration is $(\Pi, \varnothing)$

A step of a timed P system

Definition A configuration $(\Pi, \mathcal{U})$ can perform a *timed P step* $\xrightarrow{1}$ to a configuration $(\Pi', \mathcal{U}')$ if and only if:

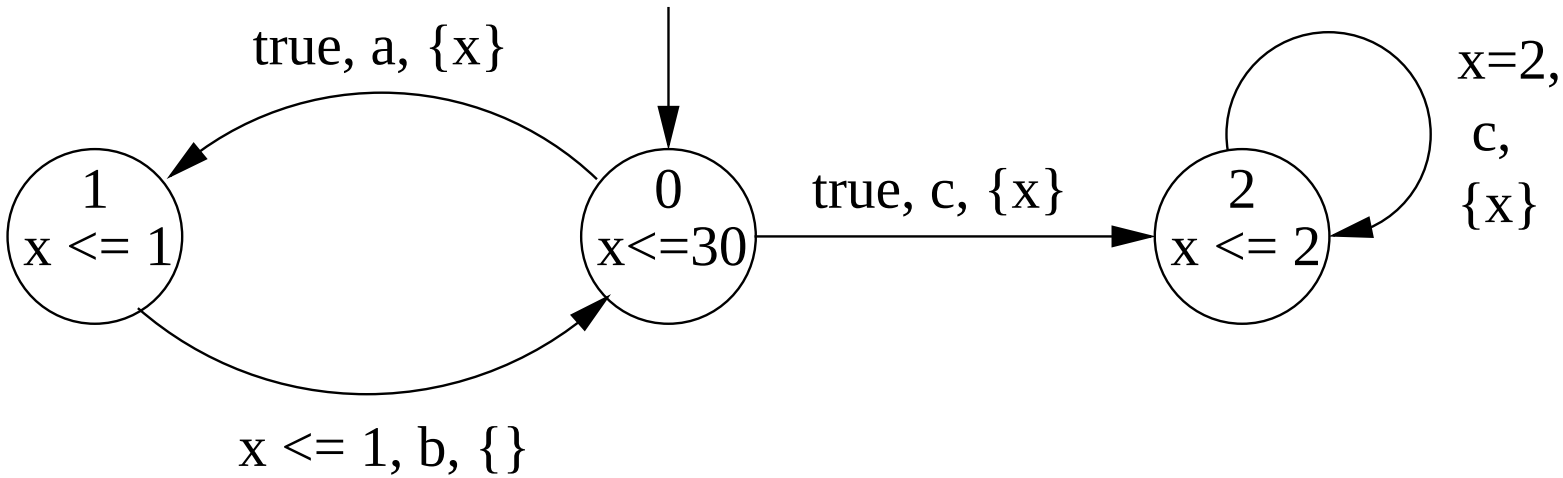
- Π' is a timed P system resulting from an evolution step of Π using maximal parallelism, where:
  - the effects of the rules $u \xrightarrow{1} v$ are visible in Π', i.e., the reactants have disappeared and the products of the rules are available
  - the effects of the rules $u \xrightarrow{n} v$ with $n > 1$ are half visible in Π'. More precisely, the reactants have disappeared, but the products are not yet available
  - for every element $\xrightarrow{1}_i v$ in $\mathcal{U}$, the objects v are added to membrane i;
- $\mathcal{U}'$ is the multiset union of
  - the multiset of all elements $\xrightarrow{k-1}_i v$ derived from all elements $\xrightarrow{k}_i v$, $k > 1$, in $\mathcal{U}$; and
  - the multiset of all elements $\xrightarrow{n-1}_i v$, $n > 1$, representing that an instance of a timed evolution rule $u \xrightarrow{n} v \in R_i$, for some i, has been fired in the evolution step of Π.

Timed Automata

We shall extend timed P systems with features from timed automata

A timed automaton is a finite state automaton extended with:

- clocks
- transitions enriched with conditions on the value of clocks and with clock reset actions
- state invariants



Timed P automata

A timed P automaton is a timed automaton with a discrete time domain in which each location is associated with a timed P system

- all timed P systems must have the same membrane structure

A computation starts in the timed P system associated with the inital location of the automaton

- after each step, clocks are increased by one

When clocks reach values that satisfy the constraint of an outgoing transition, such a transition might be fired

- the computation in the current location is stopped
- objects are moved to the location reached by the transition (in the corresponding membranes)
- some objects might be added to/removed from the skin membrane

Timed P automata: definition

Definition A *timed P automaton* is a tuple

$$T = \langle Q, \Sigma, q_0, \mathcal{E}, \mathcal{X}, F, \mathcal{R}, Inv \rangle$$

where:

- Q is a finite set of locations
- Σ is a finite alphabet of symbols
- q_0 is the initial location
- $\mathcal{E}$ is a finite set of edges
- $\mathcal{X}$ is a finite set of clocks
- $F = \langle V, \mu \rangle$ is a *timed P frame*: it contains the alphabet and the membrane structure shared by all the timed P systems
- $\mathcal{R}$ is a function assigning to every $q \in Q$ a set of sets of timed evolution rules
- Inv is a function assigning to every $q \in Q$ an *invariant*

Timed P automata: definition

Each edge is a tuple $(q, \psi, u, \gamma, \sigma, v, q')$ where:

- q is the source location
- ψ is the clock constraint
- u are the objects removed from the skin membrane
- γ is the clock reset set
- σ is a label (optional – can be used to accept languages)
- v are the objects added to the skin membrane
- q' is the target location

A *state* of execution of a timed P automaton is a tuple $\langle q, \nu, \Pi, \mathcal{U} \rangle$, where:

- q is a location
- ν is a clock valuation
- Π is the executing timed P systems
- $\mathcal{U}$ is a multiset of pending rules

Timed P automata: semantics

The behaviour of a timed P automaton is described by the labelled transition system given by the following rules:

$$\text{T1} \qquad \frac{\nu + 1 \models Inv(q) \qquad (\Pi, \mathcal{U}) \xrightarrow{1} (\Pi', \mathcal{U}')}{\langle q, \nu, \Pi, \mathcal{U} \rangle \xrightarrow[TP]{1} \langle q, \nu + 1, \Pi', \mathcal{U}' \rangle}$$

$$\text{T2} \qquad \frac{\begin{array}{c} \Pi = \langle V, \mu, w_1, w_2, \ldots, w_m, \mathcal{R}(q) \rangle \\ (q, \psi, u, \gamma, \sigma, v, q') \in \mathcal{E}, \quad \nu \models \psi, \quad u \subseteq w_1 \quad w_1' = (w_1 \setminus u) \cup v \\ \Pi' = \langle V, \mu, w_1', w_2, \ldots, w_m, \mathcal{R}(q') \rangle \end{array}}{\langle q, \nu, \Pi, \mathcal{U} \rangle \xrightarrow[TP]{\sigma} \langle q', \nu \setminus \gamma, \Pi', \mathcal{U} \rangle}$$

Timed P automata: results

A computation of a timed P automaton is *valid* (gives an output) only if

- the automaton reaches a location that is never left
- the timed P system associated with such a location halts
- the multiset of pending rules is empty

The output is the multiset of objects left in the skin membrane

Timed P automata are universal (they allow cooperative rules to be used)

- We have proved that universality holds also with non-cooperative rules

The saddleback

We have found a model for guiding the reintroduction of extirpated birds in New Zealand mainland.

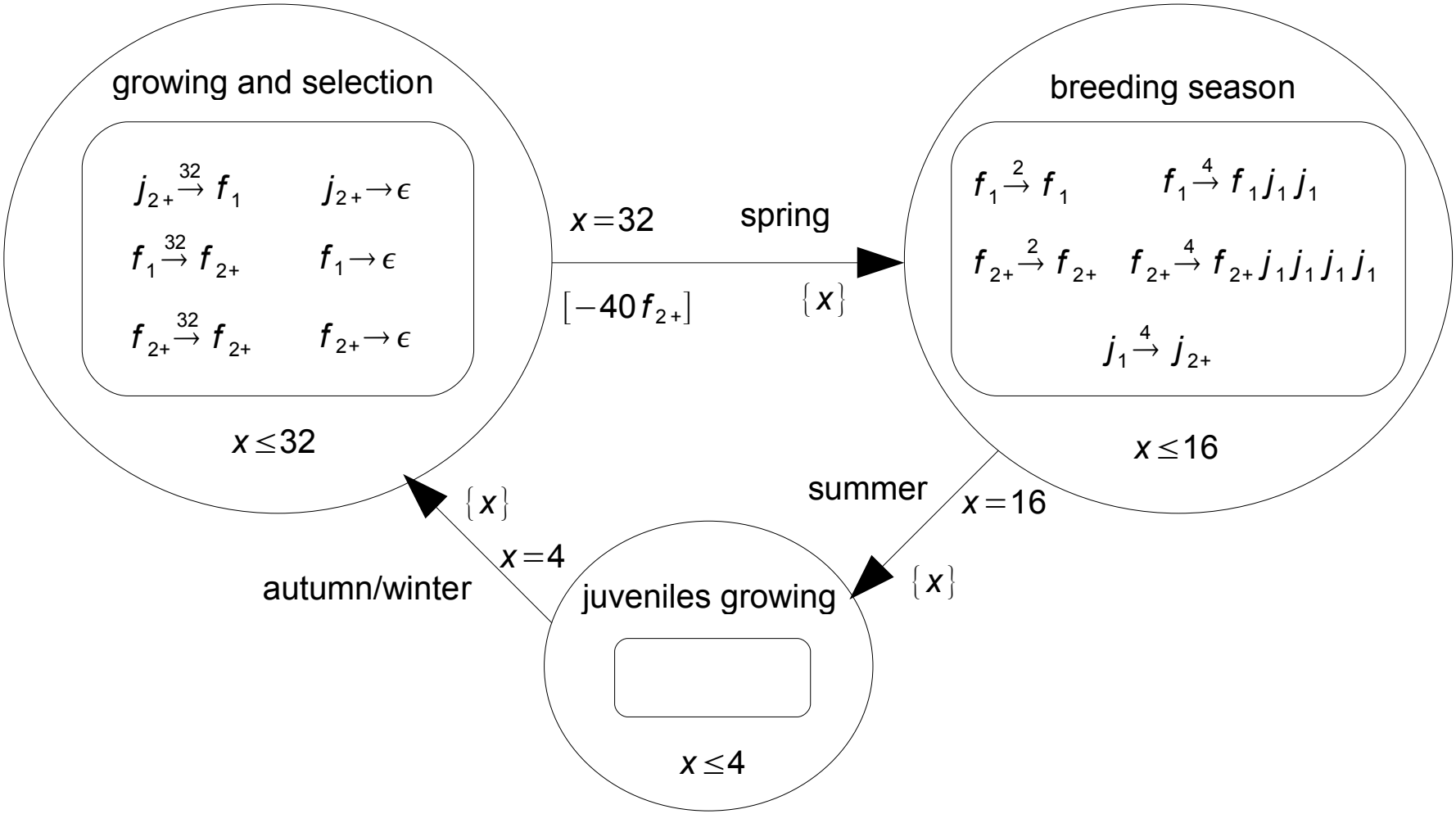
- The model is derived from the observation of the population of Saddleback birds (*Philesturnus rufusater*) on Mokoia Island

Description of the model

The model we have found:

- is a stochastic, discrete-time female-only model (the female-only approach assumes that there are sufficient males for all females to be paired)
- females are partitioned in two classes (first-year and older) with different fecundity rates (#fledgings/season)
- an annual harvest of females is scheduled, with harvesting taking place at the start of breeding season.

A timed P automaton model



References

Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S., Compositional Semantics and Behavioural Equivalences for P Systems, Theoretical Computer Science 395 (2008), pp. 77-100.

Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S., A P Systems Flat Form Preserving Step-by step Behaviour, Fundamenta Informaticae 87 (2008), pp. 1-34.

Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S., P Systems with Transport and Diffusion Membrane Channels, Fundamenta Informaticae 93 (2009), pp. 1-15.

Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tesei, L., Timed P Automata, Fundamenta Informaticae 94 (2009), pp. 1-19.

Barbuti, R., Gruska, D.P., Maggiolo-Schettini, A., Milazzo, P., Interpretation of some systems security notions in biological diagnostics, Concurrency Specification and Programming (CS&P'09), pp. 38-49.