Timed P Automata

Roberto Barbuti[1] Andrea Maggiolo Schettini[1]
Paolo Milazzo[1] Luca Tesei[2]

1. Dipartimento di Informatica, Università di Pisa, Italy
2. Dipartimento di Matematica e Informatica, Università of Camerino, Italy

Iași – September, 2008

Introduction

We are interested in modeling ecological systems

Recently, a model based on P systems of a population of Vultures of the Catalan Pyrenees has been studied

We would like to build models that can describe periodical changes in the environmental conditions

- seasons
- periodical hunt/harvest

We shall consider timed P systems and allow evolution rules to be changed when conditions on the elapsing of time are satisfied

Outline of the talk

1. Introduction
2. Background
   - P systems
   - timed P systems
   - timed automata
3. Timed P automata
   - example
   - formal definition
4. An application
   - modeling saddleback reintroduction

P systems

A *P system* consists of a *hierarchy of membranes*, each of them containing

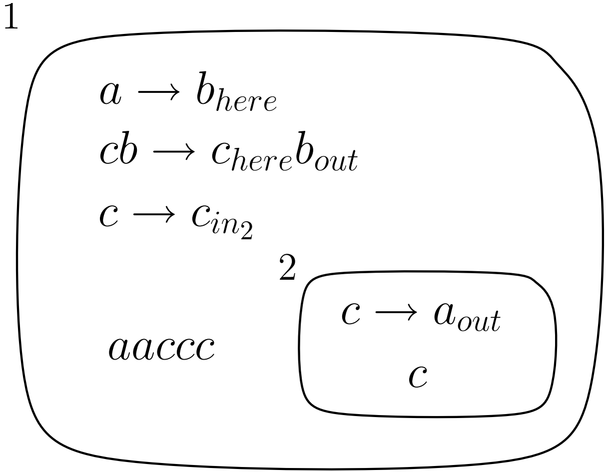
- a multiset of *objects*
- and a set of *evolution rules*

Evolution rules are applied with *maximal parallelism* to objects of the same membrane

The application of evolution rules of different membranes is synchronized by a global clock

Evolution rules can send objects into (immediately) outer and inner membranes

P systems


1
$a \longrightarrow b_{here}$
$cb \longrightarrow c_{here} b_{out}$
$c \longrightarrow c_{in_2}$
$aaccc$
2
$c \longrightarrow a_{out}$
c

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

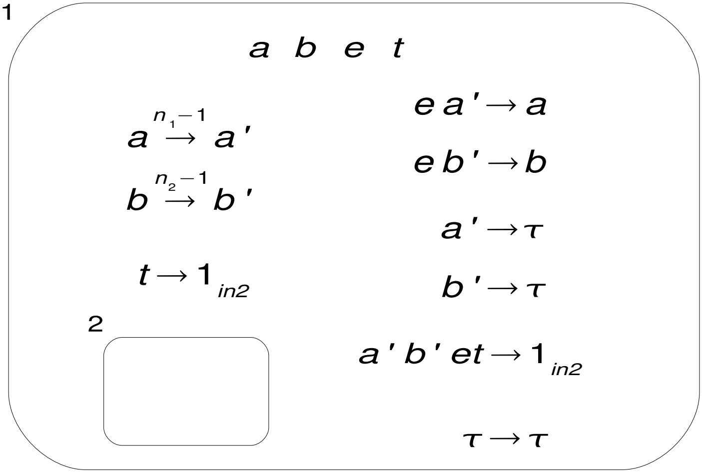
- consumed objects are immediately removed
- produced objects appear after n steps

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

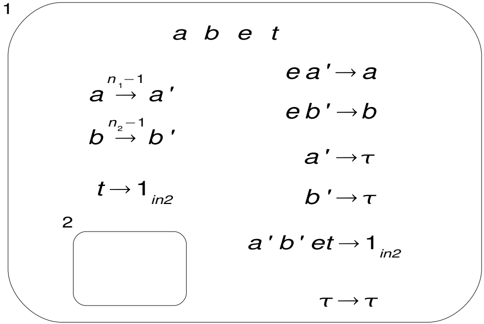
Example of timed P system

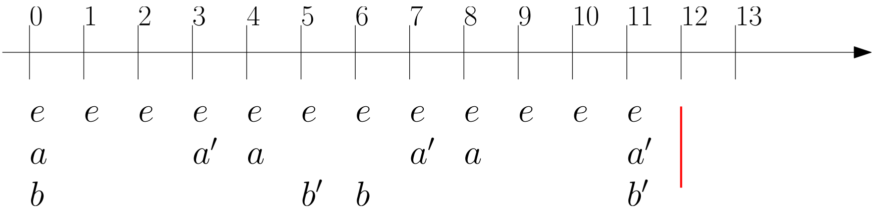
A timed P system computing the least common multiple of n_1 and n_2.



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.

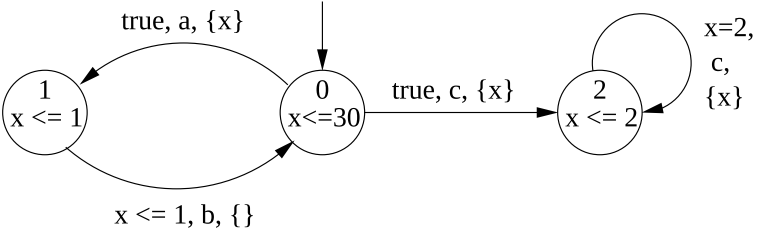
Outline of the talk

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



Outline of the talk

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

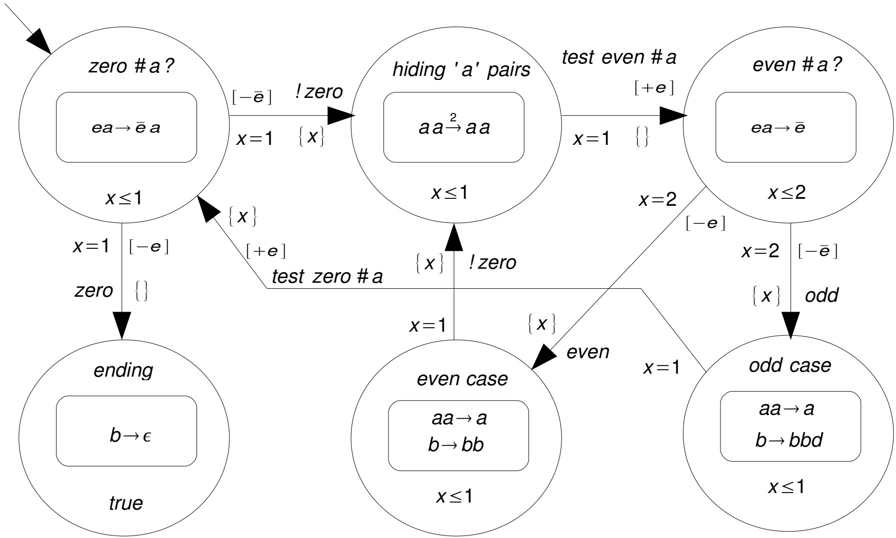
A computation starts in the timed P system associated with the inital location of the automaton

- 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

An example of timed P automaton



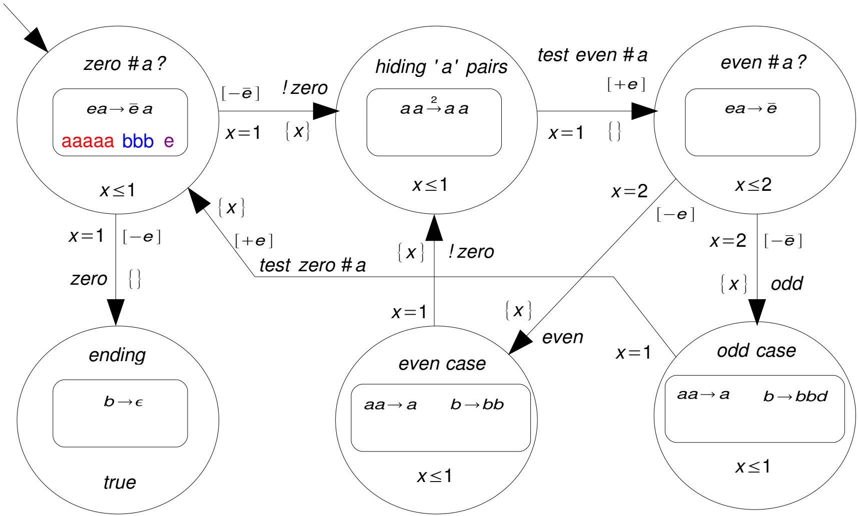
The Ancient Egyptian multiplication algorithm (2000 B.C.)

```
d ← 0
while (a ≠ 0)
   if (a is even)
      a ← a / 2
      b ← b * 2
   else
      d ← d + b
      a ← ⌊a / 2⌋
      b ← b * 2
   endif
endwhile
```

a	b
35 +	**42**
17 +	**84**
8	168
4	336
2	672
1 +	**1344**
	-------
d	**1470**

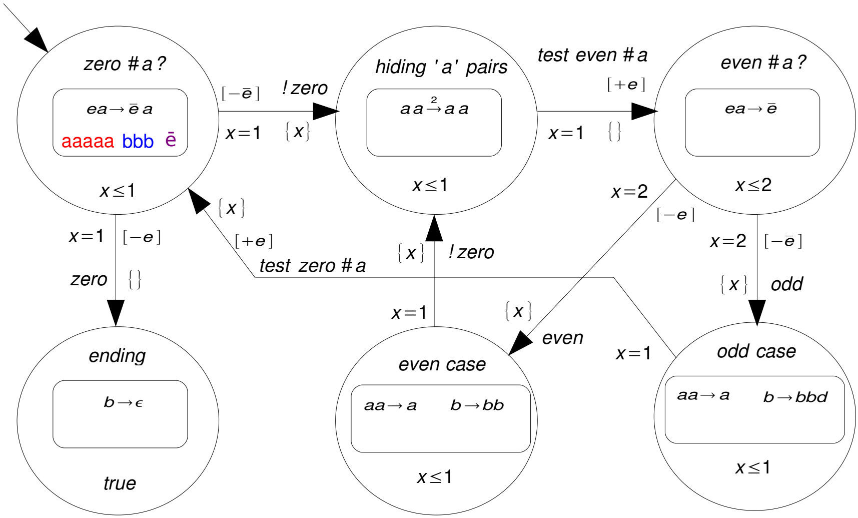
Example: 5 x 3 (step 0)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



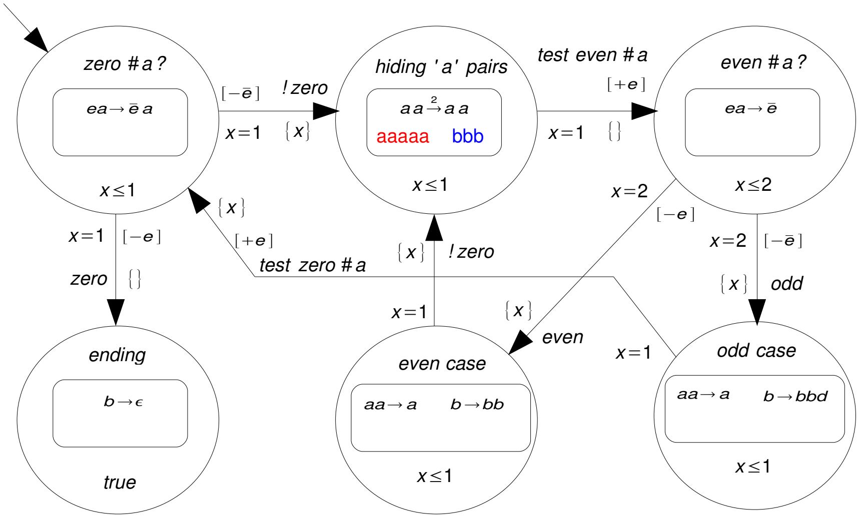
Example: 5 x 3 (step 1)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



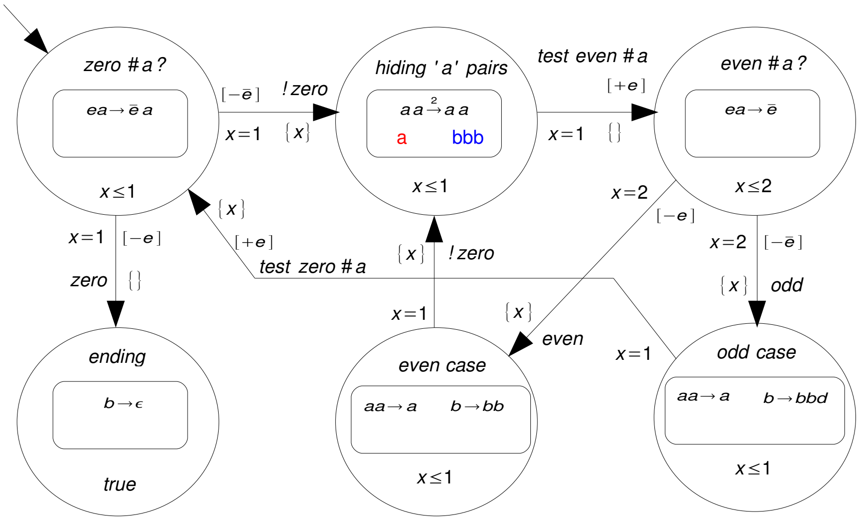
Example: 5 x 3 (step 2)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



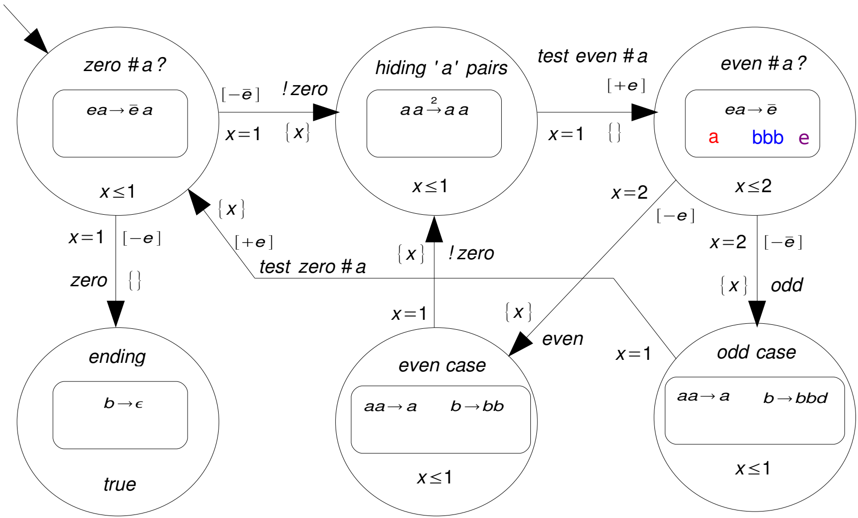
Example: 5 x 3 (step 3)

Clock $x = 1$, pending rules $\mathcal{U} = \{\xrightarrow{1} aa\,,\ \xrightarrow{1} aa\}$



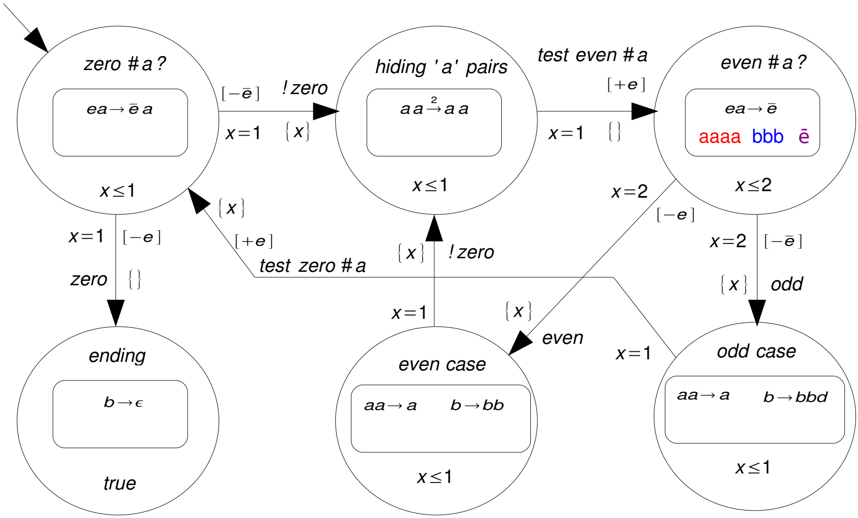
Example: 5 x 3 (step 4)

Clock $x = 1$, pending rules $\mathcal{U} = \{\xrightarrow{1} aa\,,\ \xrightarrow{1} aa\}$



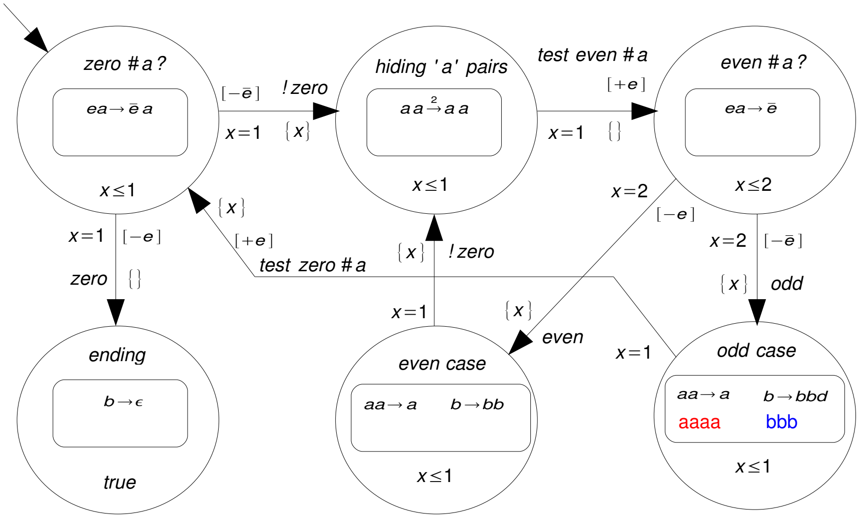
Example: 5 x 3 (step 5)

Clock $x = 2$, pending rules $\mathcal{U} = \varnothing$



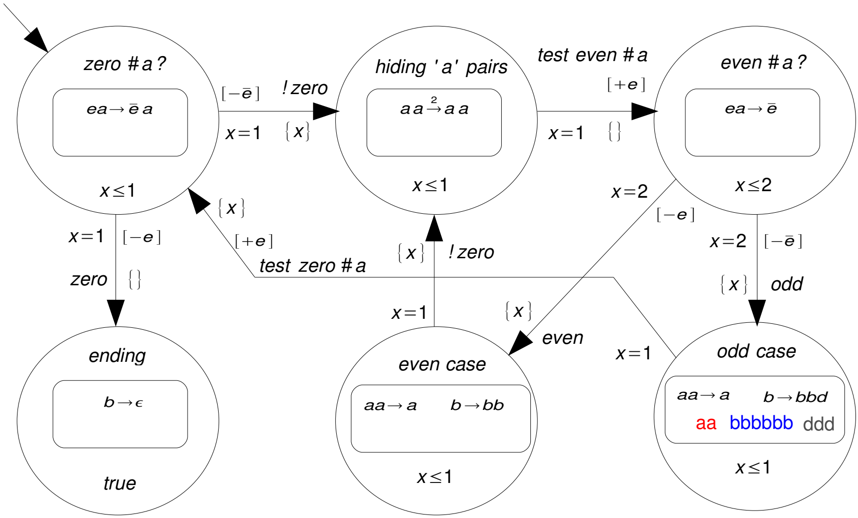
Example: 5 x 3 (step 6)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



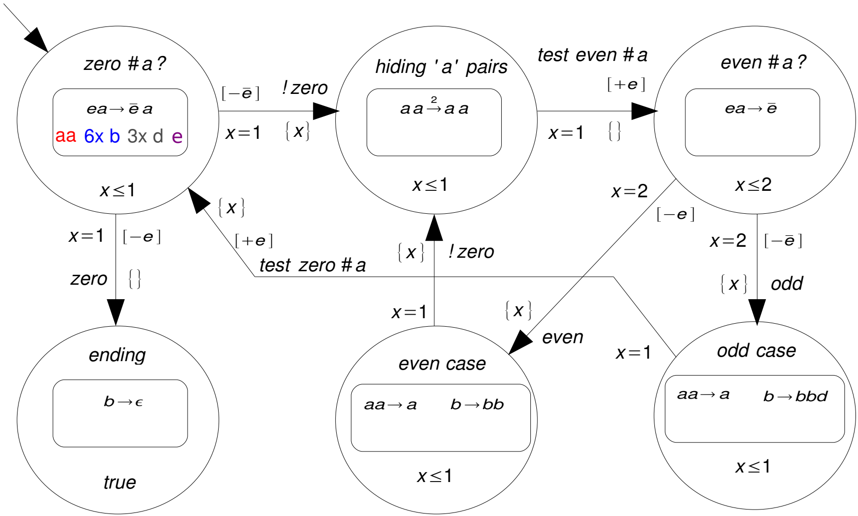
Example: 5 x 3 (step 7)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



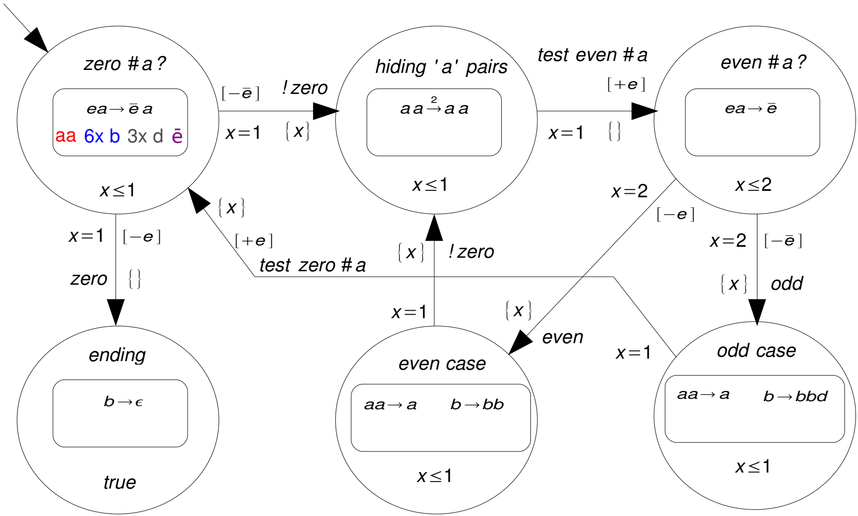
Example: 5 x 3 (step 8)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



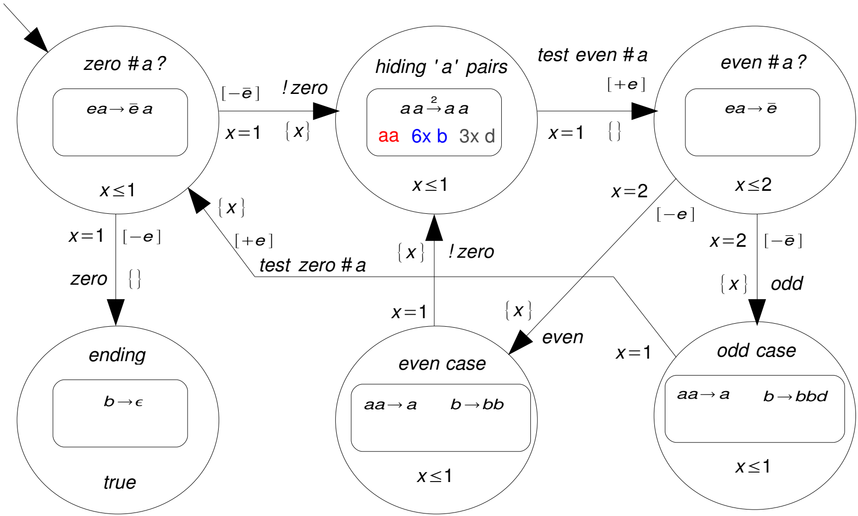
Example: 5 x 3 (step 9)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



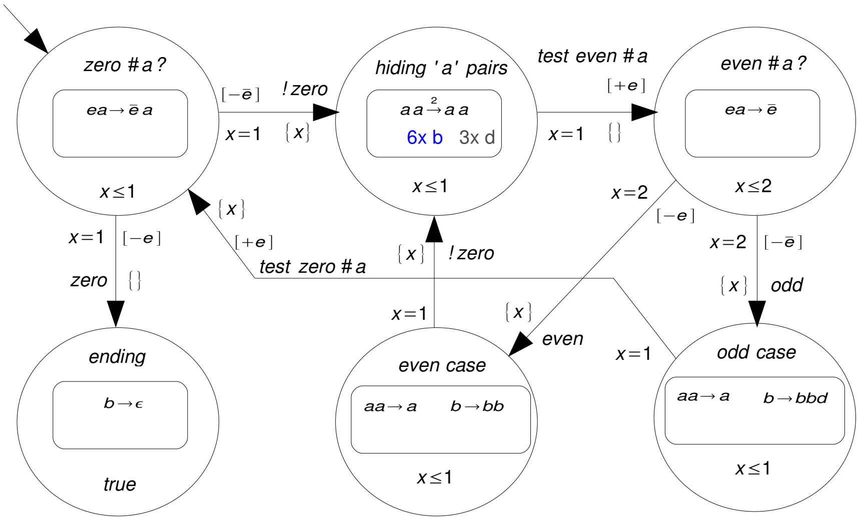
Example: 5 x 3 (step 10)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



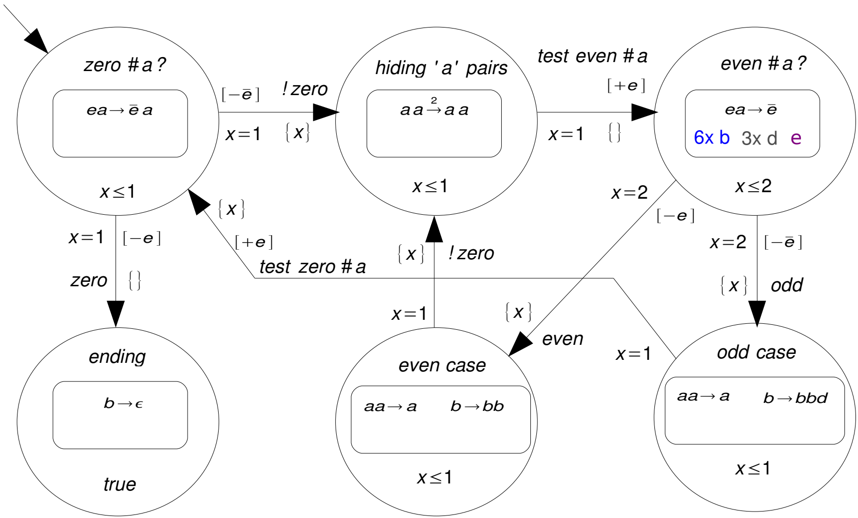
Example: 5 x 3 (step 11)

Clock $x = 1$, pending rules $\mathcal{U} = \{\xrightarrow{1} aa\}$



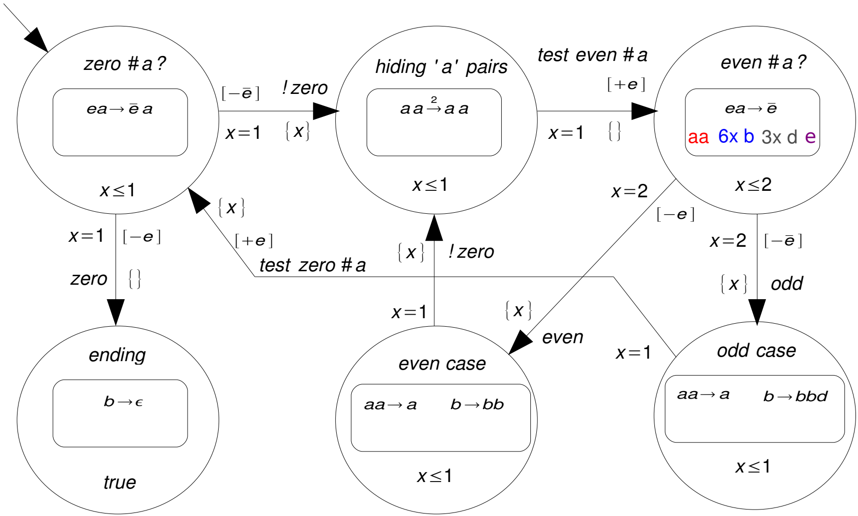
Example: 5 x 3 (step 12)

Clock $x = 1$, pending rules $\mathcal{U} = \{\xrightarrow{1} aa\}$



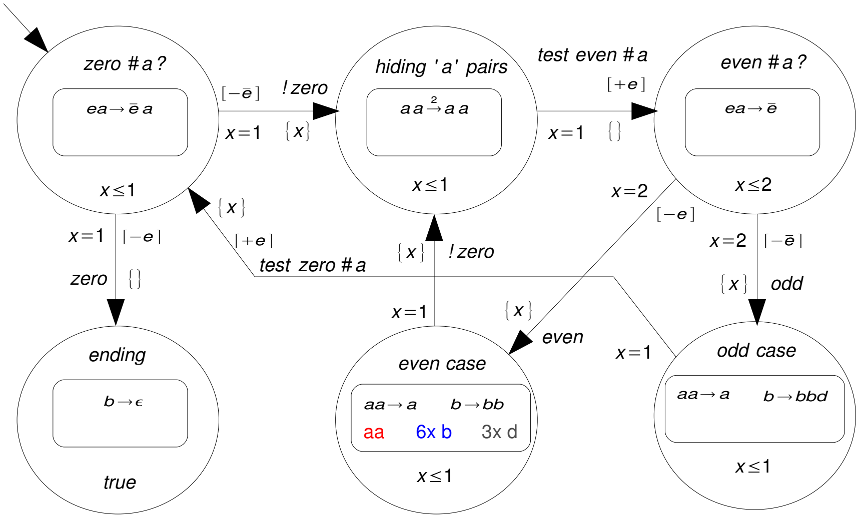
Example: 5 x 3 (step 13)

Clock $x = 2$, pending rules $\mathcal{U} = \varnothing$



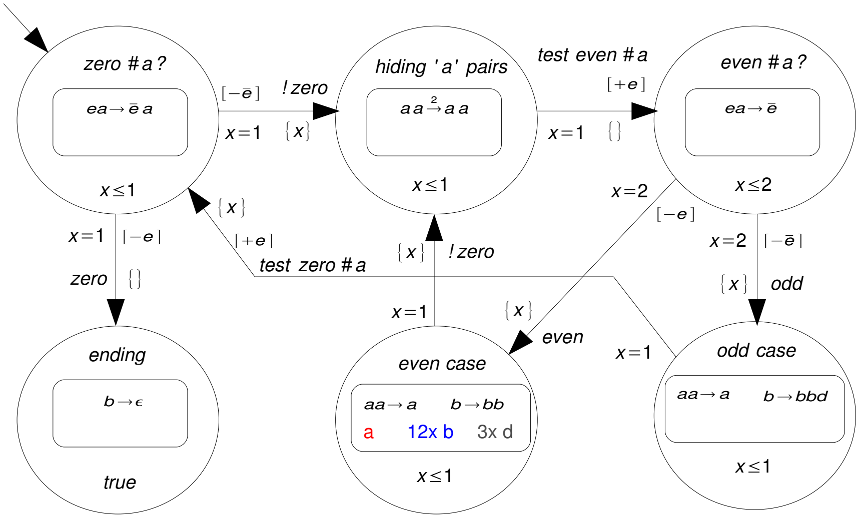
Example: 5 x 3 (step 14)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



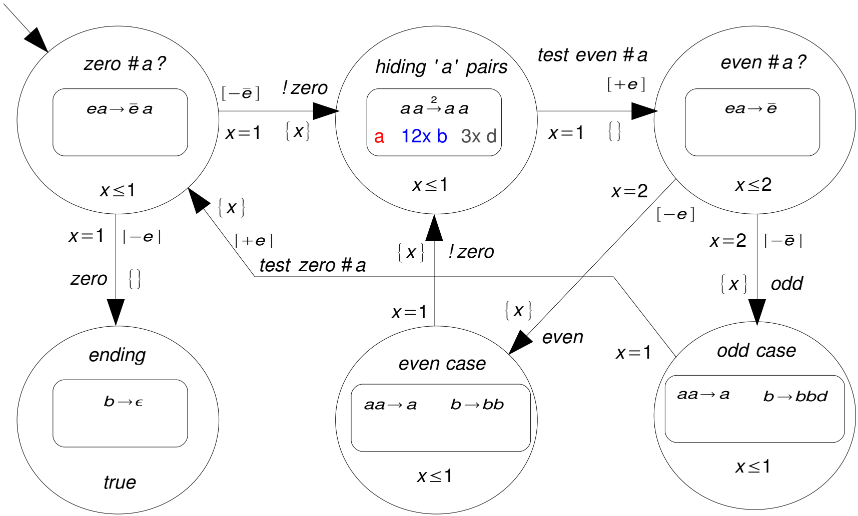
Example: 5 x 3 (step 15)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



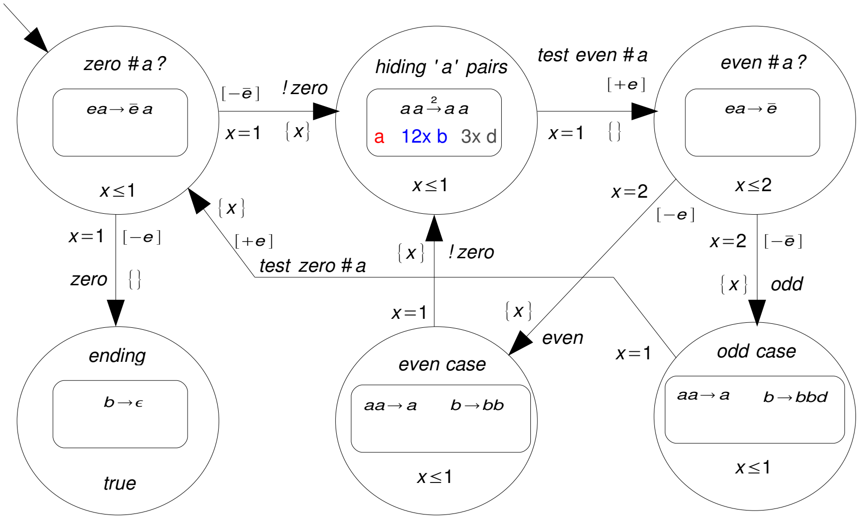
Example: 5 x 3 (step 16)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



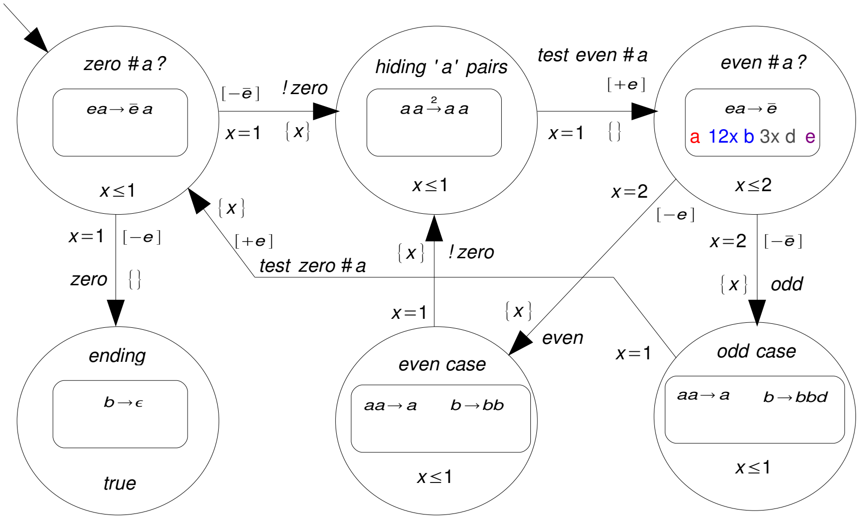
Example: 5 x 3 (step 17)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



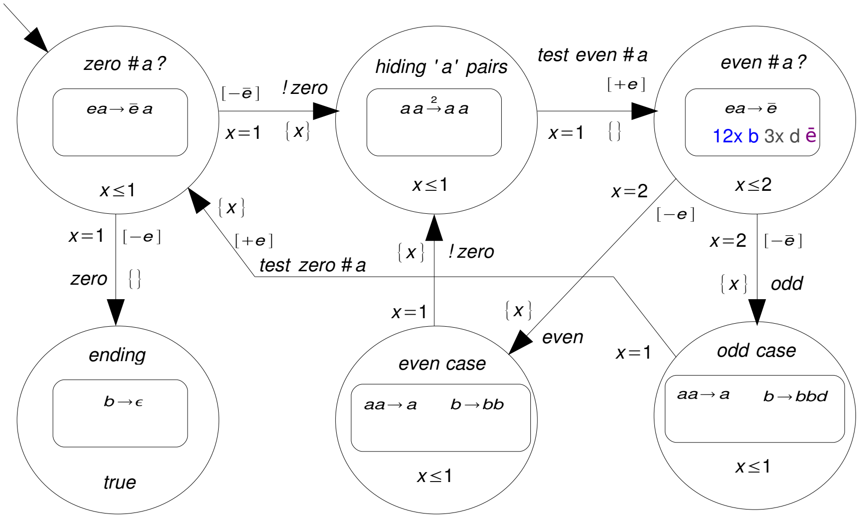
Example: 5 x 3 (step 18)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



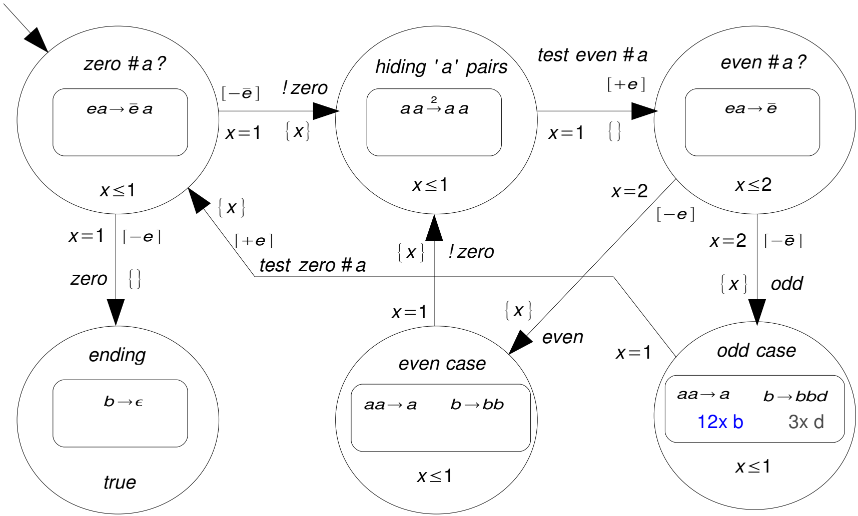
Example: 5 x 3 (step 19)

Clock $x = 2$, pending rules $\mathcal{U} = \varnothing$



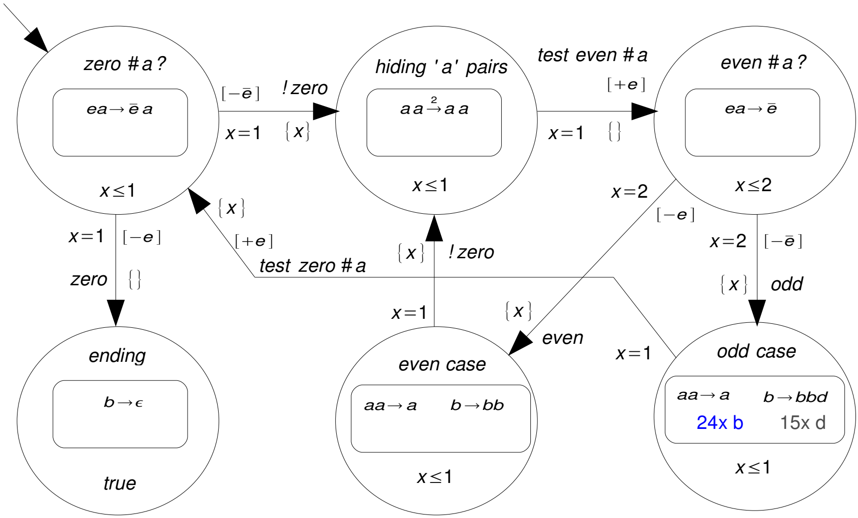
Example: 5 x 3 (step 20)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



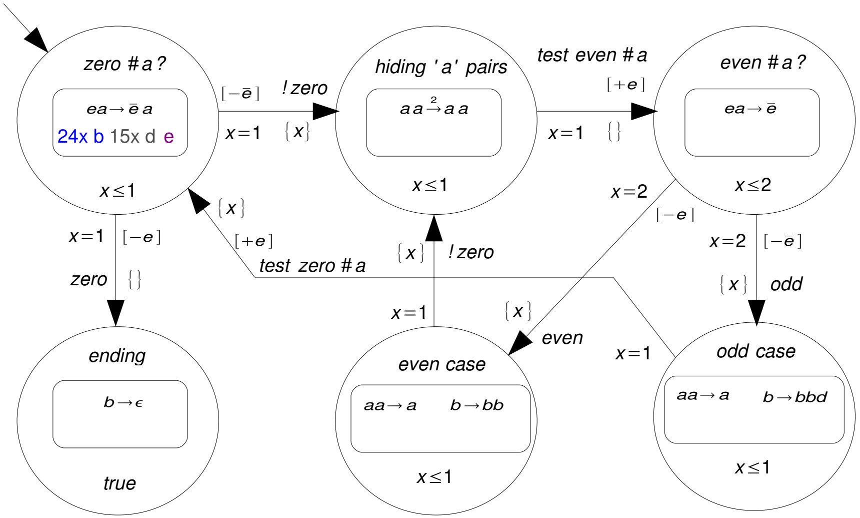
Example: 5 x 3 (step 21)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



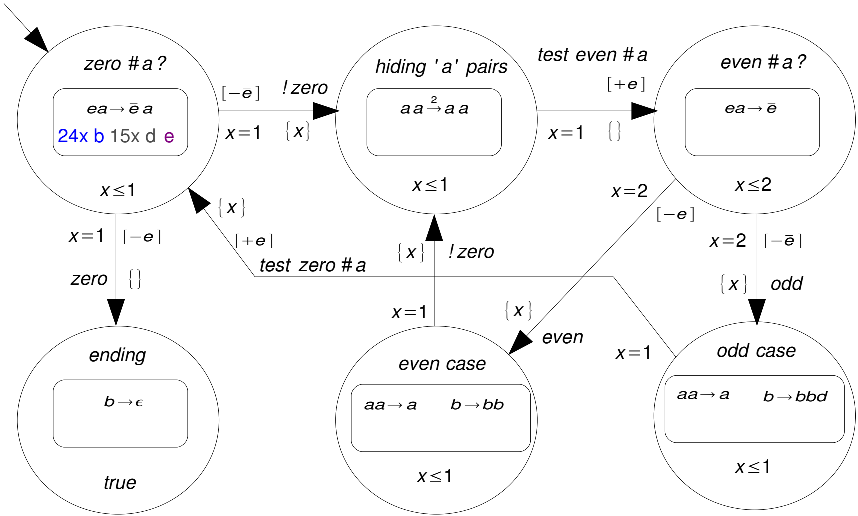
Example: 5 x 3 (step 22)

Clock $x = 0$, pending rules $\mathcal{U} = \varnothing$



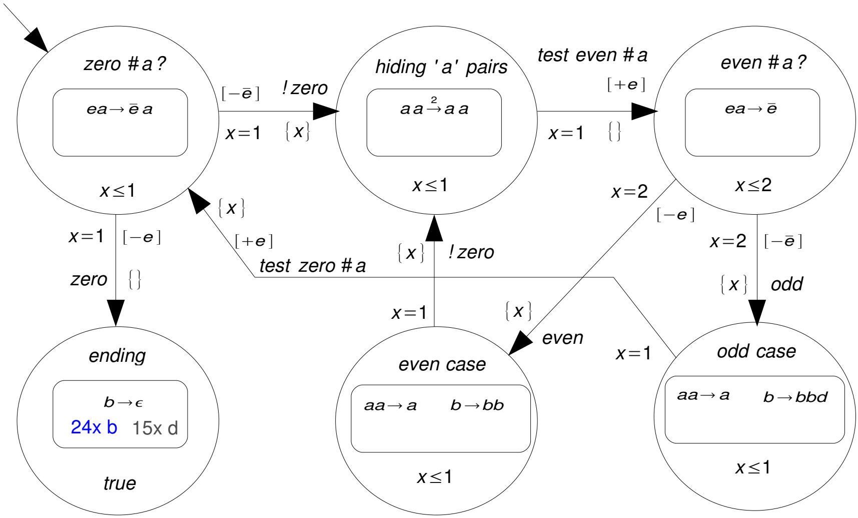
Example: 5 x 3 (step 23)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



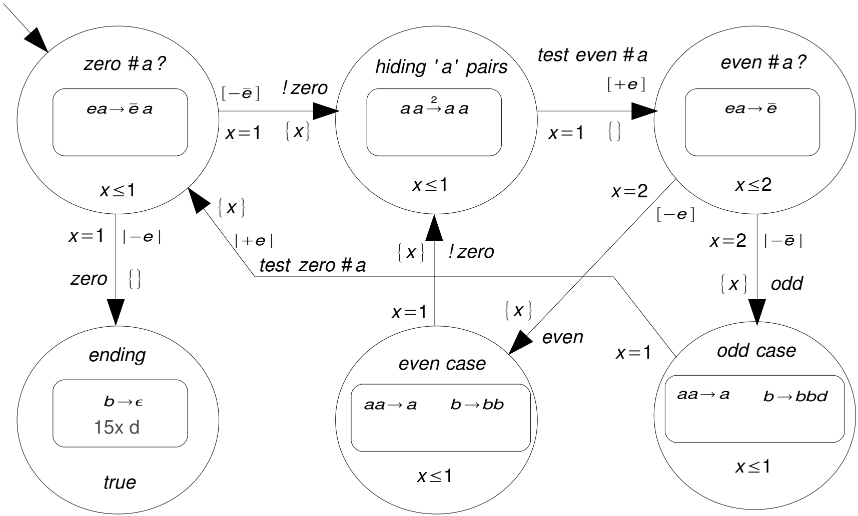
Example: 5 x 3 (step 24)

Clock $x = 1$, pending rules $\mathcal{U} = \varnothing$



Example: 5 x 3 (step 25 - THE END)

Clock $x = 2$, pending rules $\mathcal{U} = \varnothing$



Outline of the talk

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:

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

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

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 one membrane, one bi-stable catalist and an alphabet with only one symbol (apart from the catalist) is enough to get universality
- It would be interesting to check whether universality holds with non-cooperative rules

Outline of the talk

1. Introduction
2. Background
   - P systems
   - timed P systems
   - timed automata
3. Timed P automata
   - example
   - formal definition
4. An application
   - modeling saddleback reintroduction

The saddleback

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

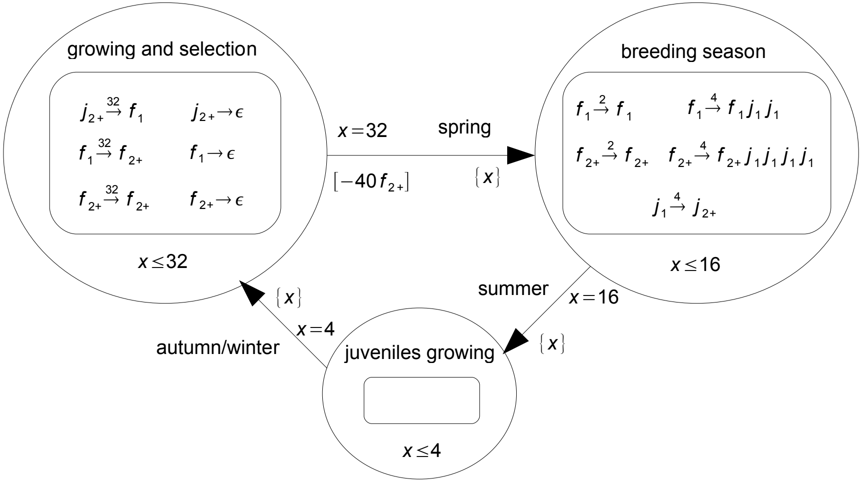
- The model is derived from the observation of the population of Saddleback birds (*Philesturnus rufusater*) on Mokoia Island

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



Conclusion

We have defined an extension of timed P system in which evolution rules may vary with time

- timed P systems + timed automata = timed P automata

Our aim was to define a formalism to model ecological systems

Future work includes:

- the development of a (stochastic) simulator
- the application to some case study
- further investigation of the computational capabilities