

# When are Timed Automata weakly timed bisimilar to Time Petri Nets ?

B. Bérard\*, F. Cassez+, S. Haddad\*, D. Lime+, O.H. Roux+

\*LAMSADE CNRS UMR 7024, Paris, France

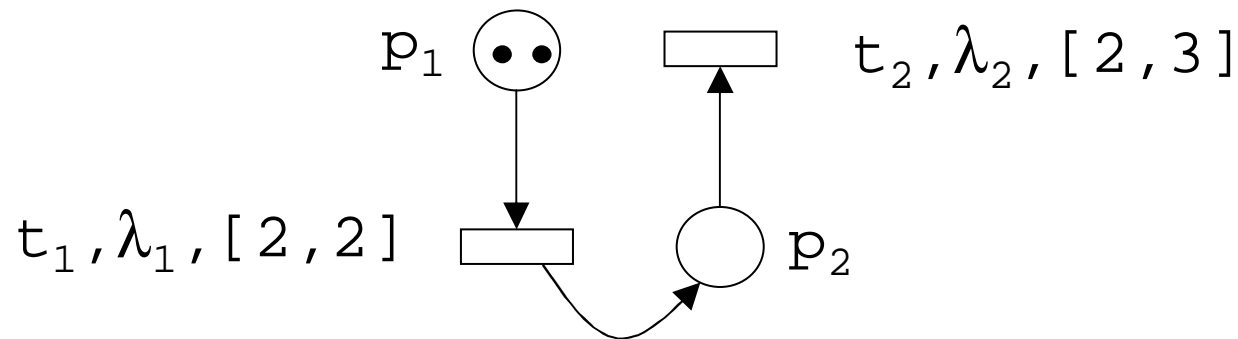
+IRCCyN CNRS UMR 6597, Nantes, France

- Time Petri nets and timed automata
- The characterisation
- Necessity proof
- Sufficiency proof

# Why time Petri nets ?

- Time Petri nets (TPNs) is a concise model for managing simultaneously concurrency and time.
- Bounded TPNs are the support of efficient algorithms deciding reachability and CTL\* formulae with exploitation of partial order techniques.
- The softwares TINA and ROMEO include numerous tools.
- TINA is integrated in a general framework for real-time architectures based on AADL.

# Time Petri nets: syntax



- Places: logical part of the state
- Tokens: current value of the logical part of the state
- Transitions: events, actions, ...
- Labels: observable behaviour
- Arcs: Pre and Post (logical) conditions of events occurrence
- Time (closed) intervals: temporal conditions of events occurrence

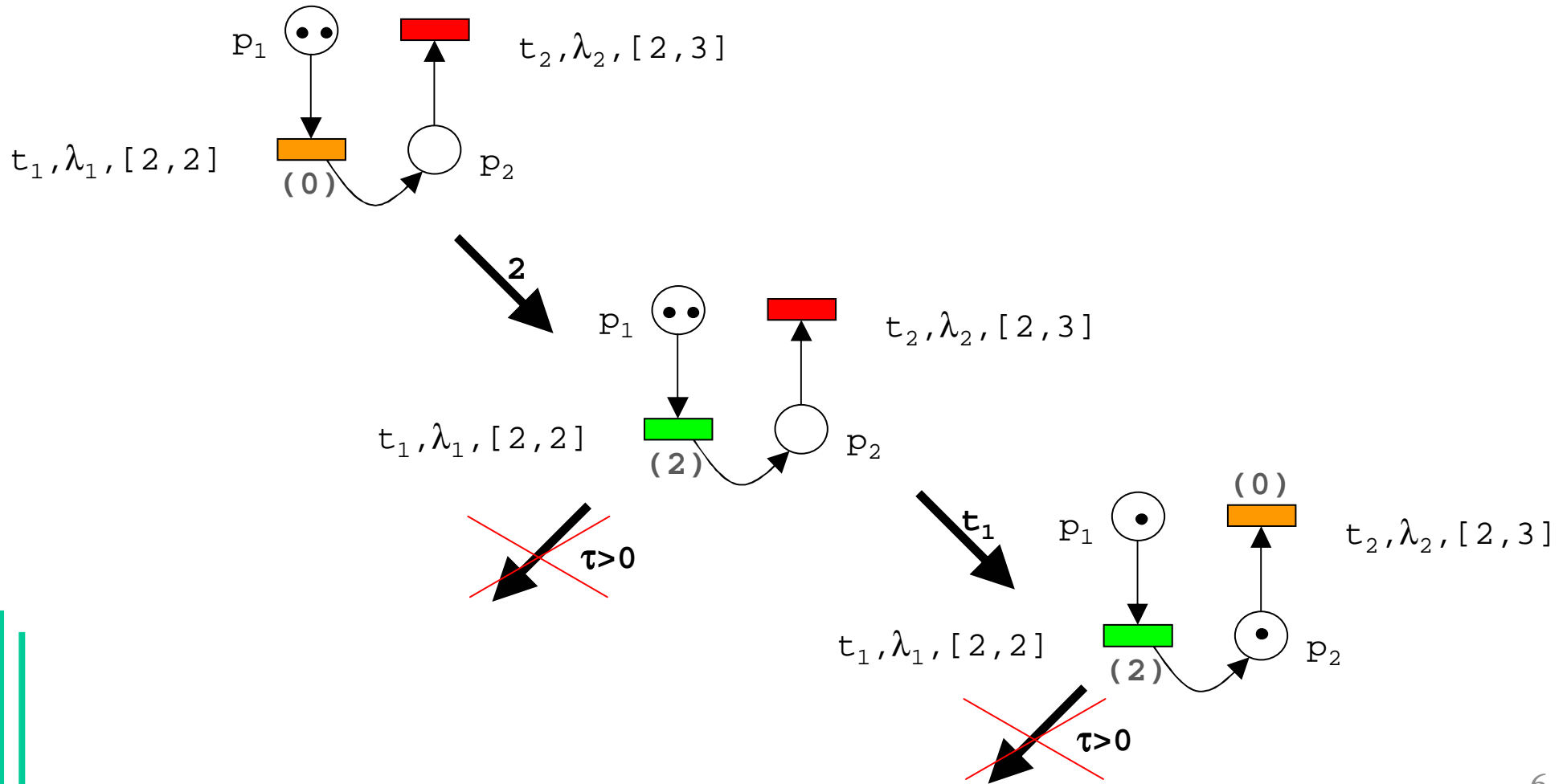
# Time Petri nets: transitions occurrence

- Logical part
  - The logical part of a state is a *marking*, i.e. a number of tokens per place.
  - A transition is *enabled* if the tokens required by the pre conditions are present in the marking.
- Timed part
  - There is an implicit *clock* per enabled transition and its value defines the timed part of the state.
  - An enabled transition is *fireable* if its clock value lies in its interval.

# Time Petri nets: changes of state

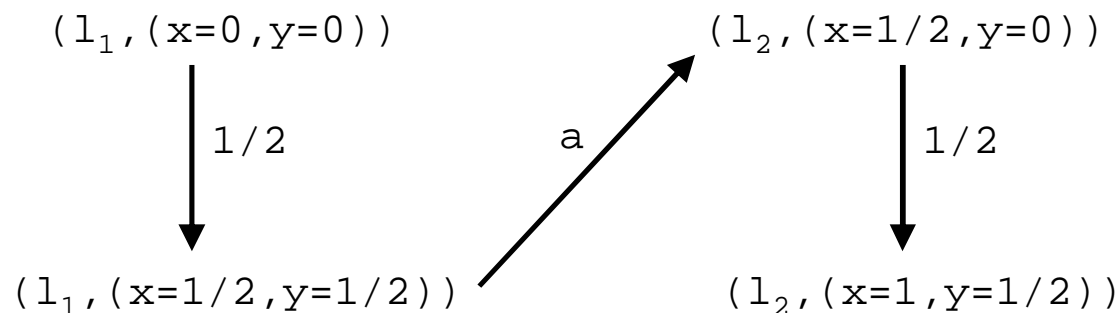
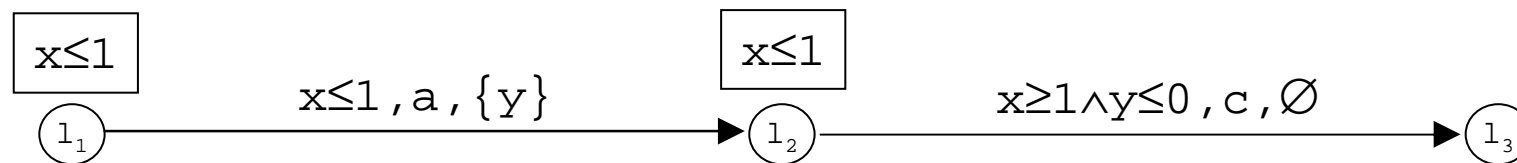
- Time elapsing
  - The marking is unchanged.
  - Time may elapse (with updates of clocks) if every clock value **does not go beyond** the corresponding interval.
- Transition firing
  - Tokens required by the pre condition are consumed and tokens specified by the post condition are produced.
  - Clocks values of *newly enabled* transitions are reset.

# Time Petri nets: an execution



# Timed automata: syntax and semantics

- Finite automata enlarged with clocks
- Elementary clock constraints:  $x \ \Xi \ h$  with  $\Xi \in \{<, >, \leq, \geq\}$
- Clock constraints: conjunction of elementary ones
- State invariants: constraints restricting time elapsing
- Transitions extended with a clock constraint and some clock resets



# Time Petri nets and timed automata (1)

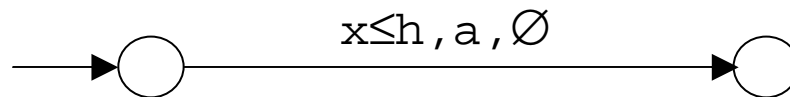
- Logical part (TPNs more expressive than TA)
  - In TPNs, implicit representation with possibly an infinite number of discrete states.
  - In TA, explicit representation ... but can be made more concise by parallel composition.
- Timed part (TA more expressive than TPNs)
  - In TPNs, transitions are ruled by a single clock.
  - In TPNs, clock resets are not arbitrary.
  - In TPNs, urgent behaviours but not lazy ones.

# Time Petri nets and timed automata (2)

- Language point of view
  - Any timed automaton may be translated in linear time into a safe (*i.e.* one bounded) time Petri net (with any kind of intervals).
  - Safe time Petri nets are exponentially more concise than timed automata (*Bouyer, Haddad, Reynier 2006*).
- Weak time bisimulation point of view
  - There exist timed automata not bisimilar to any time Petri net.
  - Characterisation of label-free TA bisimilar to a TPN ?
  - Effective translation from such a TA to a TPN ?
  - Complexities of the characterisation and the translation ?

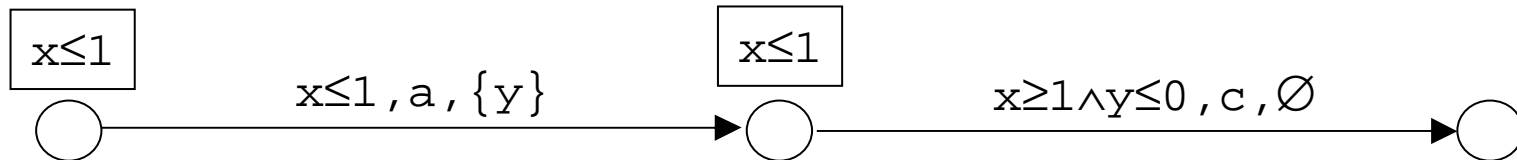
# Timing properties of TPNs

- Time elapsing does not disable transition firings (C1)
  - First idea for a characterisation of TA bisimilar to TPN
  - For instance, it can be proved that this TA is not bisimilar to any TPN

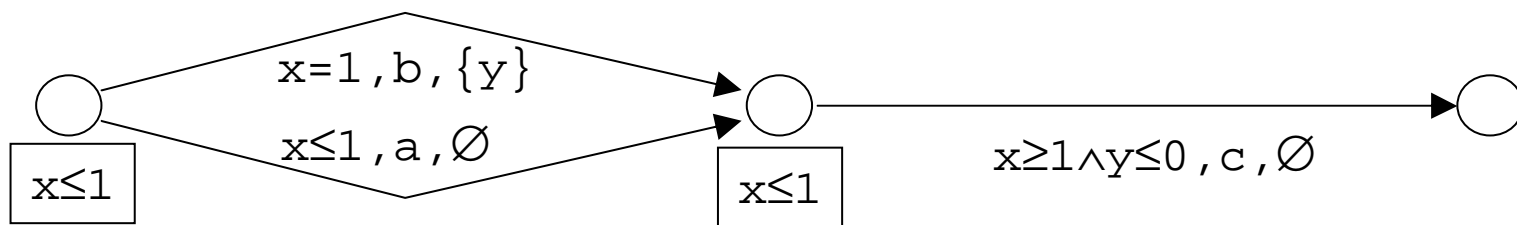
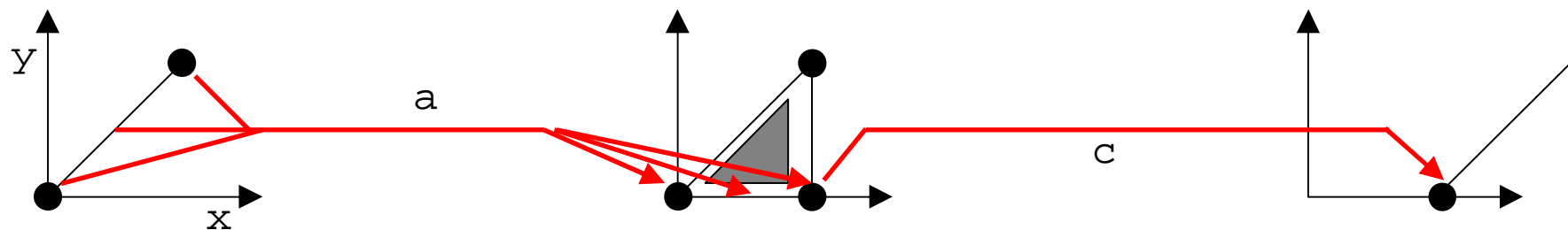


- In fact, (when possible) increasing arbitrarily the clock values of the enabled transitions does not disable transition firings (C2)
  - Second idea for a characterisation of TA bisimilar to TPN

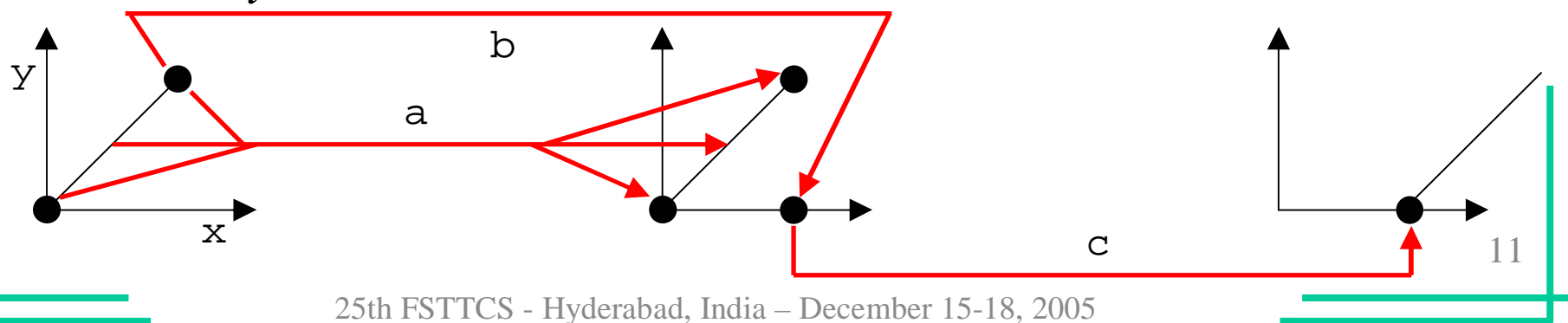
# Two timed automata



*C1 is not sufficient*

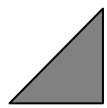


*C2 is not necessary*

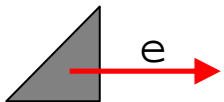
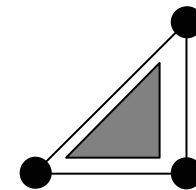


# Characterisation

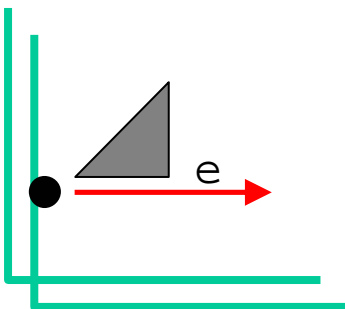
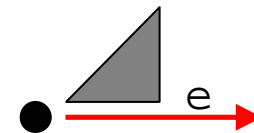
The characterisation is expressed with the help of the region automaton and the standard topology.



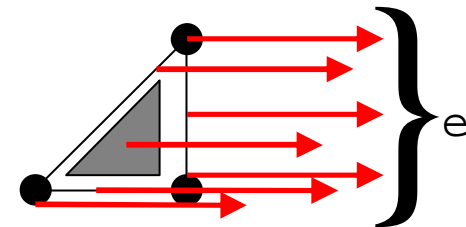
$$r \text{ reachable} \Rightarrow \forall r' \text{ s.t. } r' \cap \text{Closure}(r) \neq \emptyset, r' \text{ reachable}$$



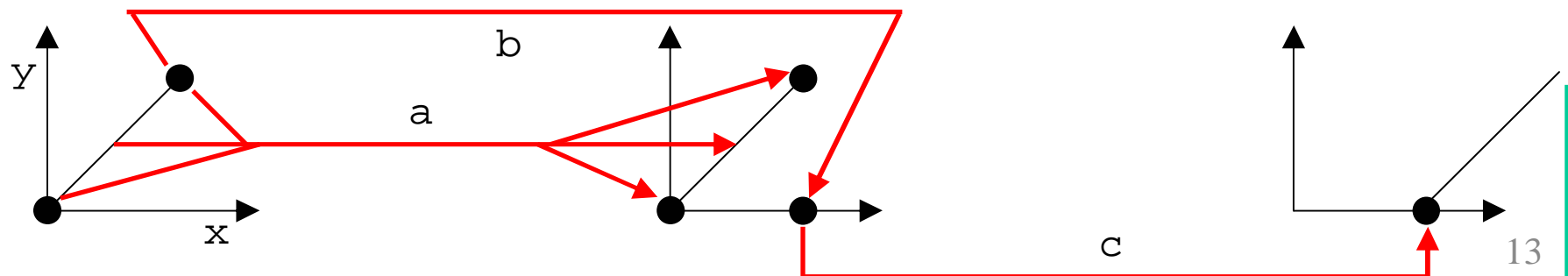
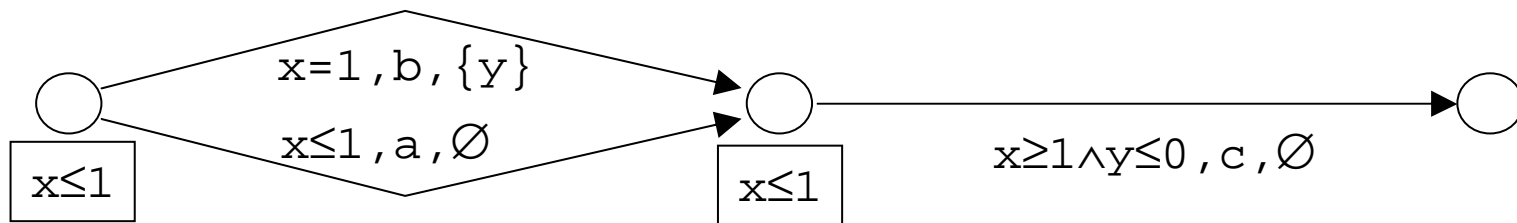
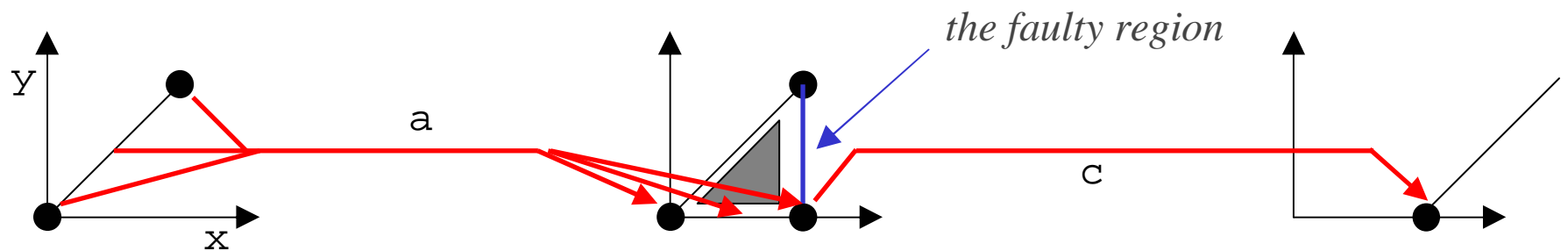
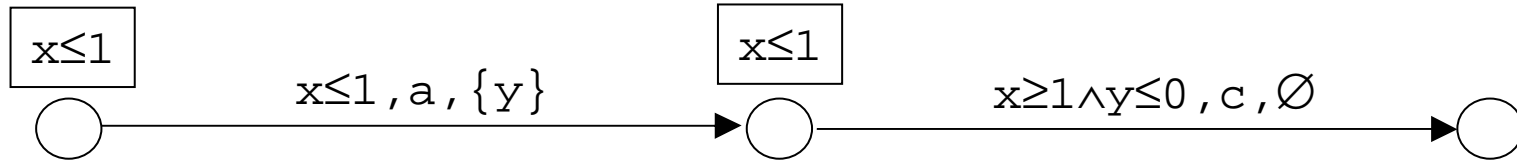
$$r \text{ reachable} \wedge r \rightarrow_e \Rightarrow \min_r \rightarrow_e$$



$$r \text{ reachable} \wedge \min_r \rightarrow_e \Rightarrow \forall r' \text{ s.t. } r' \cap \text{Closure}(r) \neq \emptyset, r' \rightarrow_e$$

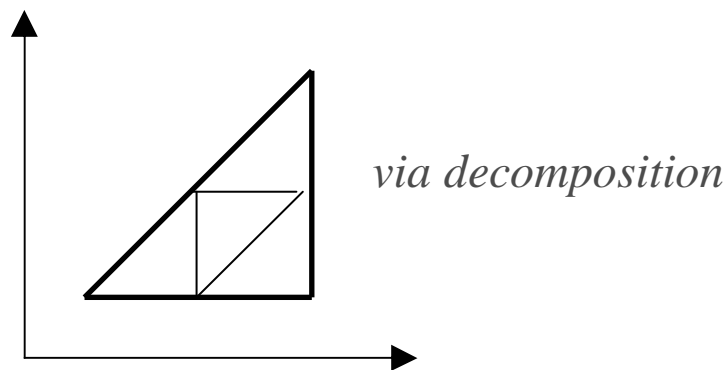


# Illustration

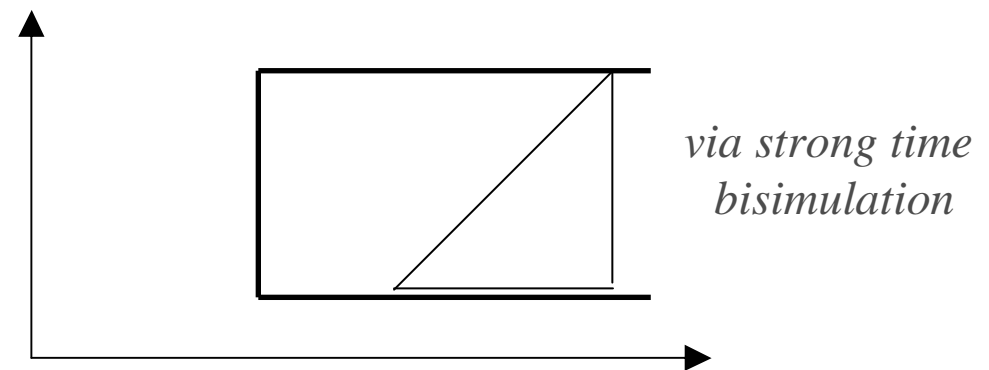


# Scheme of the necessity proof

- Let  $A$  be a TA bisimulated by a TPN  $N$  with *granularity*  $g$ ,
  - We define *uniform bisimulation* of  $A$  by  $N$  w.r.t. the  $(g, \infty)$ -region automaton (which is possibly infinite).
  - We establish the uniform bisimulation by induction on the reachability relation.
  - We deduce the condition for the  $(g, \infty)$ -region automaton.
- From  $(g, \infty)$ -regions to  $(1, K)$ -regions



From  $(g, \infty)$ -regions to  $(1, \infty)$ -regions

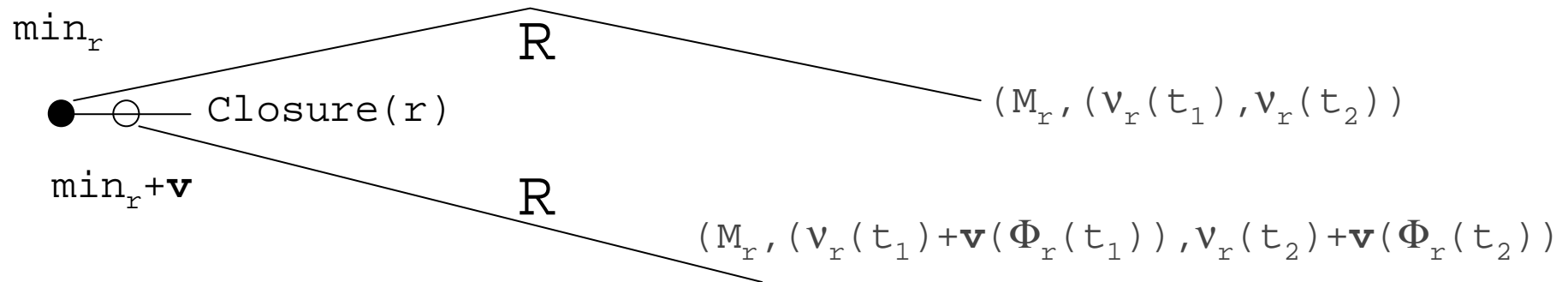


From  $(1, \infty)$ -regions to  $(1, K)$ -regions

# From bisimulation to uniform bisimulation (1)

- Let  $R$  be a bisimulation between configurations of  $A$  and  $N$ , then given a reachable time-closed region  $r$ ,  $\text{Closure}(r)$  is reachable and there exists:
  - $(M_r, v_r)$  a configuration of  $N$
  - and a mapping  $\phi_r$  from the enabled transitions of  $M_r$  to the equivalence clock classes

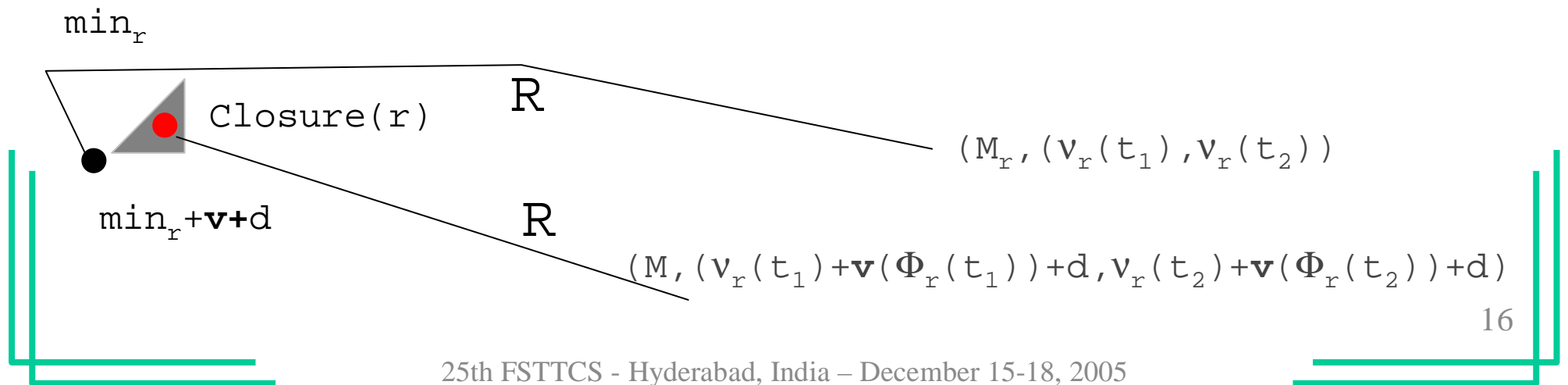
such that:



# From bisimulation to uniform bisimulation (2)

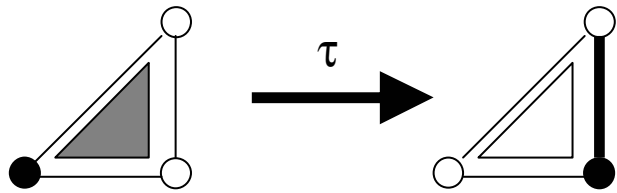
- Let  $R$  be a bisimulation between configurations of  $A$  and a  $N$ , then given a reachable time-open region  $r$ ,  $\text{Closure}(r)$  is reachable and there exists:
  - $(M_r, v_r)$  a configuration of  $N$
  - and a mapping  $\phi_r$  from the enabled transitions of  $M_r$  to the equivalence clock classes

such that:

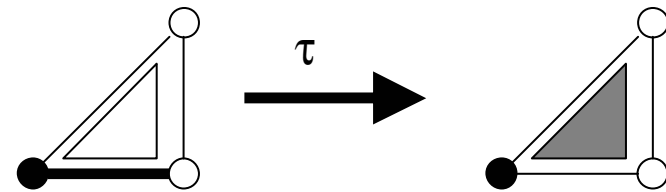


# Establishing uniform bisimulation

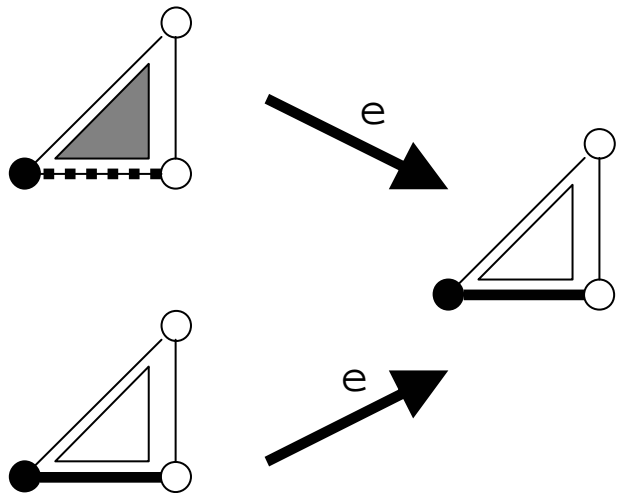
Time step into a time-closed region  
by coordinate change and restriction



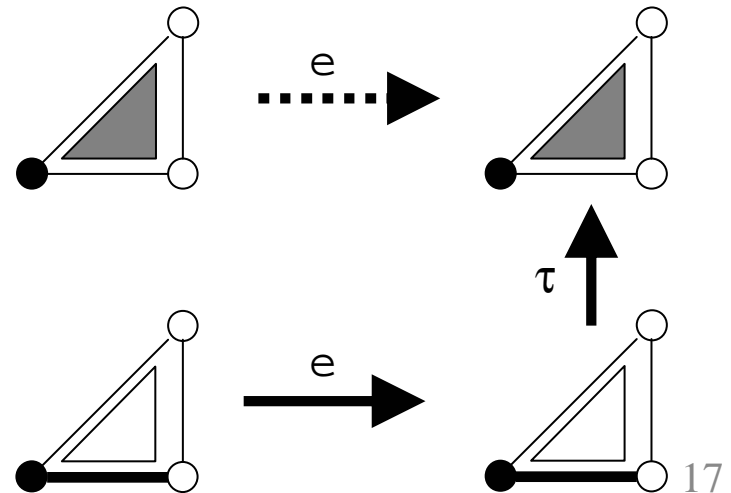
Time step into a time-open region  
using the net simulation  
of time elapsing from  $\min_r$



Discrete step into a time-closed region  
analysing the net simulation of the step  
and the effect of the clock resets

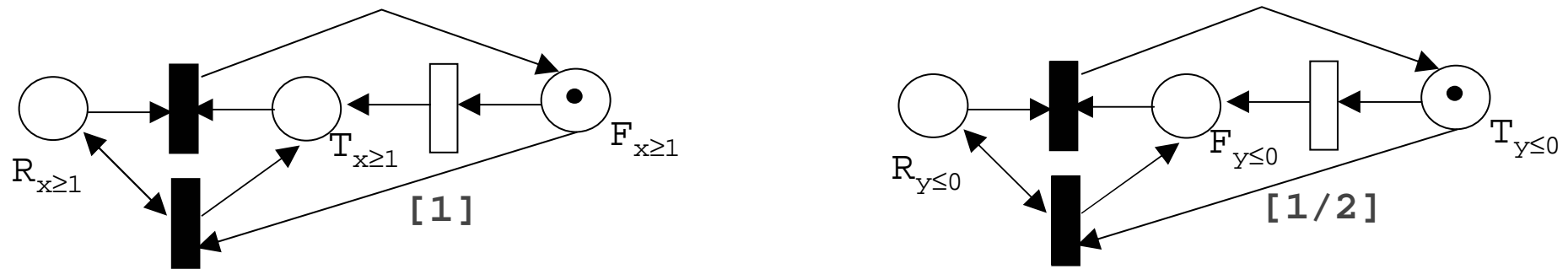


Discrete step into an time-open region  
not necessary to examine

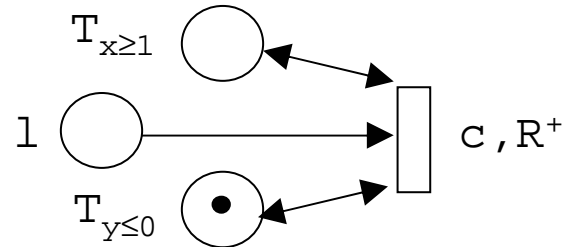


# Sufficiency condition: a structural translation

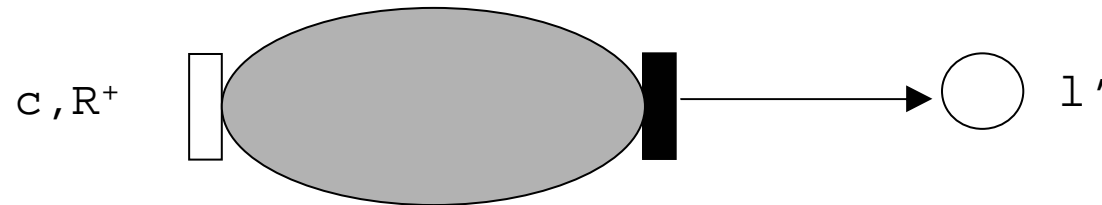
From timed conditions to logical conditions:  $x \geq 1, y \leq 0$



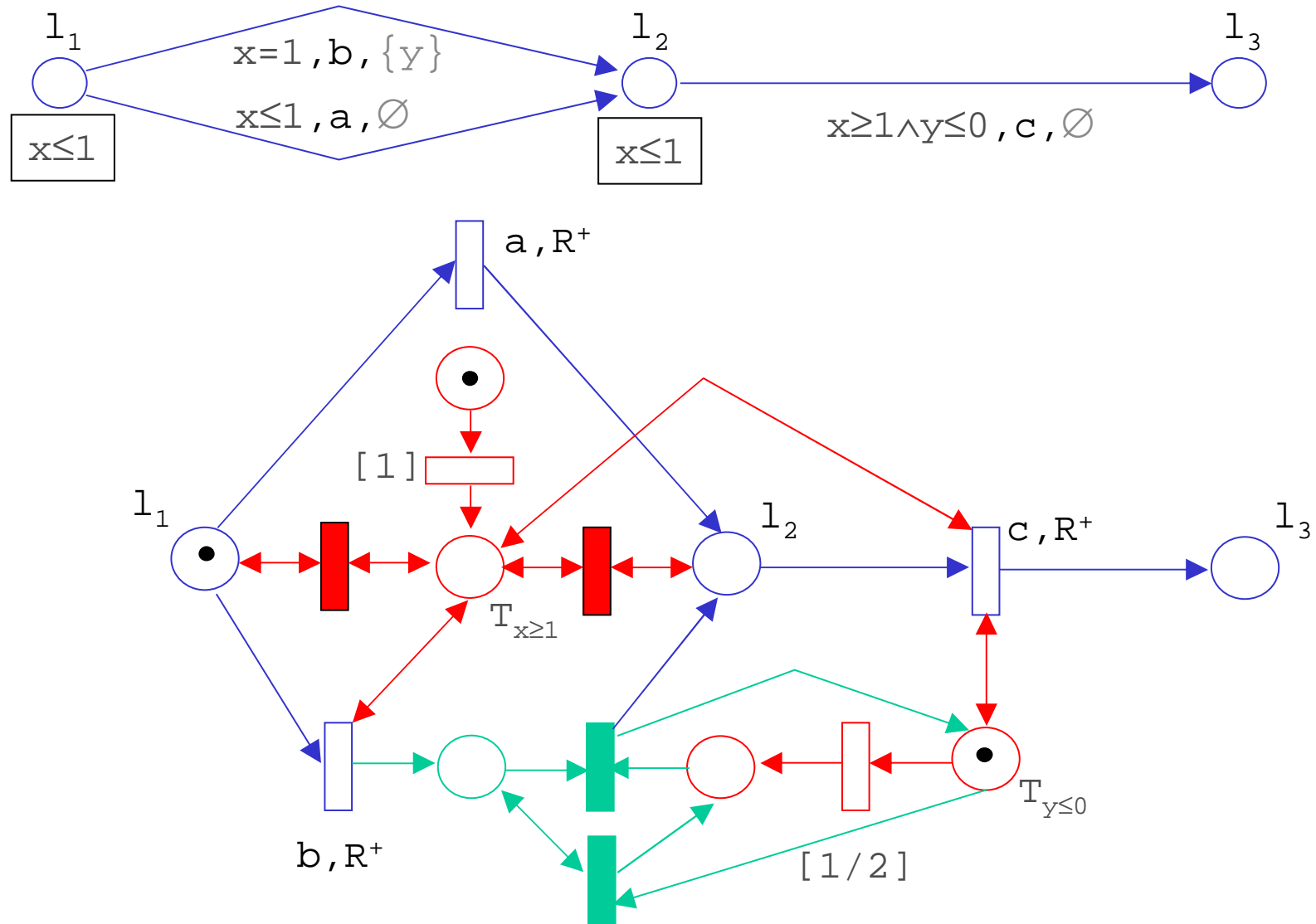
Untimed transitions for edge simulation  $(1, x \geq 1 \wedge y \leq 0, c, \{ \dots \}, 1')$



Clock resets obtained by subnet ``calls''

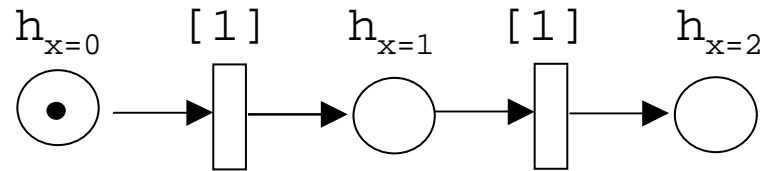


# Structural translation: illustration



# Sufficiency condition: a behavioural translation

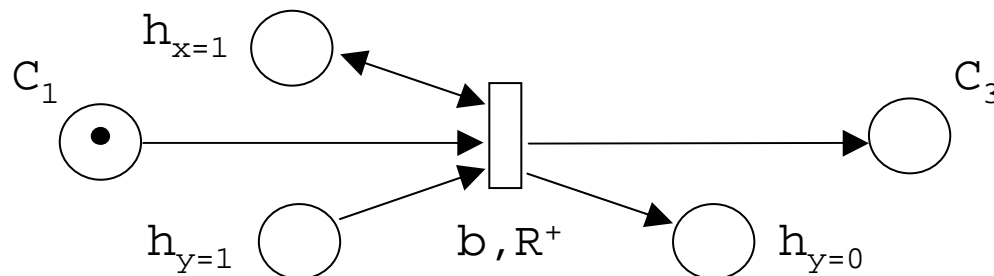
Tracking the integer values of clocks until  $K$  the maximal constant  $+1$ .



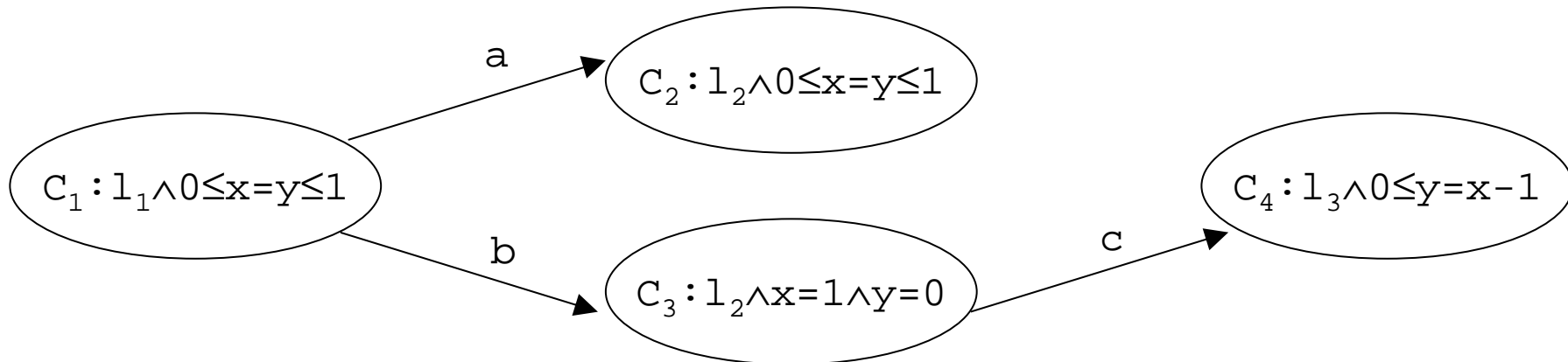
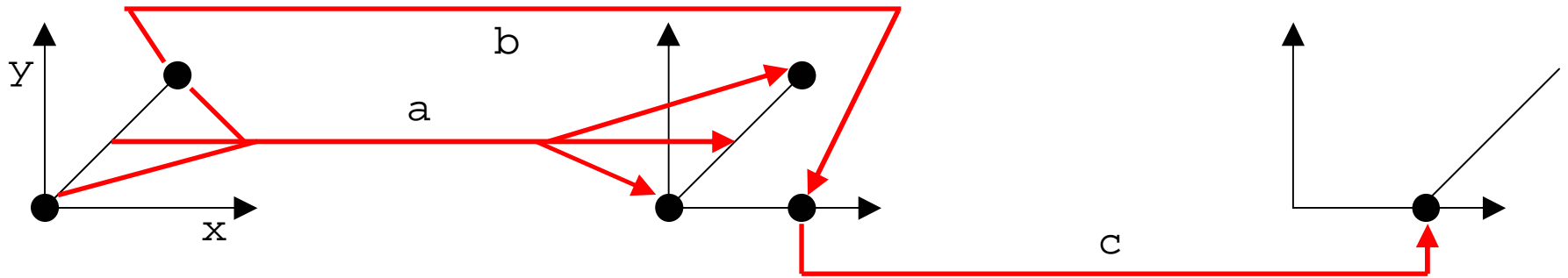
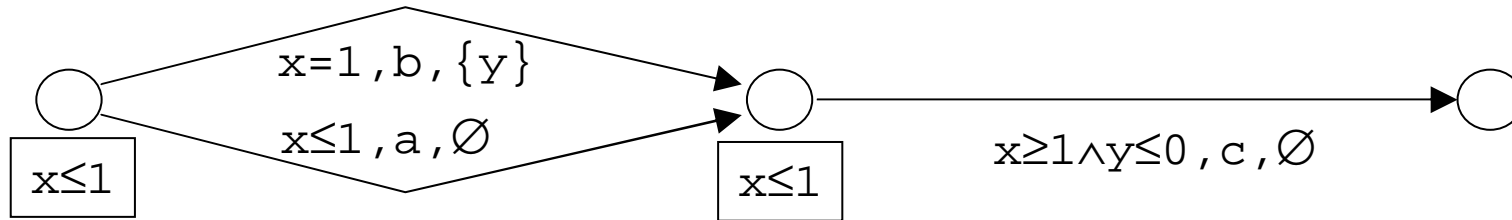
Construction of the zone graph (deterministic and time-invariant)

Edge simulation for pairs (integer points, zone) where

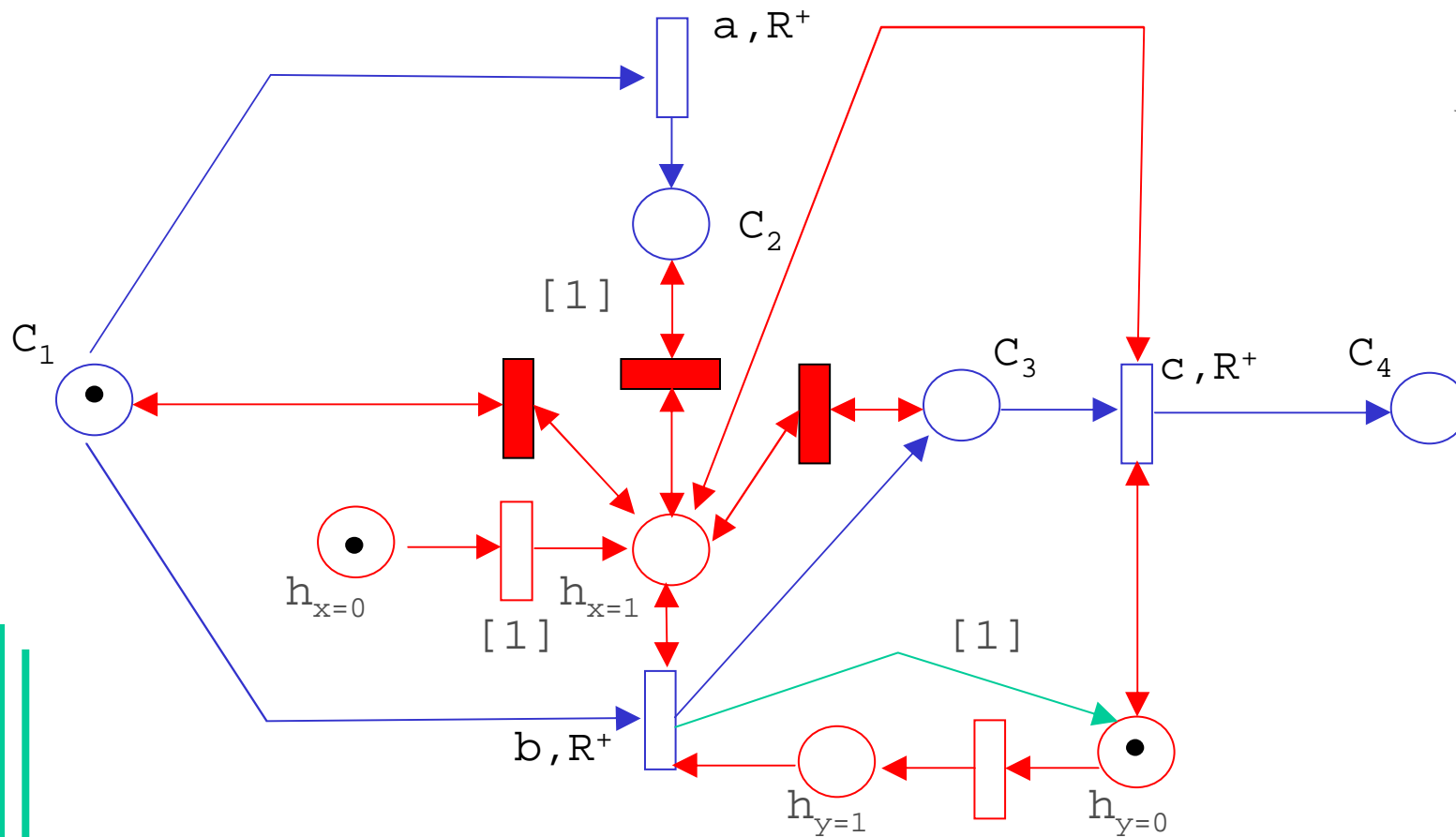
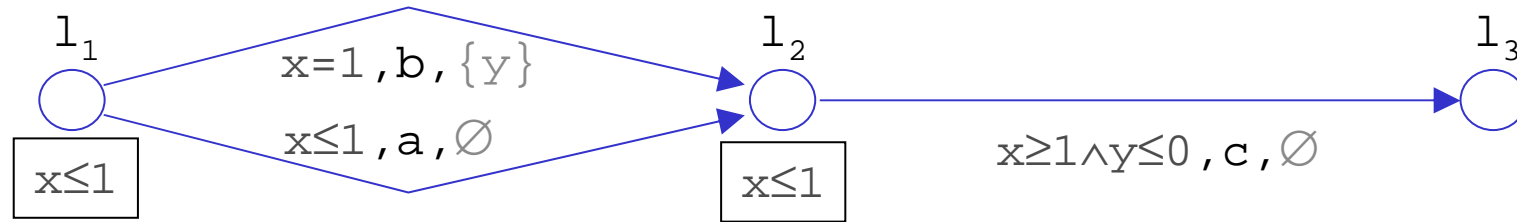
- the point is inside  $[0, K]^X$ .
- the point is strongly time bisimilar to a point of the zone.



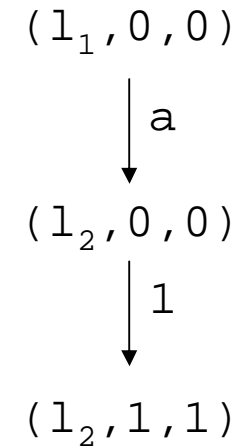
# Illustration: the zone graphe



# Behavioural translation: illustration



Why the zone graph?



# Complexity results

- The reachability problem of bisimulable TA is PSPACE-complete (*adaptation of a construction of Aceto and Laroussinie 2002*).
- The membership problem to the class of bisimulable TA is PSPACE-complete (*exploration of paths in the region automaton and idem*).
- The size of the structural safe TPN is linear w.r.t. the size of the TA.
- The size of the behavioural safe TPN is exponential w.r.t. the size of the TA.

# Conclusion and perspectives

- Characterisation of TA bisimilar to TPNs
- Efficient translation of such a TA into a TPN
- Open questions:
  - a polynomial translation to an integer TPN
  - a characterisation for TA with diagonal constraints
  - a characterisation for TPNs with any kind of intervals
- Experiments of the practical efficiency of TA verification *via* translation