Unfoldings of Networks of Timed Automata

Franck Cassez Thomas Chatain Claude Jard
Patricia Bouyer Serge Haddad Pierre-Alain Reynier

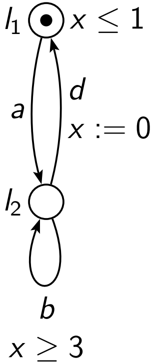
Rennes, December 3, 2008

Unfoldings [McMillan '93]

- First defined for Petri nets
- Then extended to other true concurrency models [Esparza, Römer '99]
- Compact representation of the executions
- Explicit representation of concurrency
- Avoid computation of interleavings
- Model-checking, diagnosis, asynchronous circuits
- Optimization: adequate orders [Esparza, Römer, Vogler '02]

Timed Automata [Alur, Dill '94]

- $\langle L, l_0, \Sigma, X, T, Inv \rangle$
- Transitions $t \stackrel{\text{def}}{=} \langle l, g, a, R, l' \rangle$, with
  - source: $l \stackrel{\text{def}}{=} \alpha(t) \in L$
  - target: $l' \stackrel{\text{def}}{=} \beta(t) \in L$
  - guard: $g \stackrel{\text{def}}{=} \gamma(t)$
  - label: $a \stackrel{\text{def}}{=} \lambda(t) \in \Sigma$
  - resetted clocks: $R \stackrel{\text{def}}{=} \rho(t) \subseteq X$



Timed Automata: Semantics

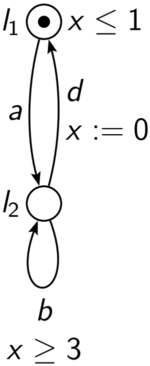
State $\langle l, dor, \theta\rangle$

- location: $l \in L$
- current date: $\theta \in \mathbb{R}$
- date of latest reset for every clock: $\forall x \in X \quad dor(x) \leq \theta$

The transition t can occur at date $\theta' \geq \theta$ from state $\langle l, dor, \theta\rangle$, if:

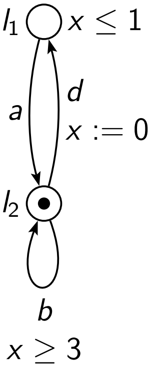
- the invariant of l is satisfied until date θ': $\theta' - dor \models Inv(l)$
- $l = \alpha(t)$
- the guard of t is satisfied at date θ': $\theta' - dor \models \gamma(t)$

Example of Timed Automaton



date: $\theta = 0$
$dor(x) = 0$

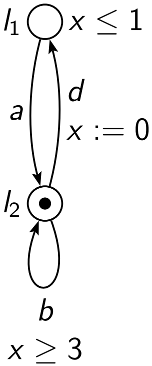
Example of Timed Automaton



date: $\theta = 0.7$
$dor(x) = 0$

$(a, 0.7)$

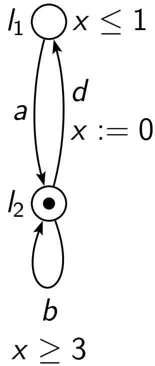
Example of Timed Automaton



date: $\theta = 3$
$dor(x) = 0$

$(a, 0.7), (b, 3)$

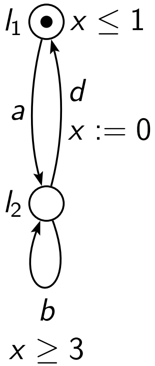
Example of Timed Automaton



date: $\theta = 3.5$
$dor(x) = 0$

$(a, 0.7), (b, 3), (b, 3.5)$

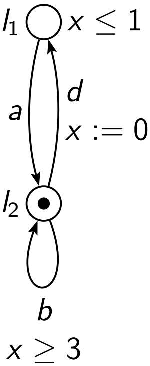
Example of Timed Automaton



date: $\theta = 4$
$dor(x) = 4$

$(a, 0.7), (b, 3), (b, 3.5), (d, 4)$

Example of Timed Automaton

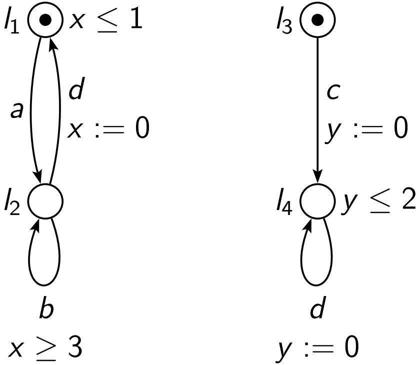


date: $\theta = 5$
$dor(x) = 4$

$(a, 0.7), (b, 3), (b, 3.5), (d, 4), (a, 5)$

Networks of Timed Automata

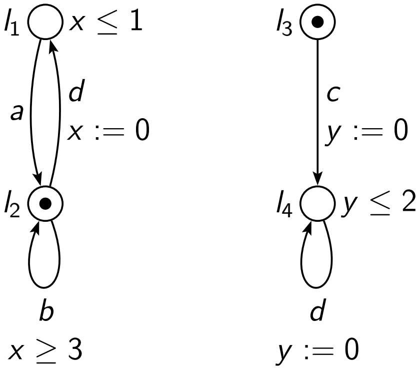
- synchronization by shared labels
- local clocks



date: $\theta = 0$
$dor(x) = 0$
$dor(y) = 0$

Networks of Timed Automata

- synchronization by shared labels
- local clocks



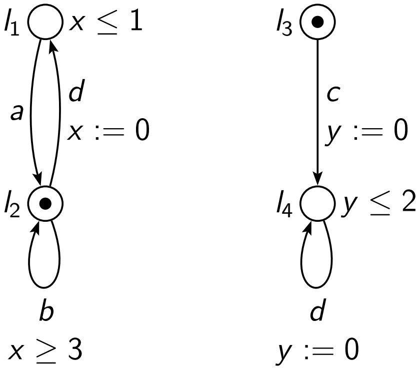
date: $\theta = 0.7$
$dor(x) = 0$
$dor(y) = 0$

$(a, 0.7)$

Networks of Timed Automata

- synchronization by shared labels
- local clocks

date: $\theta = 3$
$dor(x) = 0$
$dor(y) = 0$

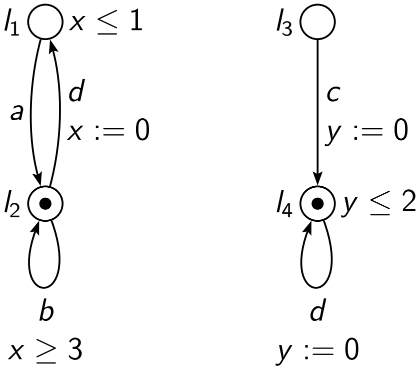


$(a, 0.7), (b, 3)$

Networks of Timed Automata

- synchronization by shared labels
- local clocks

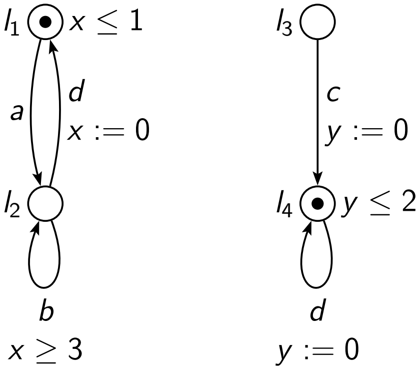
date: $\theta = 4$
$dor(x) = 0$
$dor(y) = 4$



$(a, 0.7), (b, 3), (c, 4)$

Networks of Timed Automata

- synchronization by shared labels
- local clocks



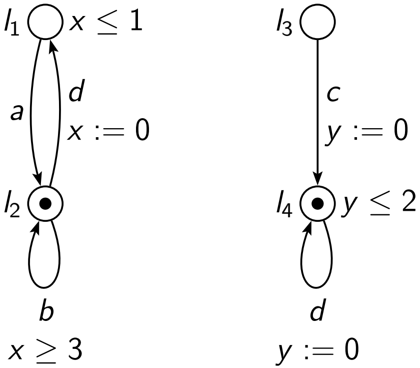
date: $\theta = 4$
$dor(x) = 4$
$dor(y) = 4$

$(a, 0.7), (b, 3), (c, 4), (d, 4)$

Networks of Timed Automata

- synchronization by shared labels
- local clocks

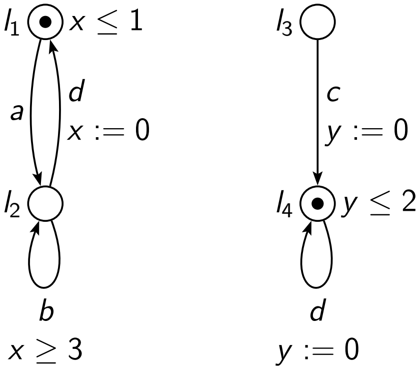
date: $\theta = 5$
$dor(x) = 4$
$dor(y) = 4$



$(a, 0.7), (b, 3), (c, 4), (d, 4), (a, 5)$

Networks of Timed Automata

- synchronization by shared labels
- local clocks

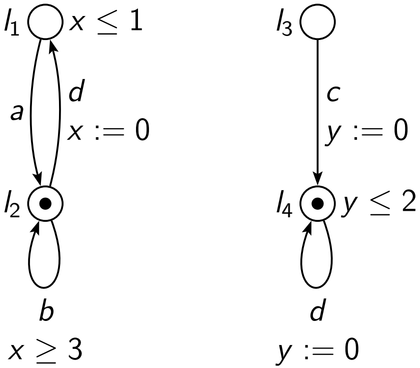


date: $\theta = 6$
$dor(x) = 6$
$dor(y) = 6$

$(a, 0.7), (b, 3), (c, 4), (d, 4), (a, 5), (d, 6)$

Networks of Timed Automata

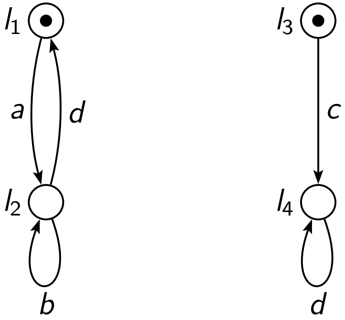
- synchronization by shared labels
- local clocks



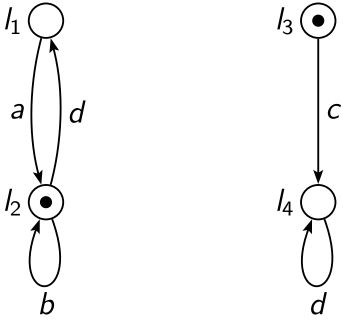
date: $\theta = 7$
$dor(x) = 6$
$dor(y) = 6$

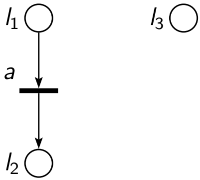
$(a, 0.7), (b, 3), (c, 4), (d, 4), (a, 5), (d, 6), (a, 7)$

Processes of Networks of *Untimed* Automata



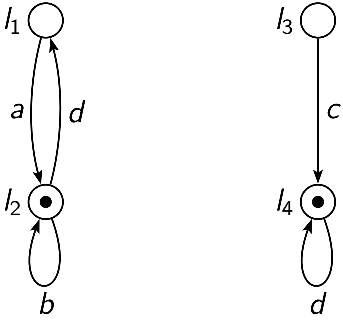
Processes of Networks of *Untimed* Automata

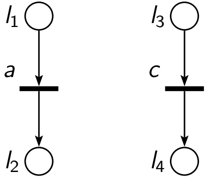




a

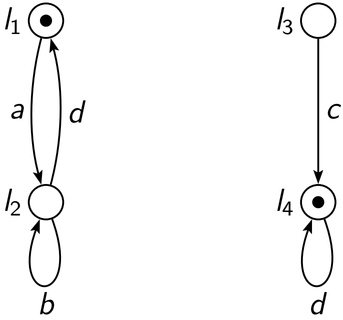
Processes of Networks of *Untimed* Automata

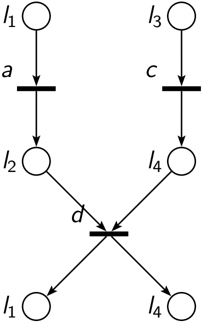




a, c

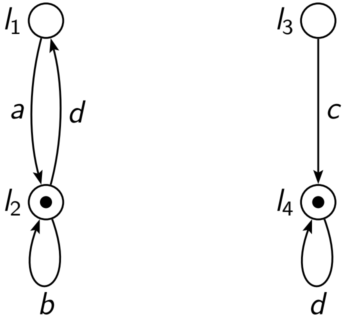
Processes of Networks of *Untimed* Automata

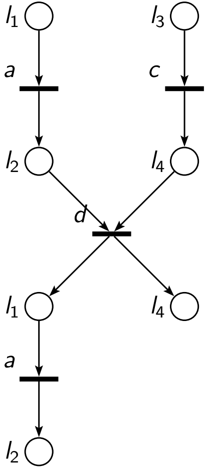




a, c, d

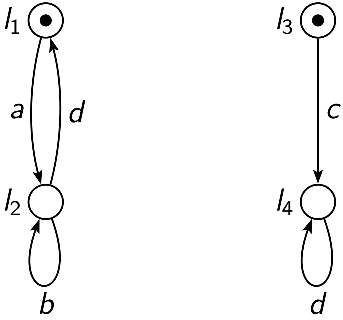
Processes of Networks of *Untimed* Automata

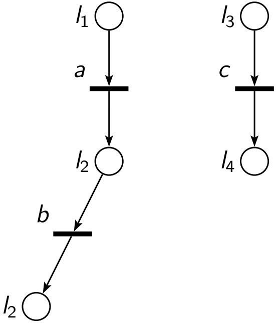




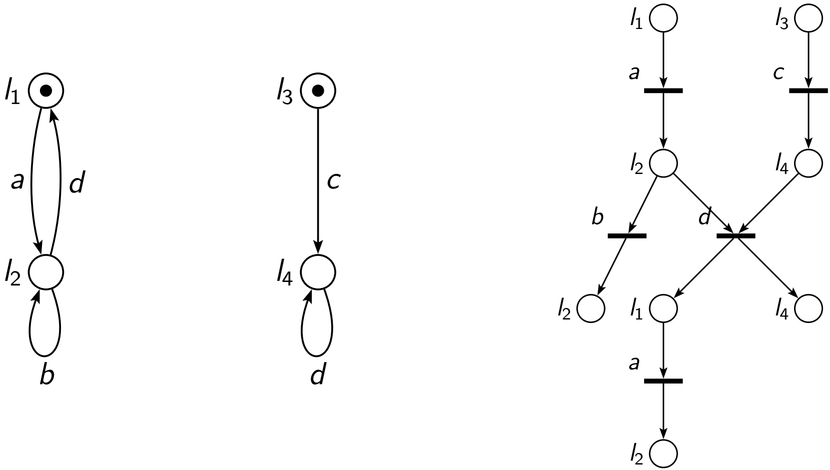
a, c, d, a

Processes of Networks of *Untimed* Automata



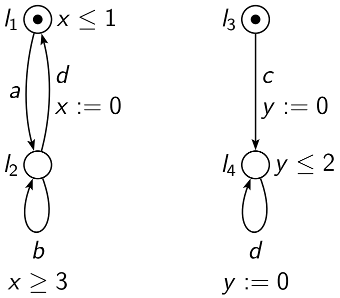


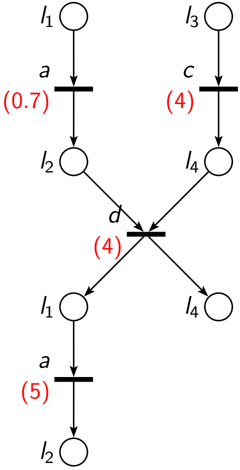
Unfoldings of Networks of *Untimed* Automata



Processes of NTA

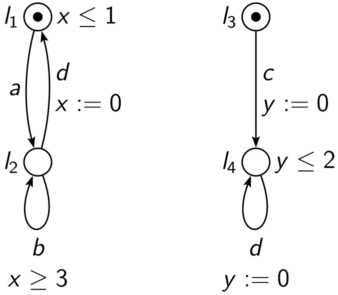
$(a, 0.7), (c, 4), (d, 4), (a, 5)$

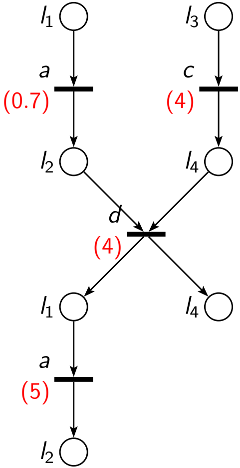




Processes of NTA

$(a, 0.7), (c, 4), (d, 4), (a, 5)$

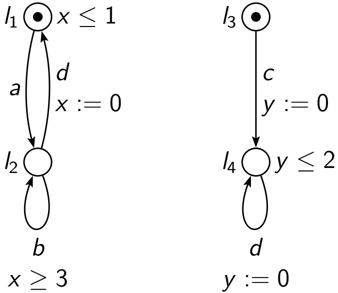


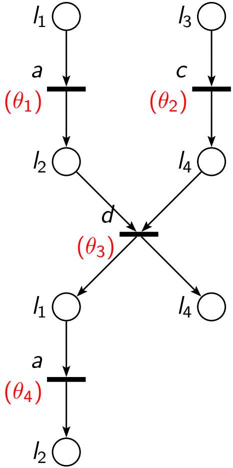


Other dates are possible with the same structure

Symbolic Processes of NTA

$(a, \theta_1), (c, \theta_2), (d, \theta_3), (a, \theta_4)$





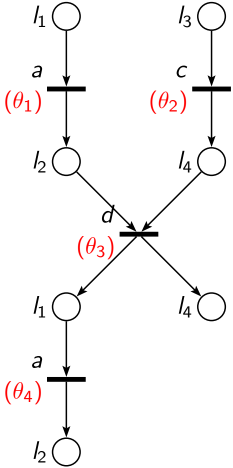
Other dates are possible with the same structure $\rightarrow$ parameters

Symbolic Processes of NTA: Symbolic constraints

- induced by
  - guards
  - invariants
  - causality
- convex union of zones [Ben Salah, Bozga, Maler, '06]
- analog of [Aura, Lilius, '00] for NTA

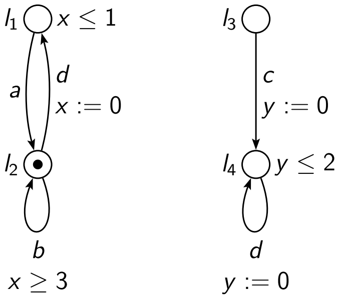
$$\begin{aligned} \theta_1 &\le 1 \\ \theta_3 - \theta_2 &\le 2 \\ \theta_4 - \theta_3 &\le 1 \\ \theta_1 &\le \theta_3 \end{aligned}$$

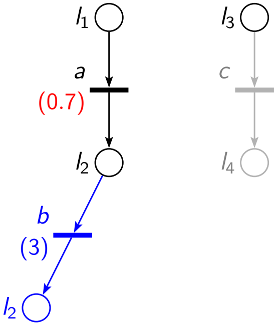
$$\begin{aligned} \theta_2 &\le \theta_3 \\ \theta_3 &\le \theta_4 \\ \theta_4 - \theta_3 &\le 2 \end{aligned}$$



Difficulties with Time in Unfoldings

- In untimed nets, feasability of an event is a local property.
- In NTA, it depends on the context.
- In simple case (no invariants), it is still a local property.

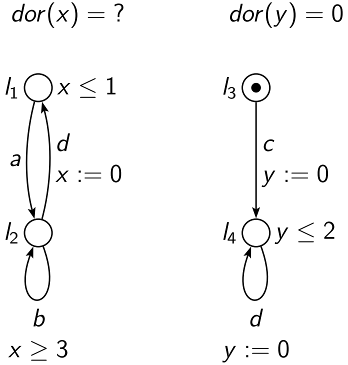




Concurrent Operational Semantics for NTA

How to simulate a NTA without using clocks,
but with as much concurrency as possible?

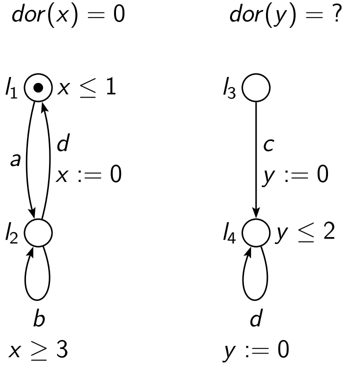
- Look for local conditions to play a transition.
- Executions must respect the usual semantics.
- Notion of partial state L: for each automaton, either $\langle l_i, dor_i, \theta_i \rangle$ or $\bullet$.



Concurrent Operational Semantics for NTA

How to simulate a NTA without using clocks,
but with as much concurrency as possible?

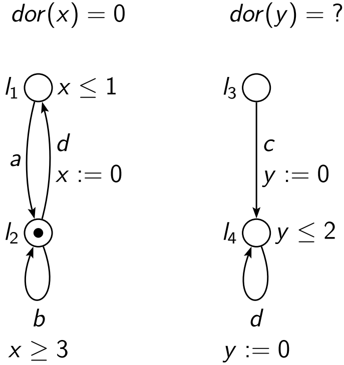
- Look for local conditions to play a transition.
- Executions must respect the usual semantics.
- Notion of partial state L: for each automaton, either $\langle l_i, dor_i, \theta_i \rangle$ or $\bullet$.



Concurrent Operational Semantics for NTA

How to simulate a NTA without using clocks,
but with as much concurrency as possible?

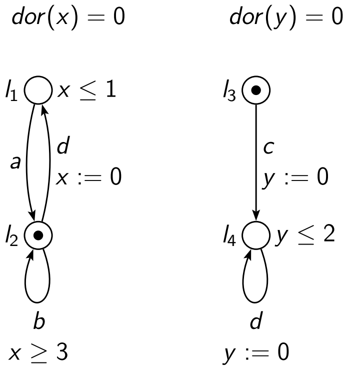
- Look for local conditions to play a transition.
- Executions must respect the usual semantics.
- Notion of partial state L: for each automaton, either $\langle l_i, dor_i, \theta_i \rangle$ or $\bullet$.



Concurrent Operational Semantics for NTA

How to simulate a NTA without using clocks,
but with as much concurrency as possible?

- Look for local conditions to play a transition.
- Executions must respect the usual semantics.
- Notion of partial state L: for each automaton, either $\langle l_i, dor_i, \theta_i \rangle$ or $\bullet$.



Local Conditions to Take Transitions

To take t at θ from L, we want:

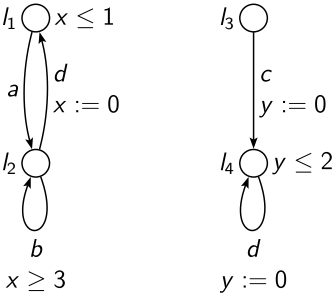
for all context S of L, t can occur at θ from $L \cup S$.

We have:

t can occur at θ from $L \cup S$

if

$\begin{cases} \text{the automata concerned by } t \text{ agree} \\ \text{no invariant in } L \cup S \text{ expires before } \theta \end{cases}$



Local Conditions to Take Transitions

To take t at θ from L, we want:

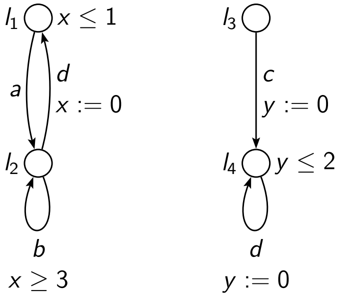
for all context S of L, t can occur at θ from $L \cup S$.

We have:

t can occur at θ from $L \cup S$

if

$$\begin{cases} \text{the automata concerned by } t \text{ agree} \\ L \text{ is stable in } S \text{ until } \theta \end{cases}$$



Local Stability Condition

Intuition

$$LSC(L, \theta) \implies$$

for all context S of L, L is stable in S until θ.

Completeness

Global states are stable until the date where one of their invariants expires.

Local Stability Condition

Several choices to define $LSC(L, \theta)$:

- trivial choice: L is a global state.
- BHR: L involves all the automata that have invariants.
- more generic: L contains enough information to check that no automaton of L may be forced to synchronize earlier than θ with another automaton.

A proposition for $LSC(L, \theta)$

Definition: $LSC(L, \theta)$ holds iff

L contains enough information to check that no automaton of L may be forced to synchronize earlier than θ with another automaton:

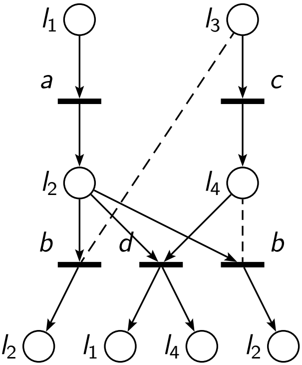
$$\begin{cases} \forall i \in J_L \quad \theta - dor_i \models Inv_i(l_i) \\ \forall t \in Sync \\ \qquad I_t \cap J_L \neq \emptyset \implies \begin{cases} \quad I_t \subseteq J_L \\ \vee \quad \exists i \in I_t \cap J_L \quad l_i \neq \alpha_i(t_i) \\ \vee \quad \exists i \in I_t \cap J_L \quad \theta - dor_i \not\models \gamma_i(t_i) \\ \vee \quad \forall i \in I_t \setminus J_L \quad Inv(\alpha_i(t_i)) \equiv \mathtt{true} \end{cases} \end{cases}$$

where

- I_t is the set of automata involved in transition t;
- J_L is the set of automata whose state is defined in the partial state L.

Symbolic Unfoldings of NTA

- In symbolic unfoldings: keep track of all the partial state L (not only the part that participates in t) $\longrightarrow$ use read arcs.
- Any configuration (process) of the unfolding maps (by removing the read arcs) to a pre-process (i.e. a prefix of a process) of the NTA.
- Use only minimal sets L to increase concurrency.



Conclusion

- concurrent operational semantics for NTA
- parameterized local stability condition
- solve constraints on the dates of the events
- study of the form of the constraints
  $\longrightarrow$ finite complete prefix of the unfolding
- if there is no urgency, the unfolding is simply the superimposition of the processes