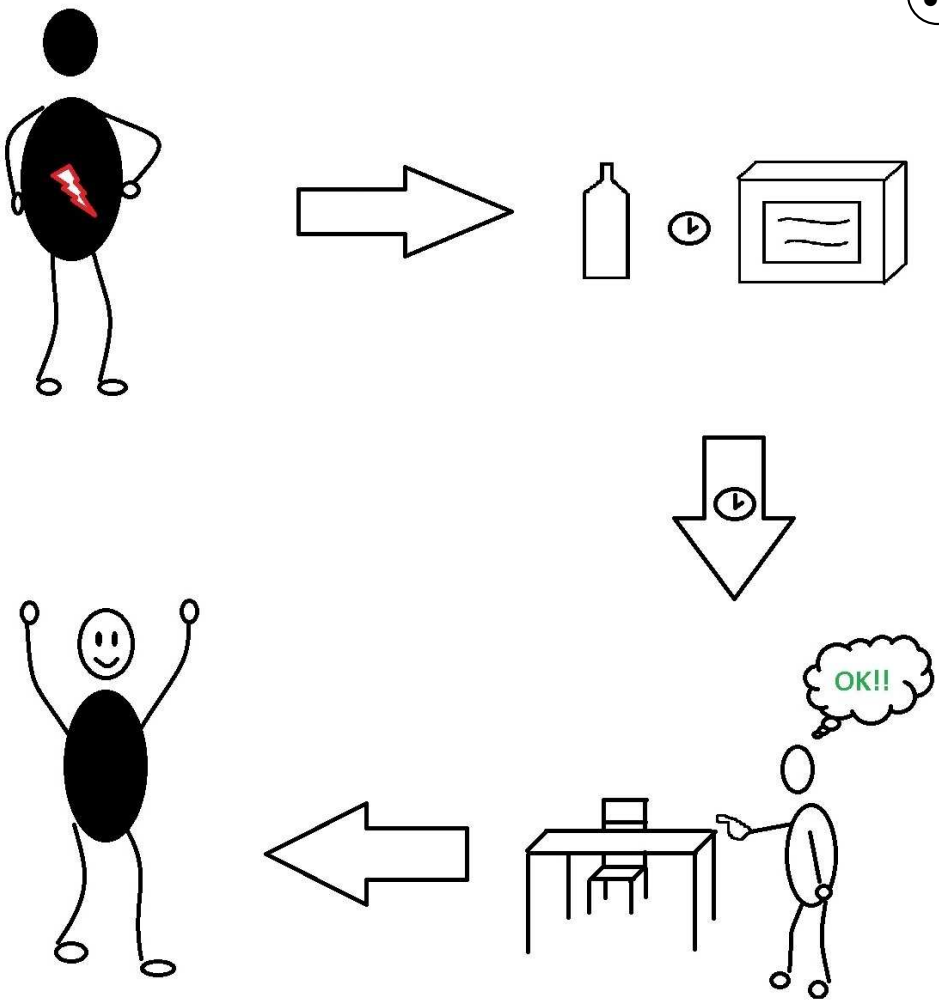
Timed extensions of Petri nets with data

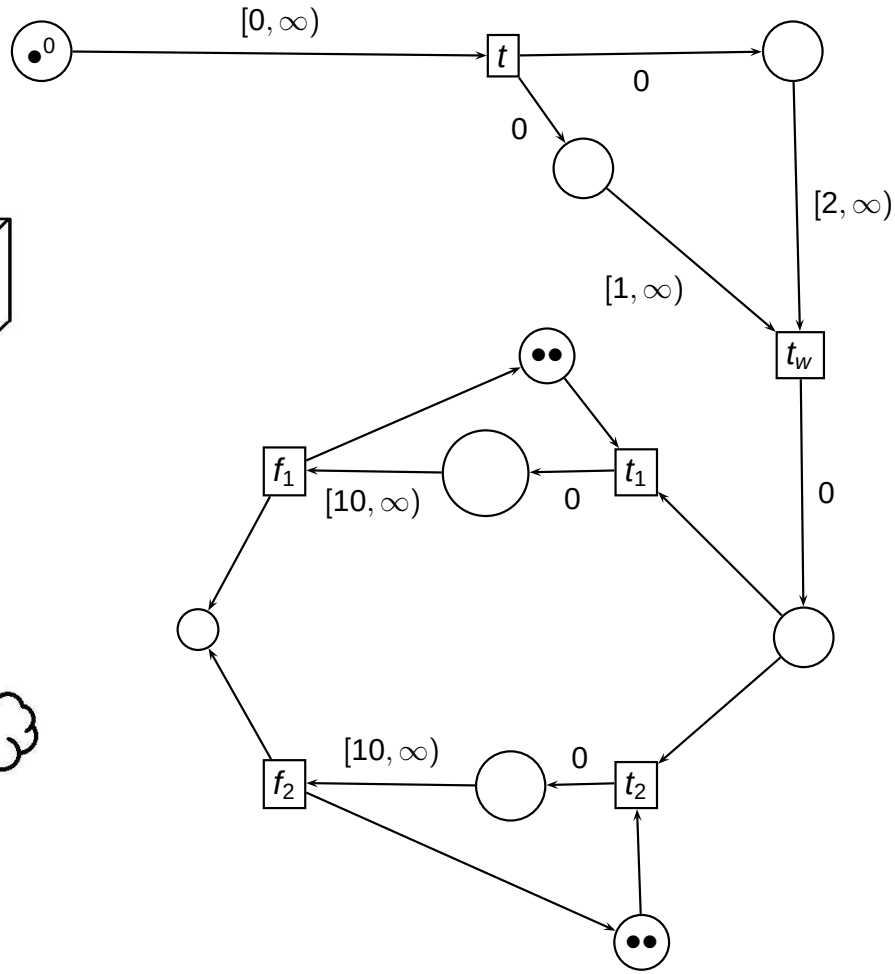
María Martos-Salgado and Fernando Rosa-Velardo

Universidad Complutense de Madrid.
2nd Workshop on Parameterized Verification, PV 2015

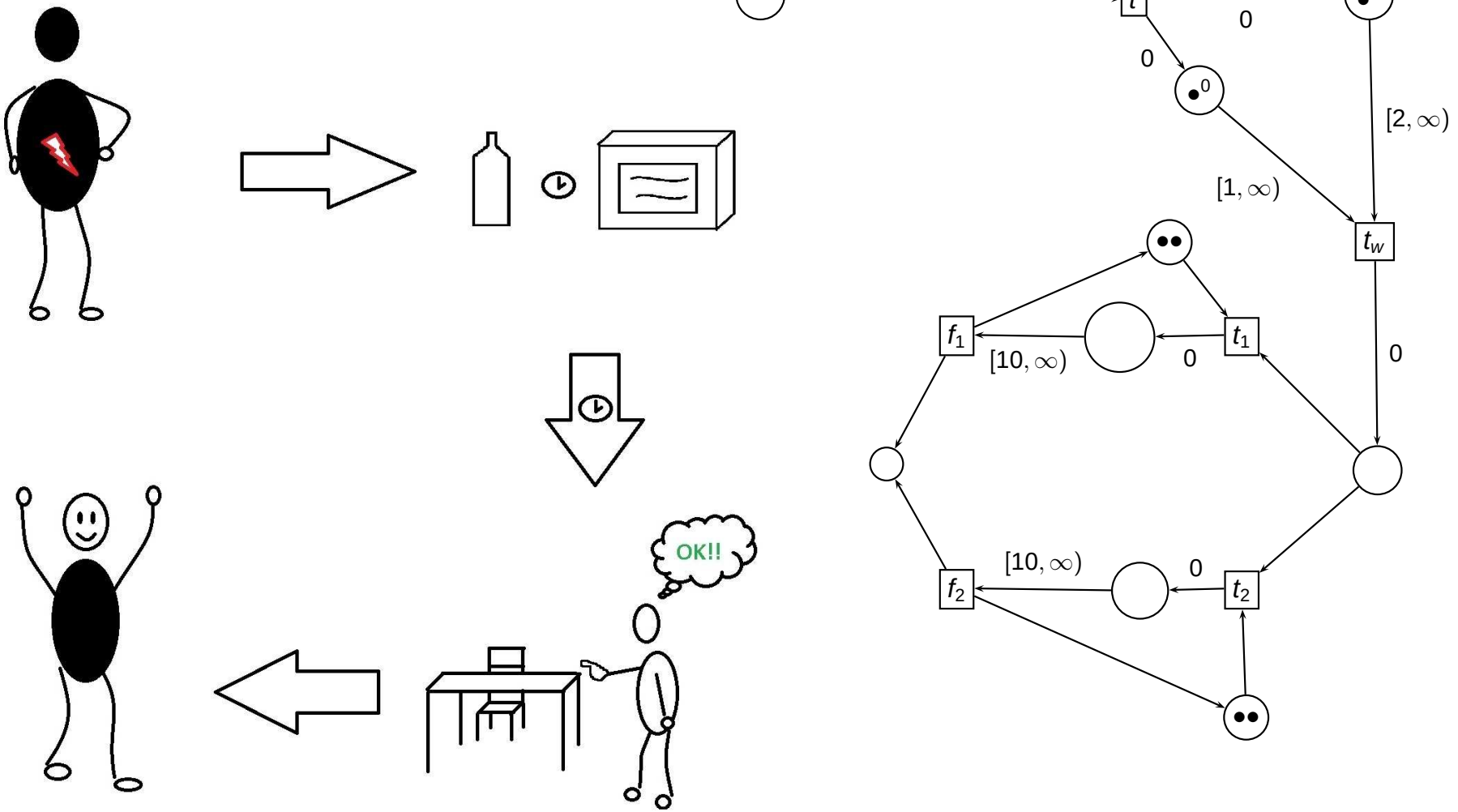
Madrid
September 5^{th} 2015

Timed Petri Nets

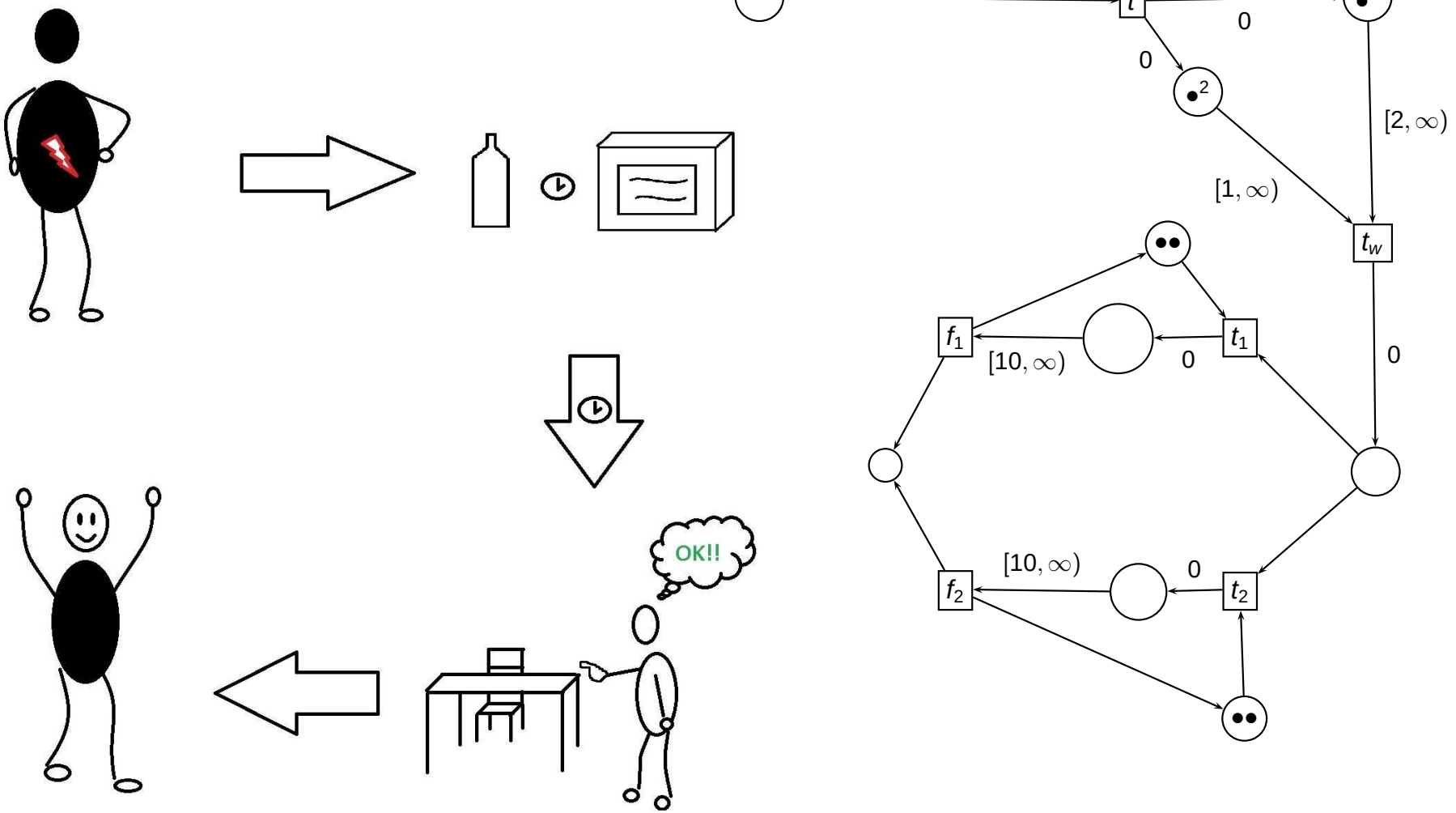




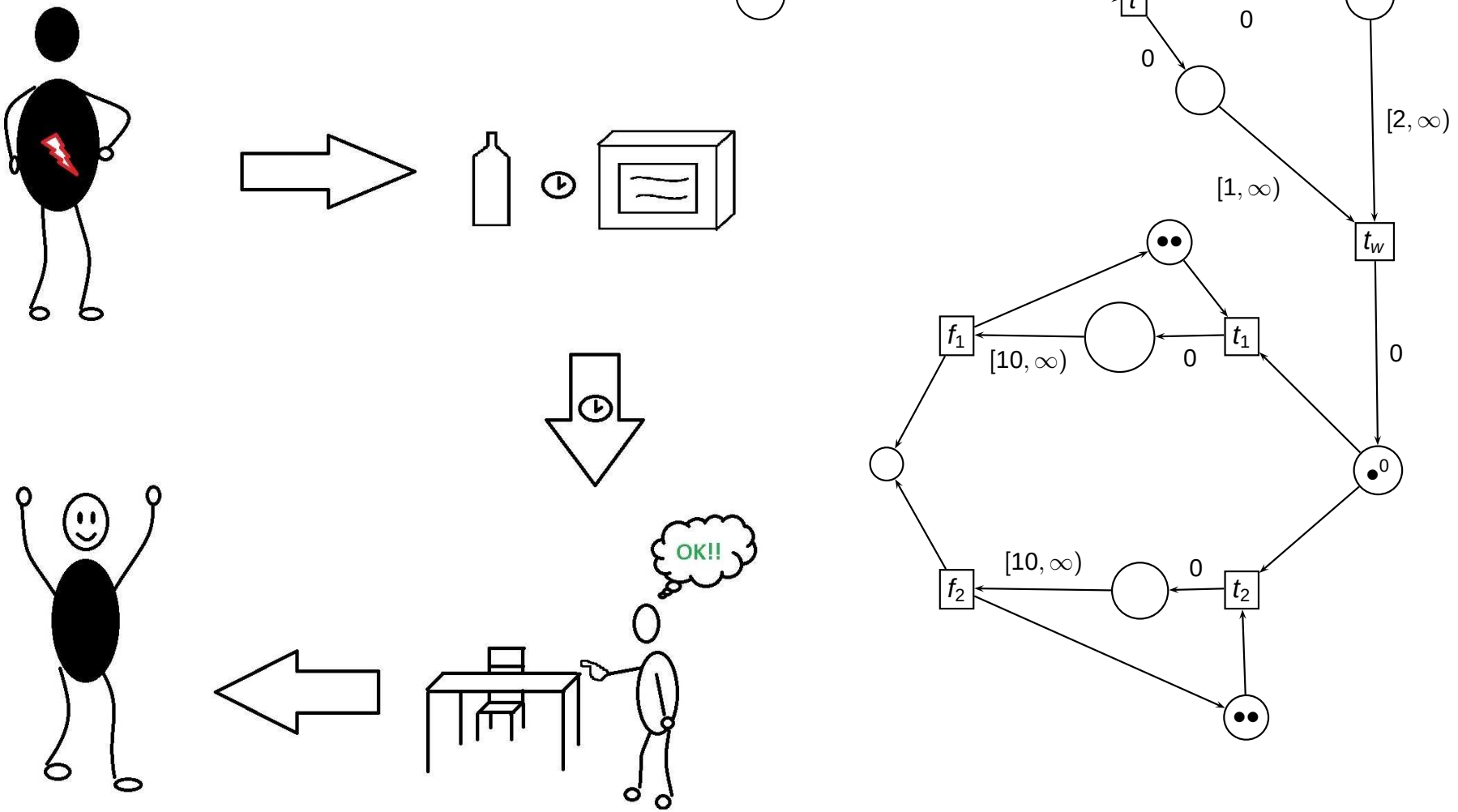
Timed Petri Nets



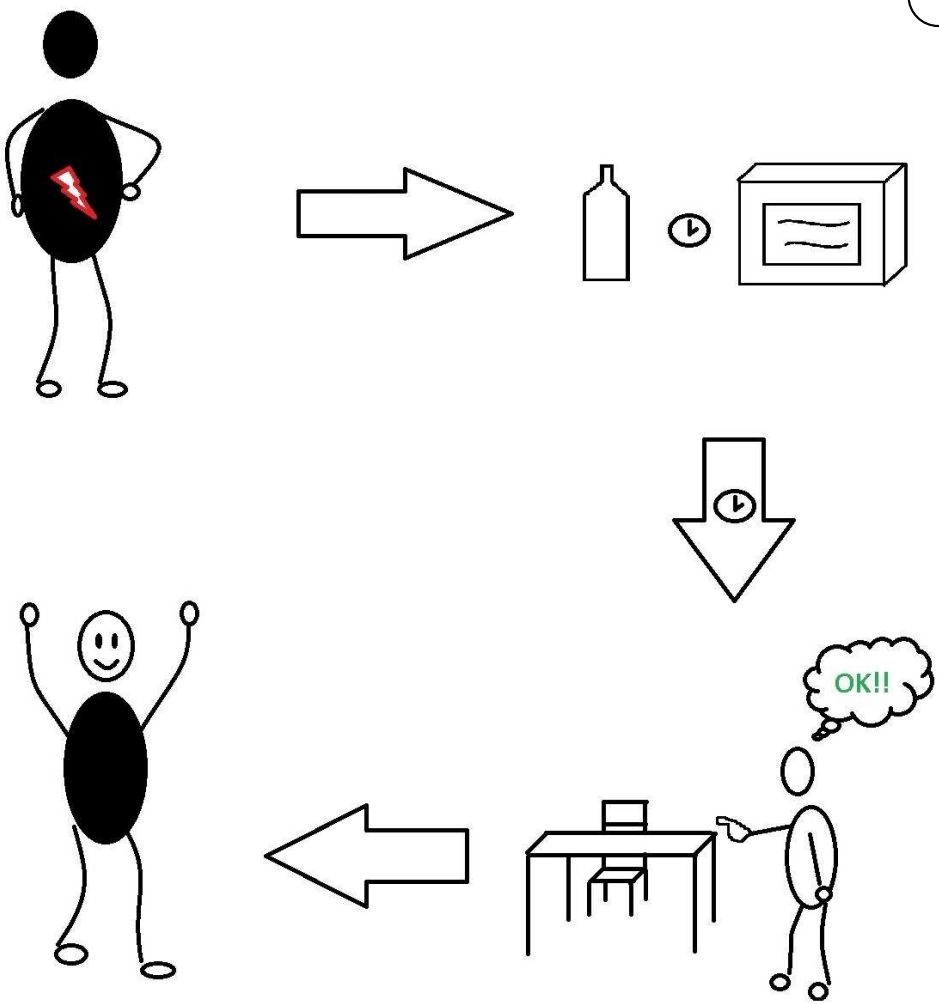
Timed Petri Nets

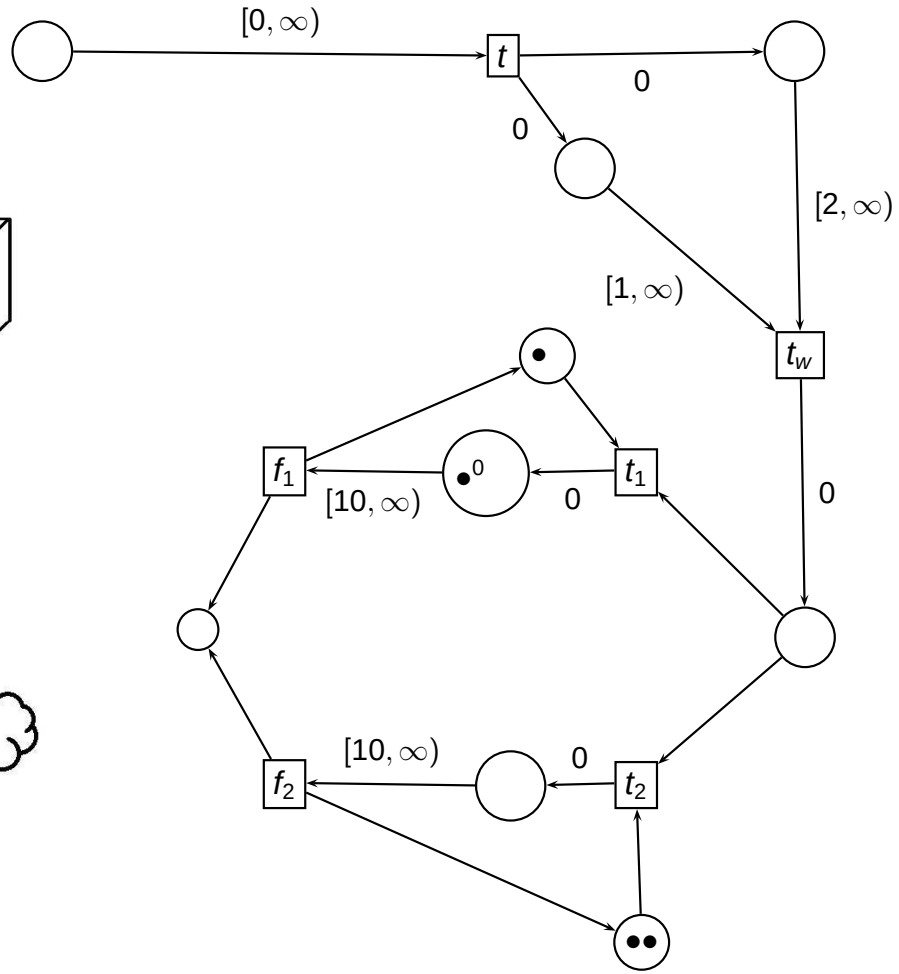


Timed Petri Nets

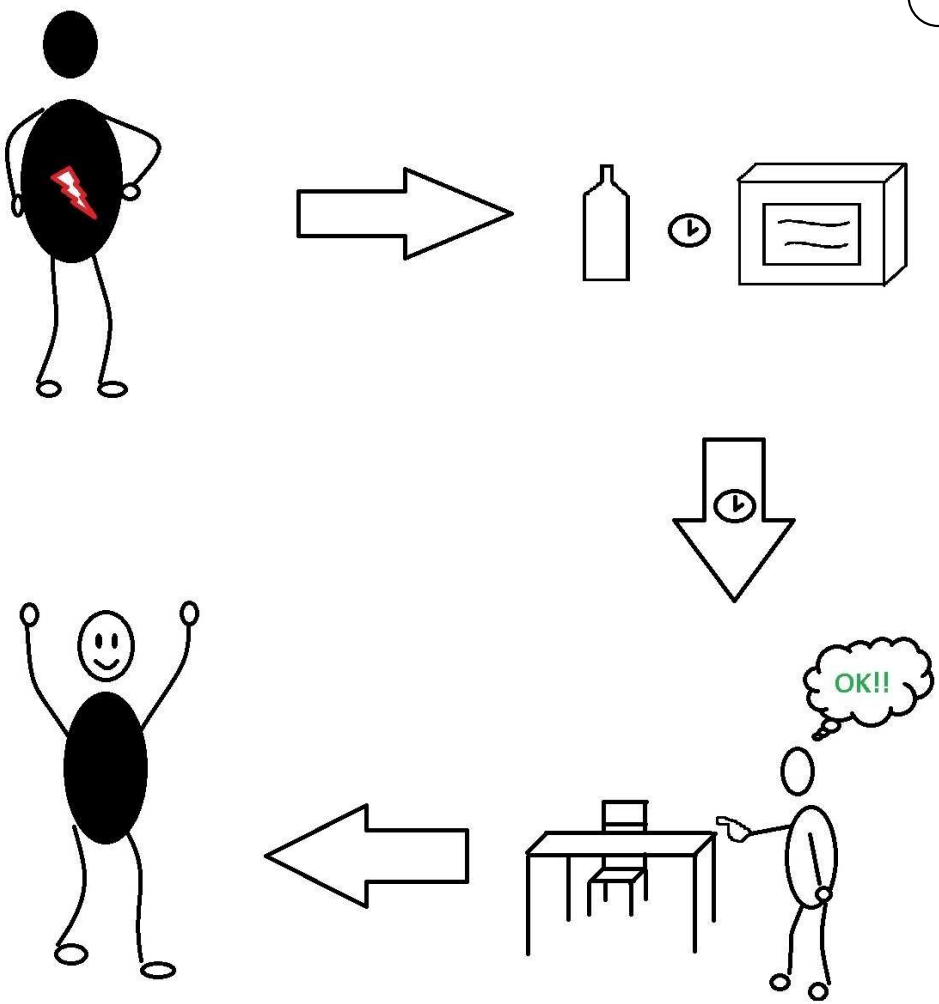


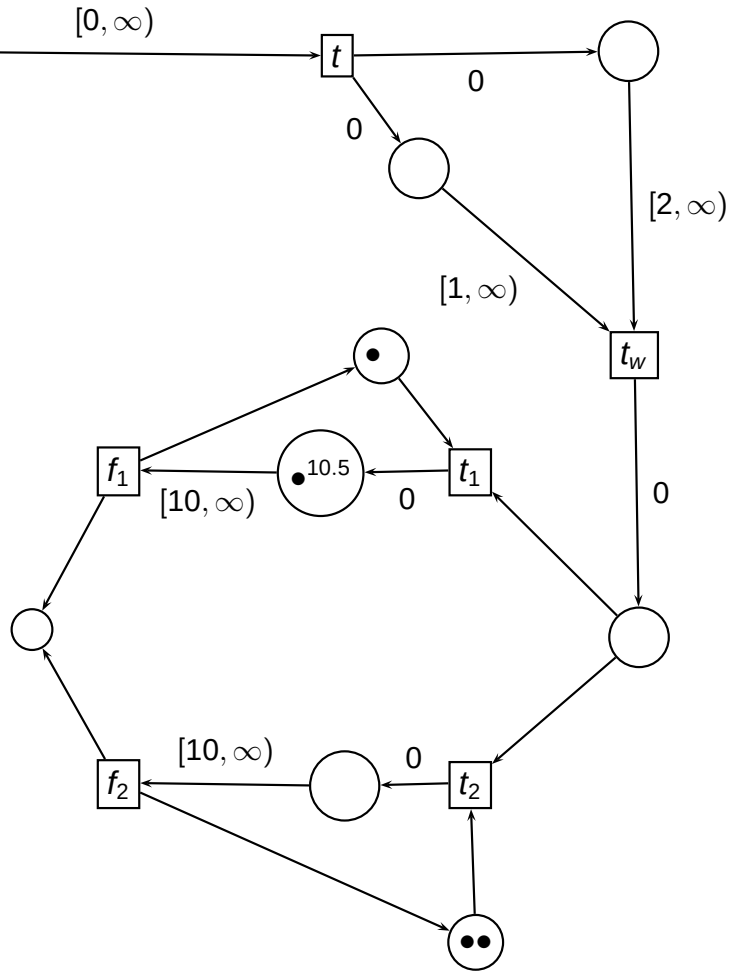
Timed Petri Nets



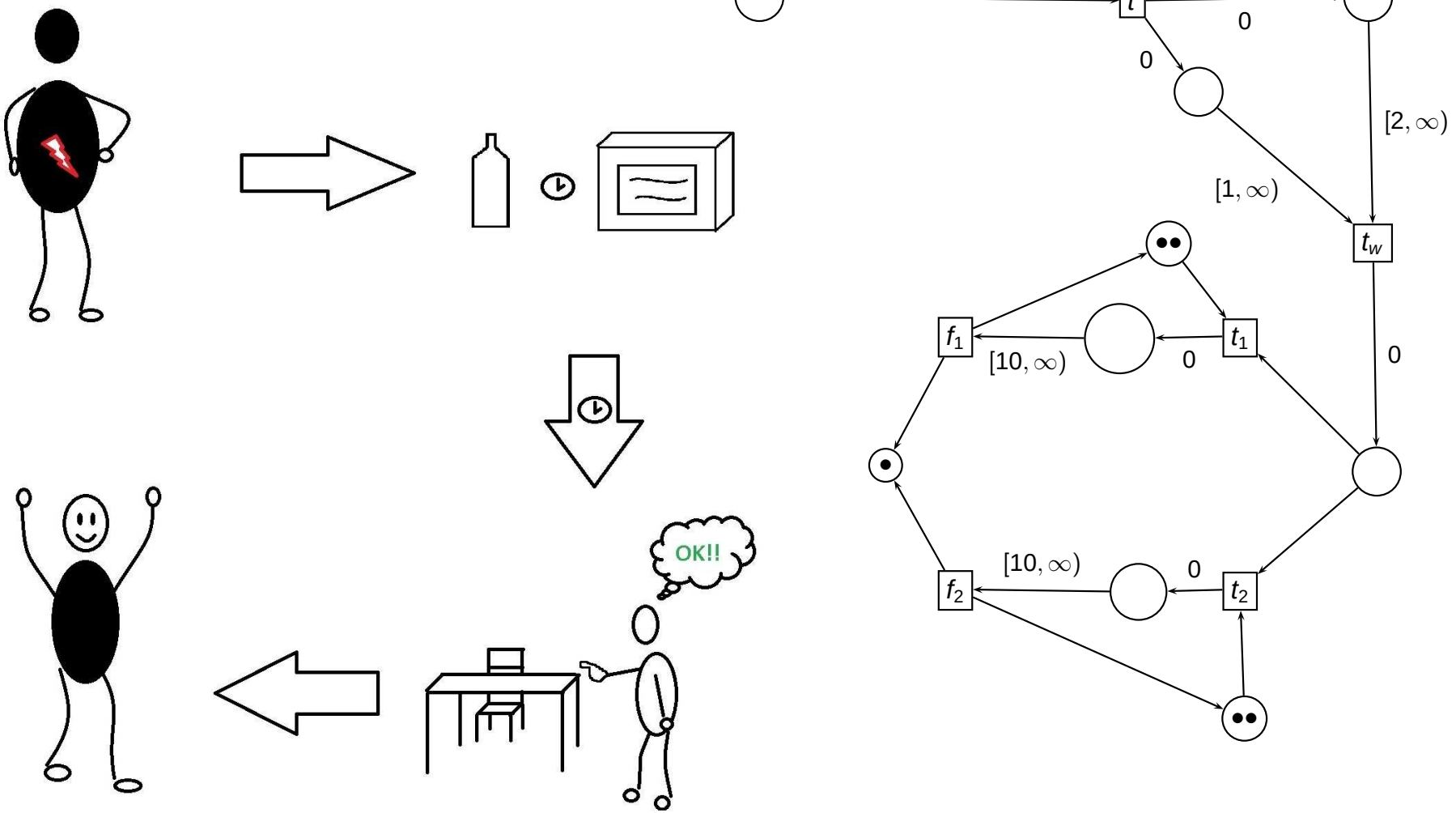


Timed Petri Nets





Timed Petri Nets



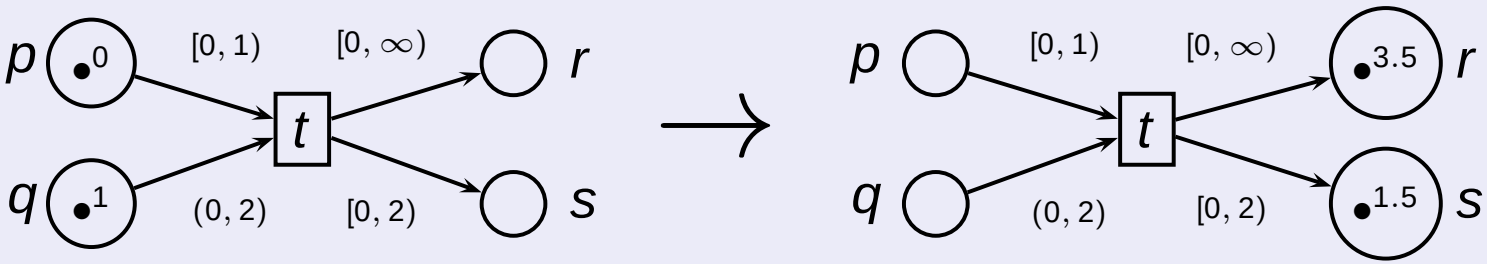
Timed Petri Nets

Definition

A *Timed Petri Net* (*TdPN*) is a tuple $N = \langle P, T, F, H \rangle$, where:

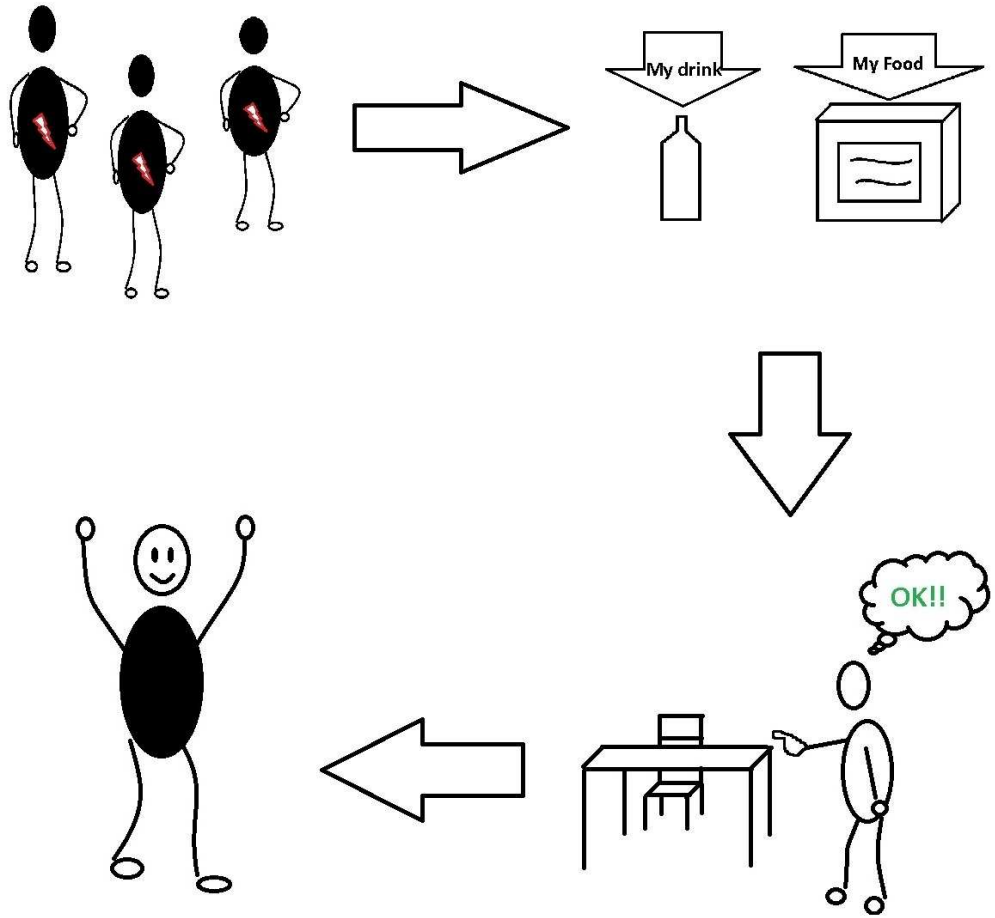
- P and T are finite disjoint sets of places and transitions.
- $F, H : P \times T \to \mathcal{I}^{\oplus}$.

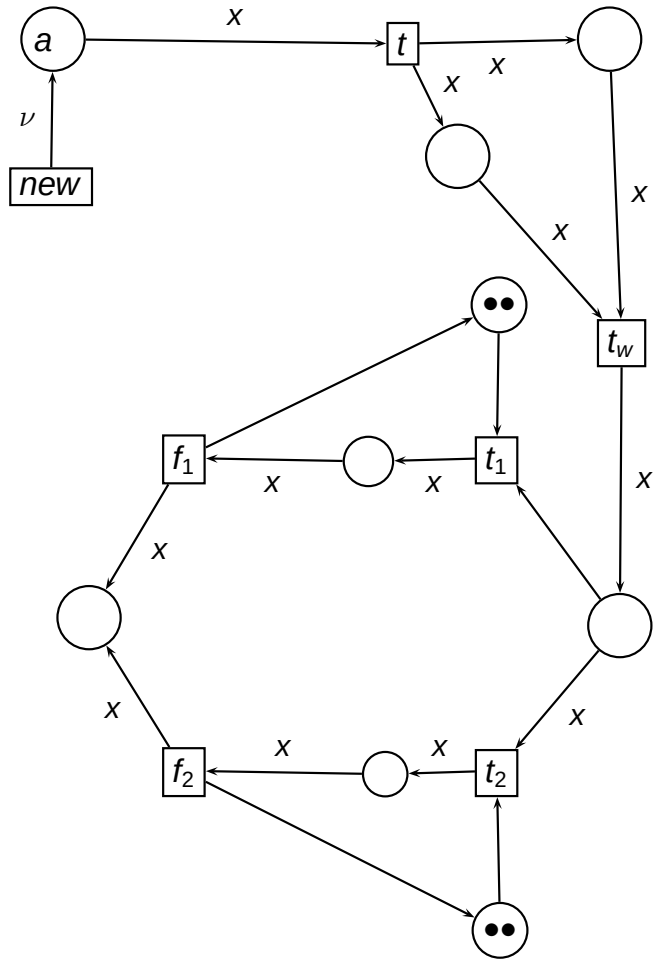
A marking of a *TdPN* is a finite multiset M over $P \times \mathbb{R}_{\geq 0}$.



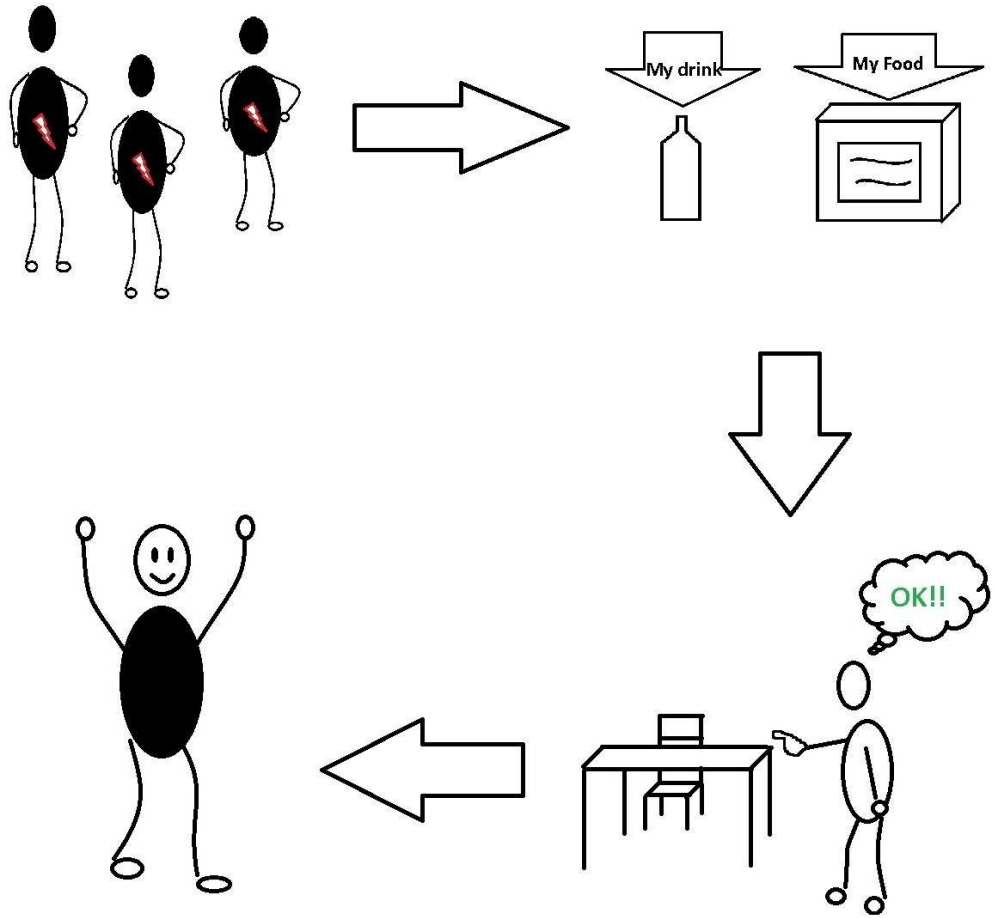
$F(p,t) = \{[0,1)\}$, $H(s,t) = \{[0,2)\}$.

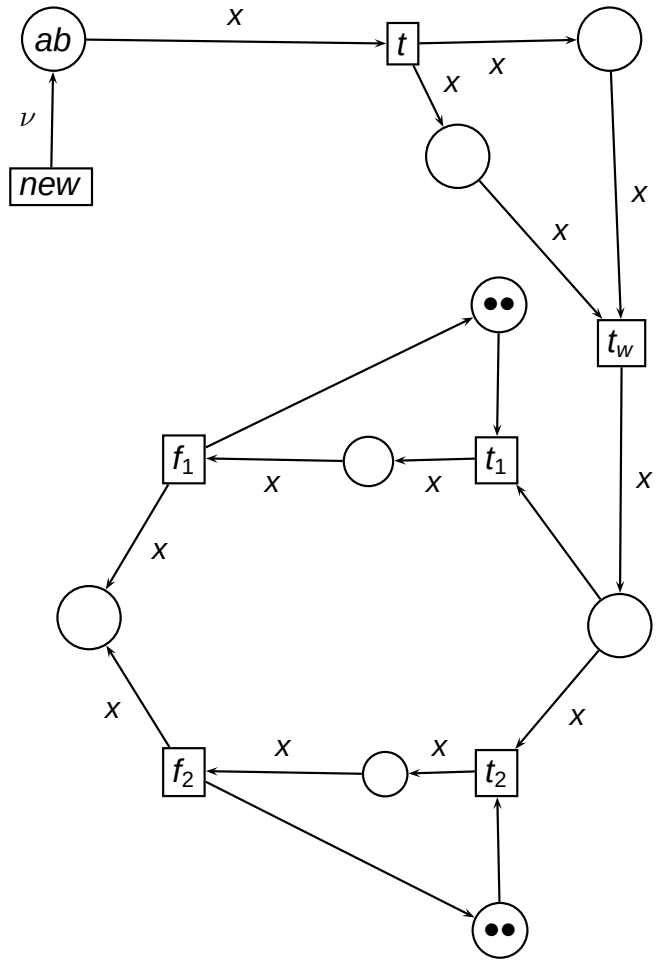
ν-Petri Nets



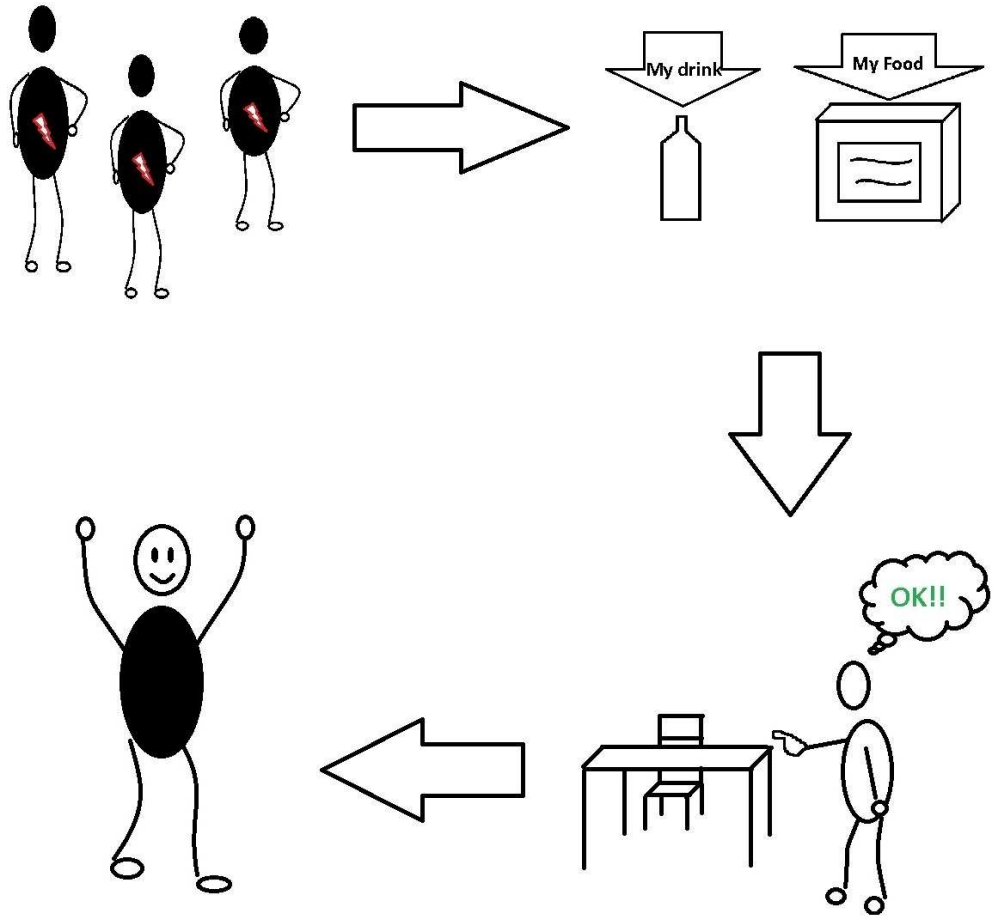


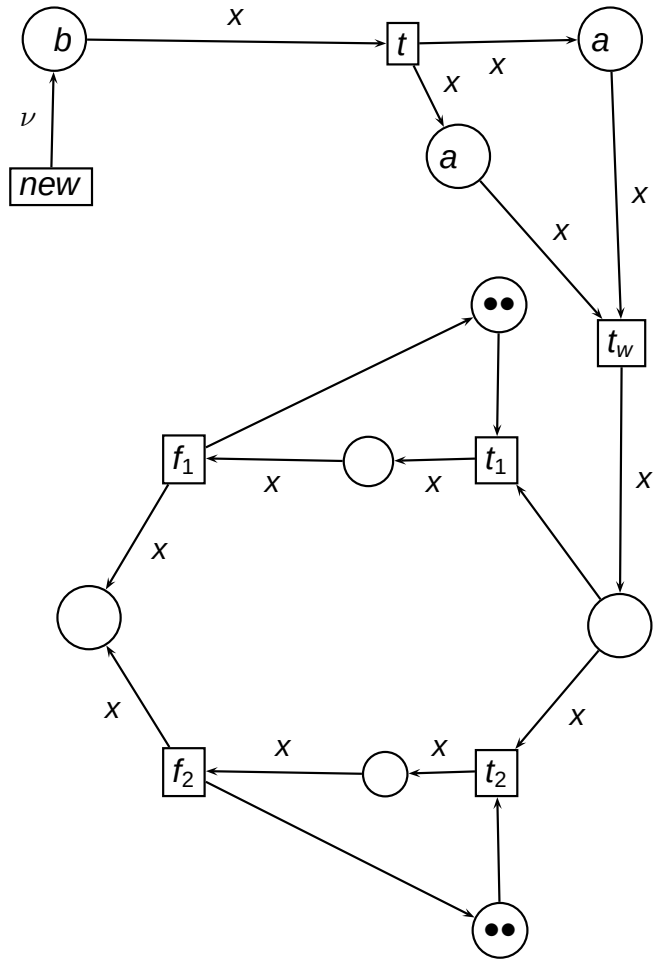
ν-Petri Nets



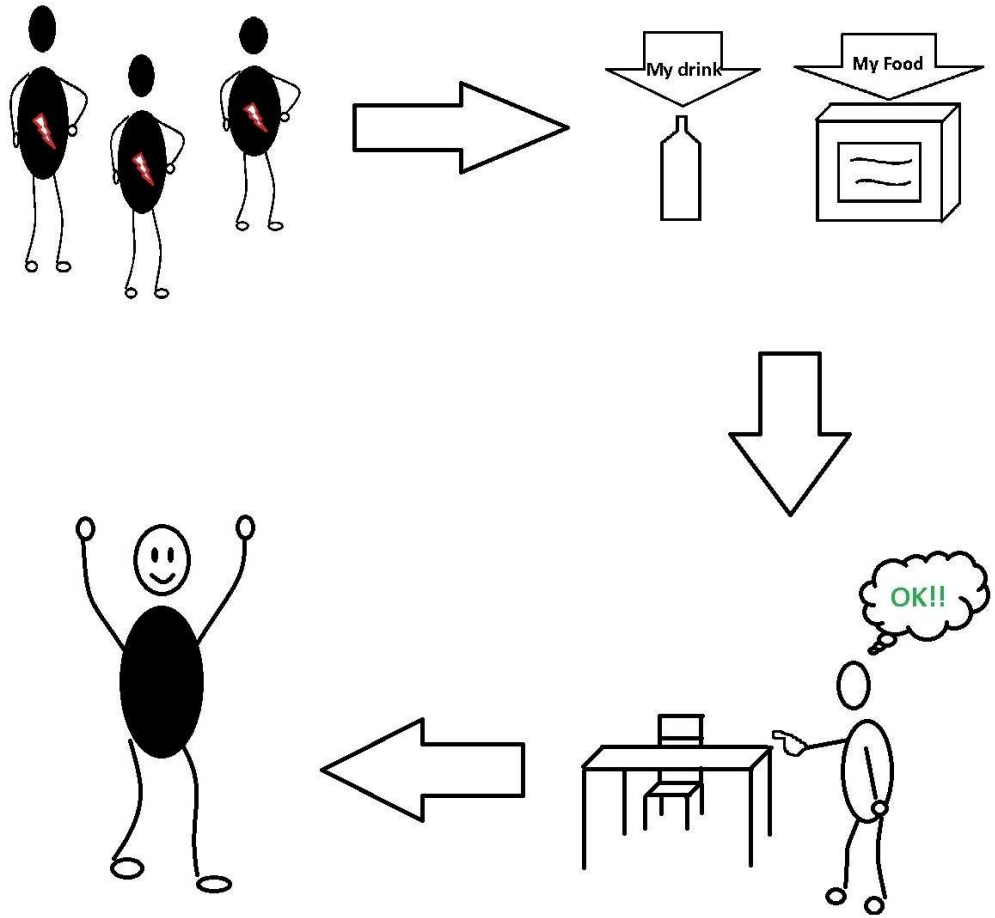


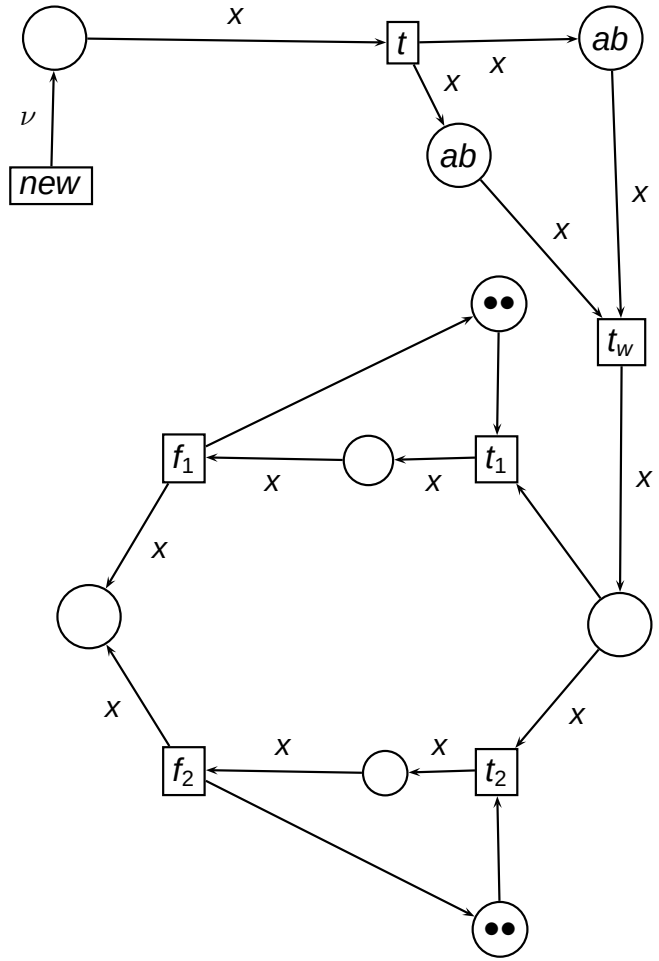
ν-Petri Nets



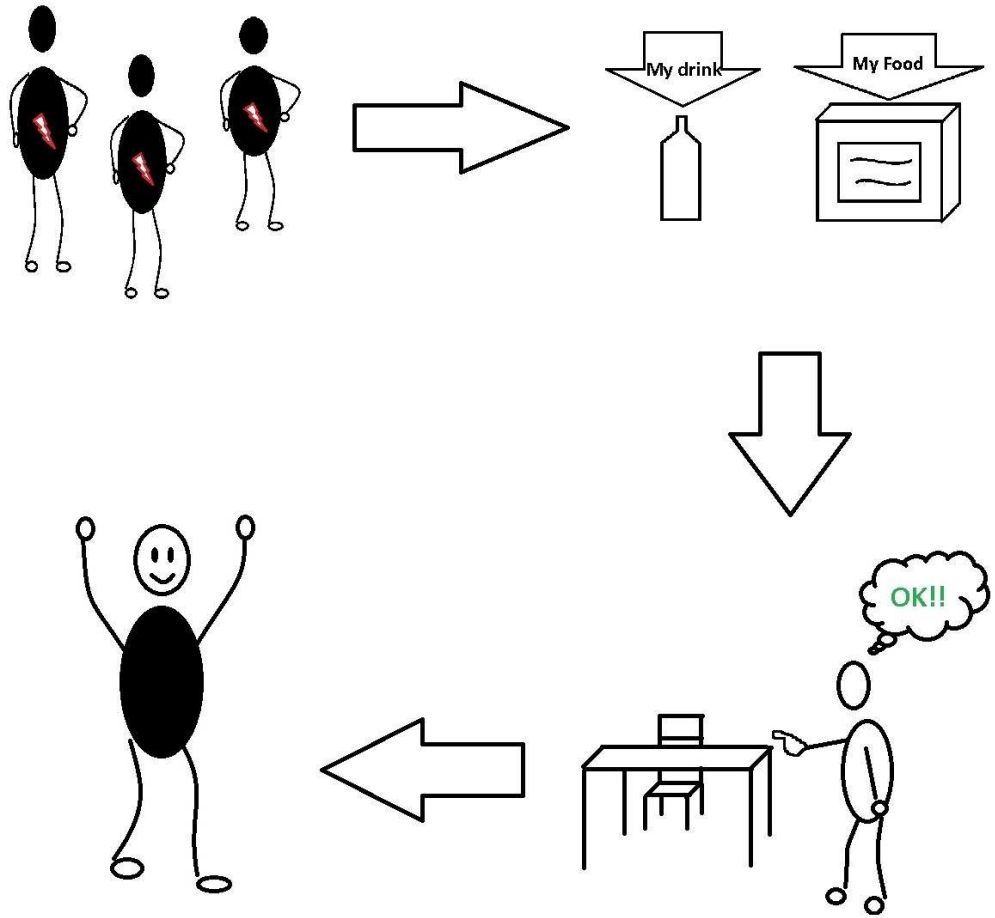


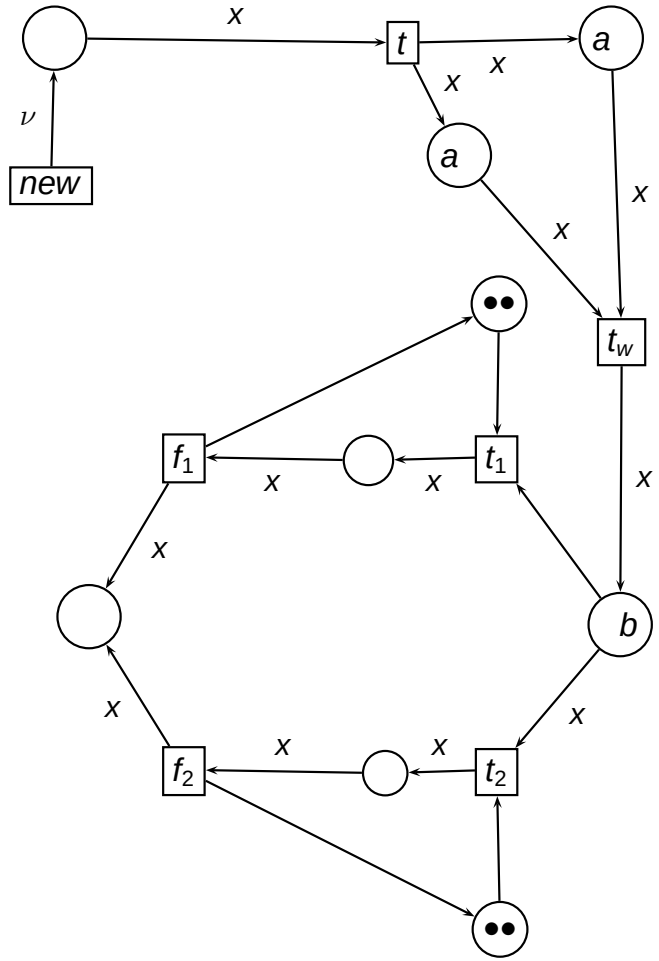
ν-Petri Nets



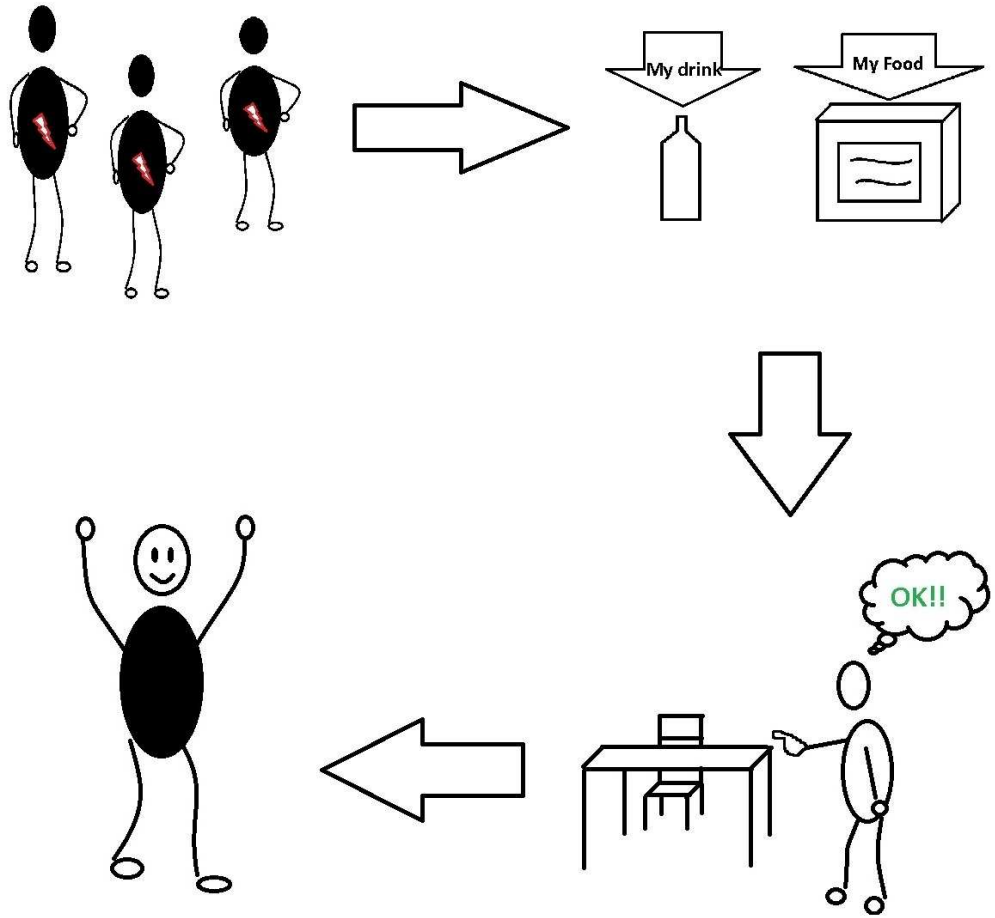


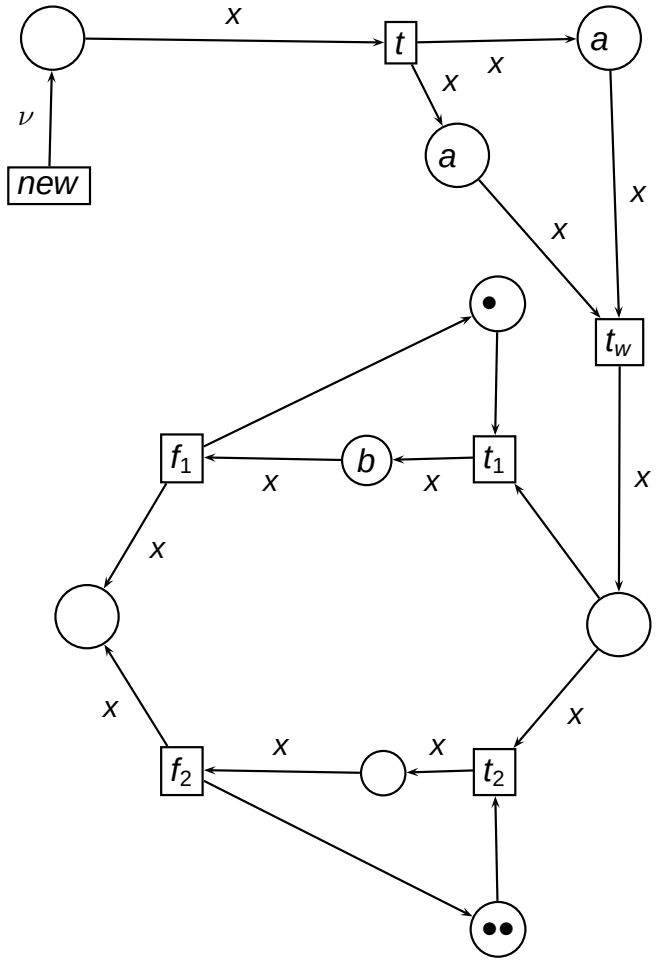
ν-Petri Nets



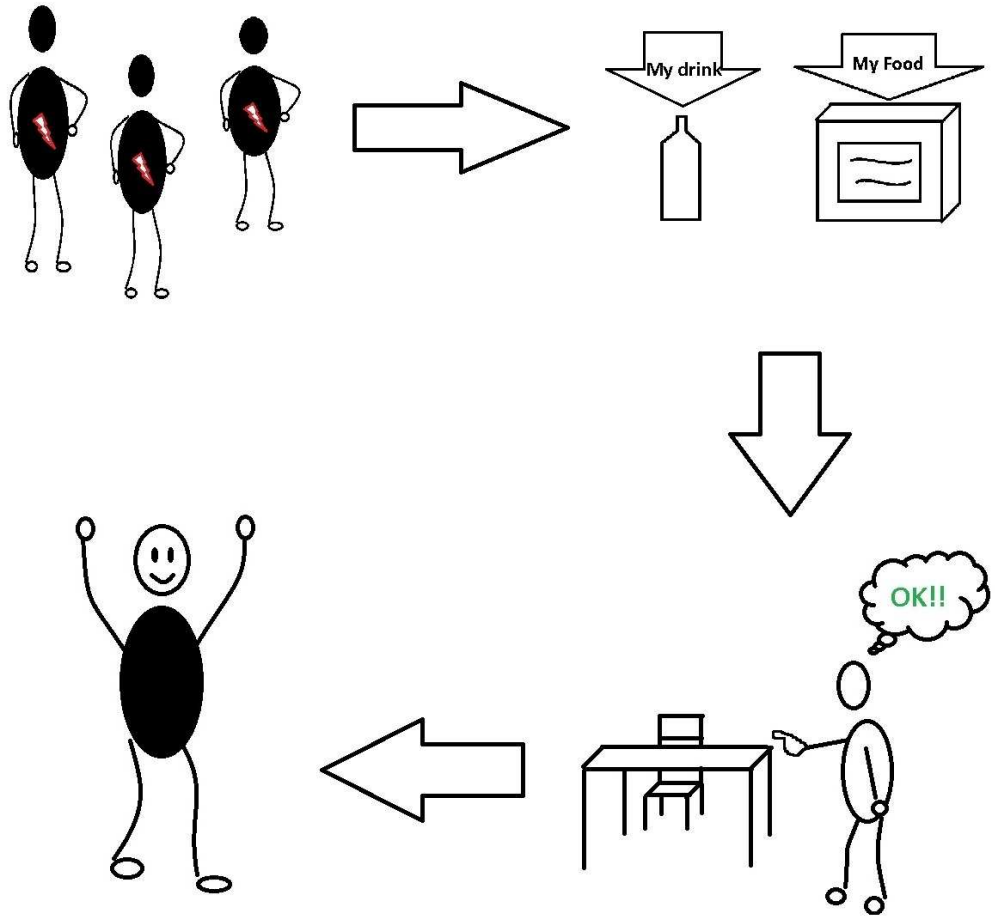


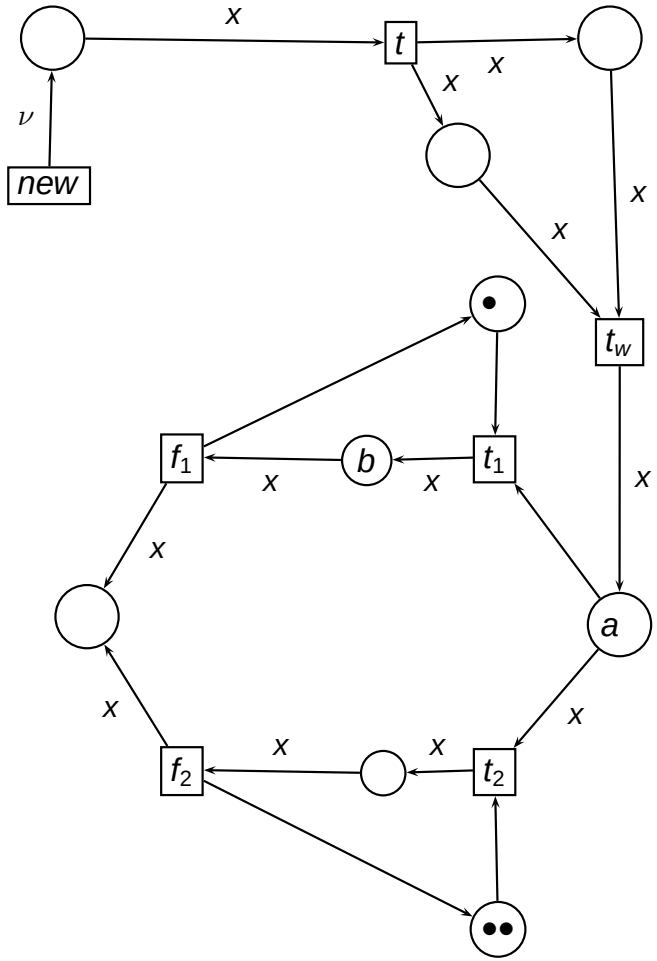
ν-Petri Nets



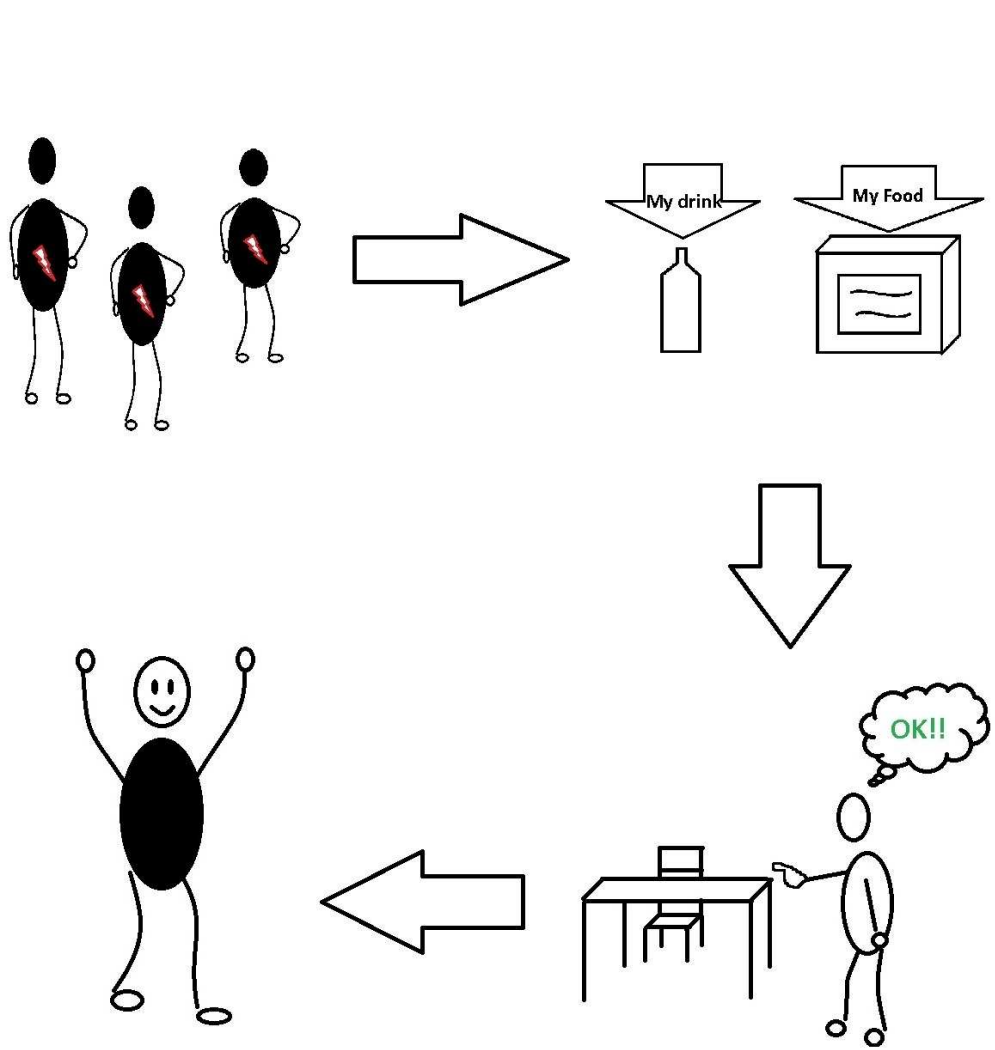


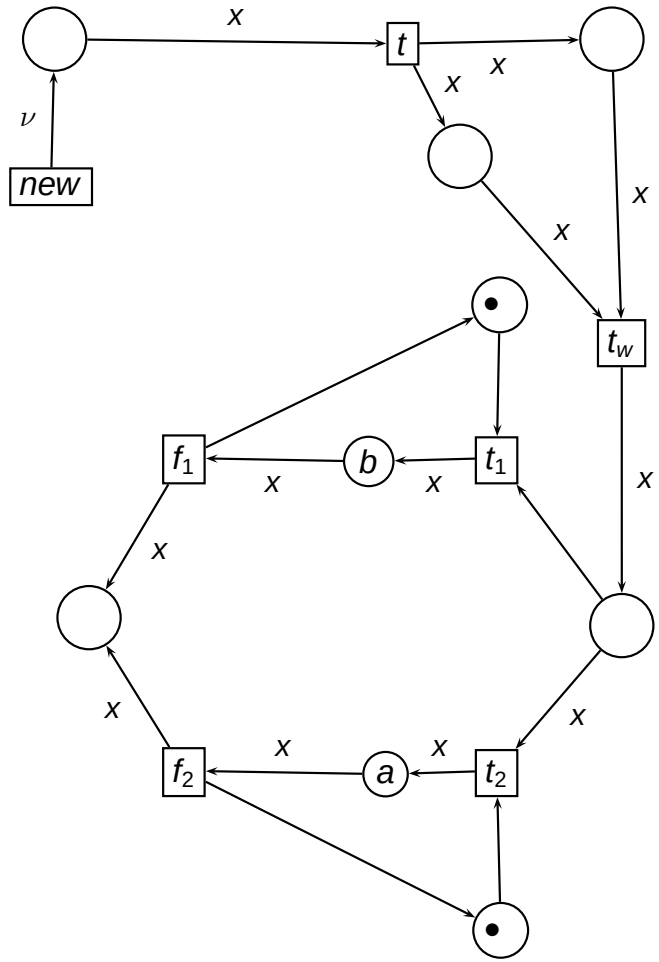
ν-Petri Nets



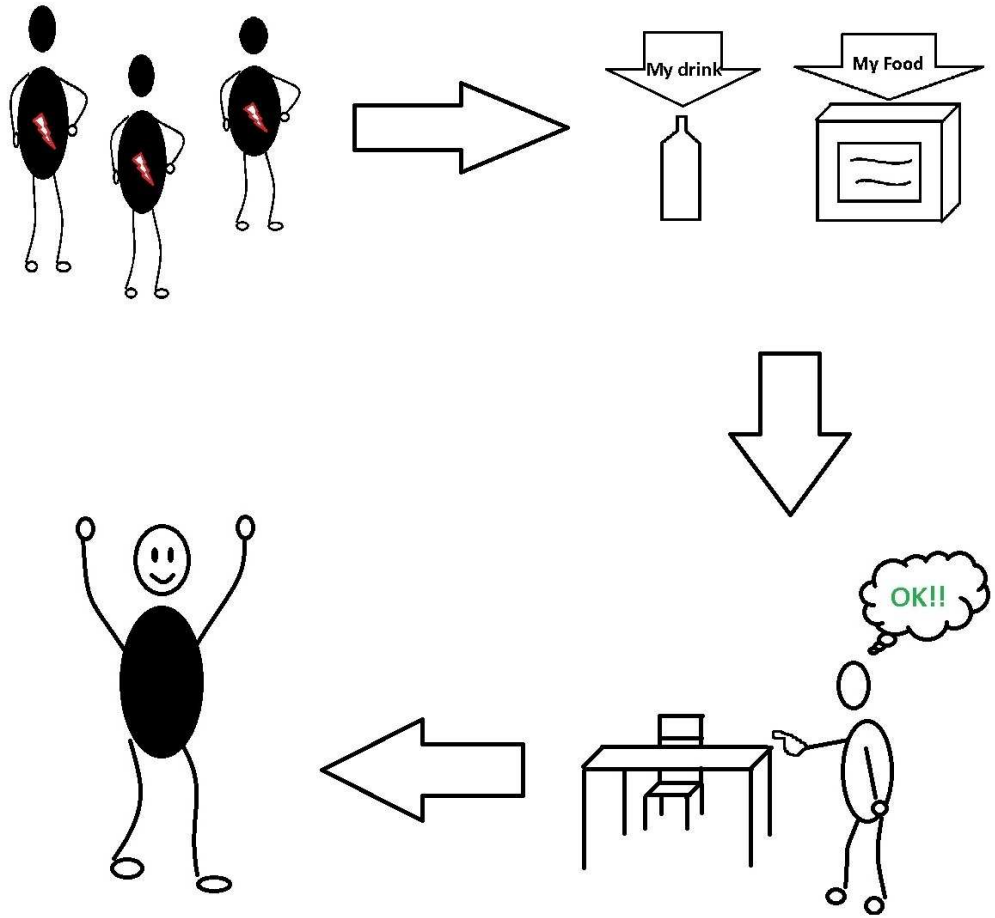


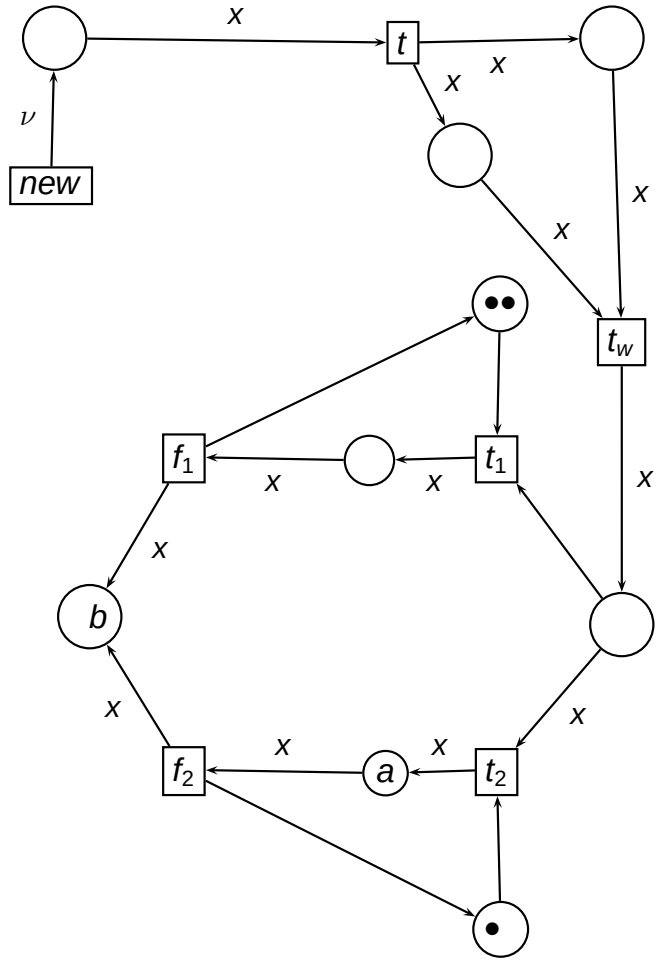
ν-Petri Nets



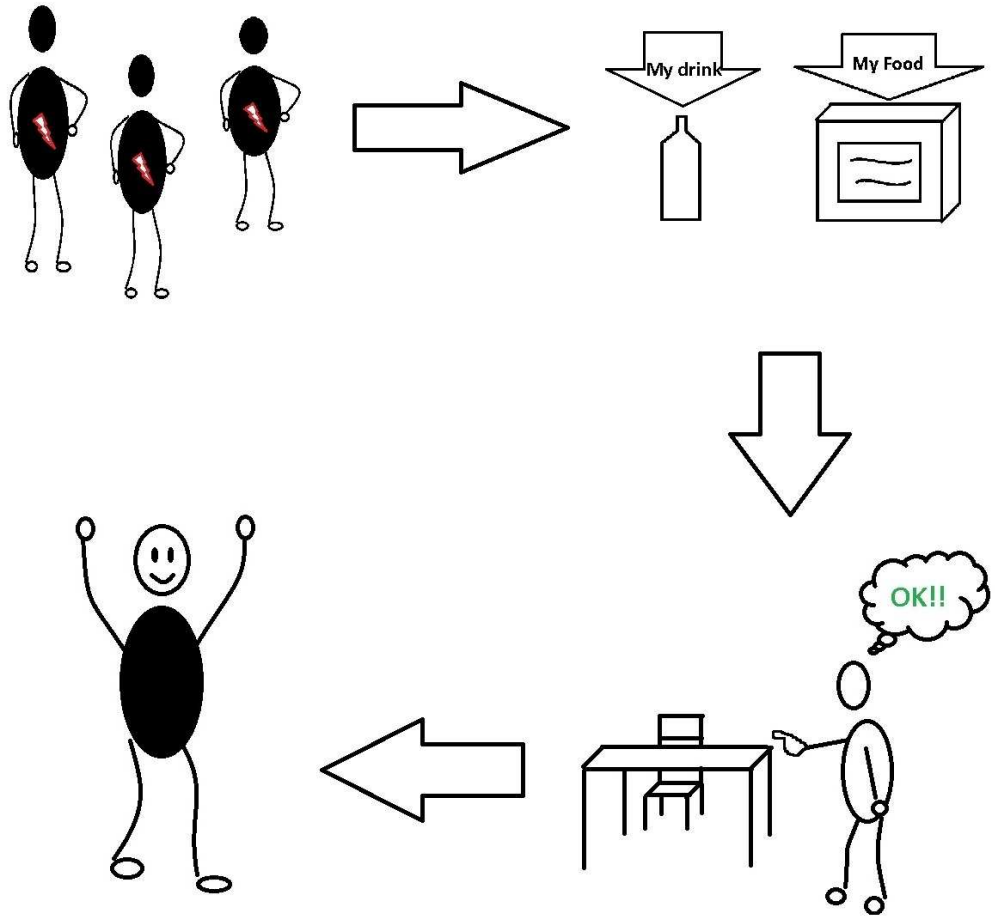


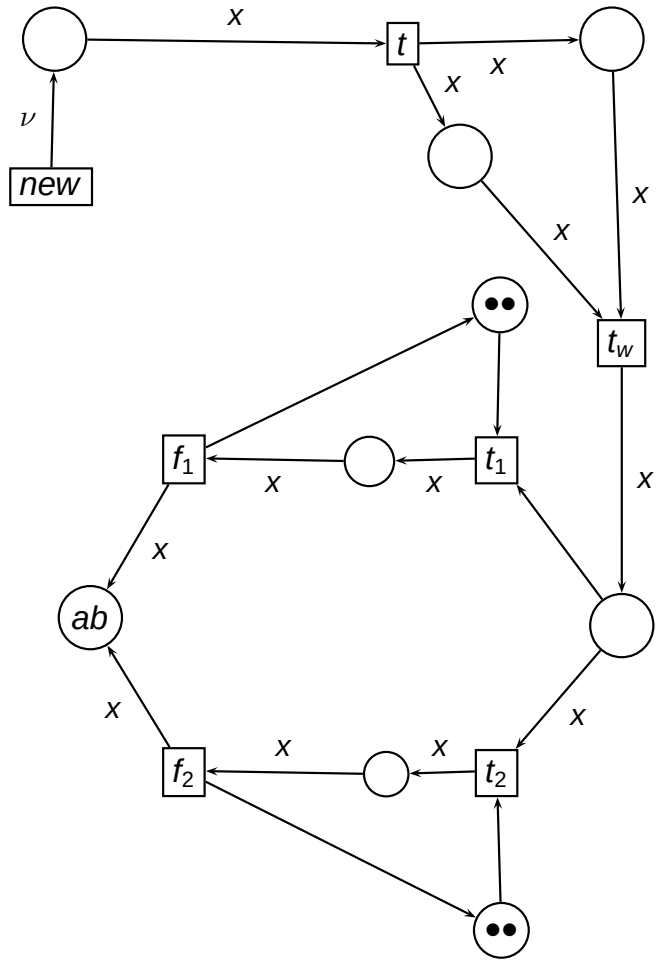
ν-Petri Nets





ν-Petri Nets





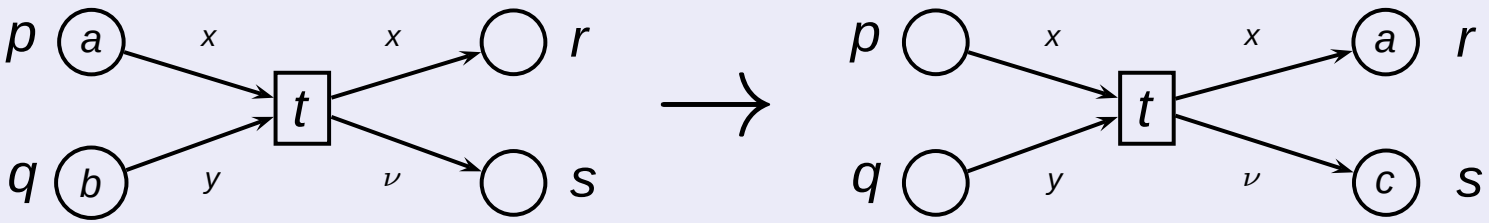
ν-Petri Nets

Definition

A *ν-Petri Net* (ν-*PN*) is a tuple $N = \langle P, T, F, H \rangle$, where:

- P and T are finite disjoint sets,
- $F, H : T \to (P \times Var)^{\oplus}$.

A marking of a ν-*PN* is a multiset over $P \times Id$.



$F(t) = \{(p,x),(q,y)\}$, $H(t) = \{(r,x),(s,\nu)\}$.

Verification

Control-state reachability

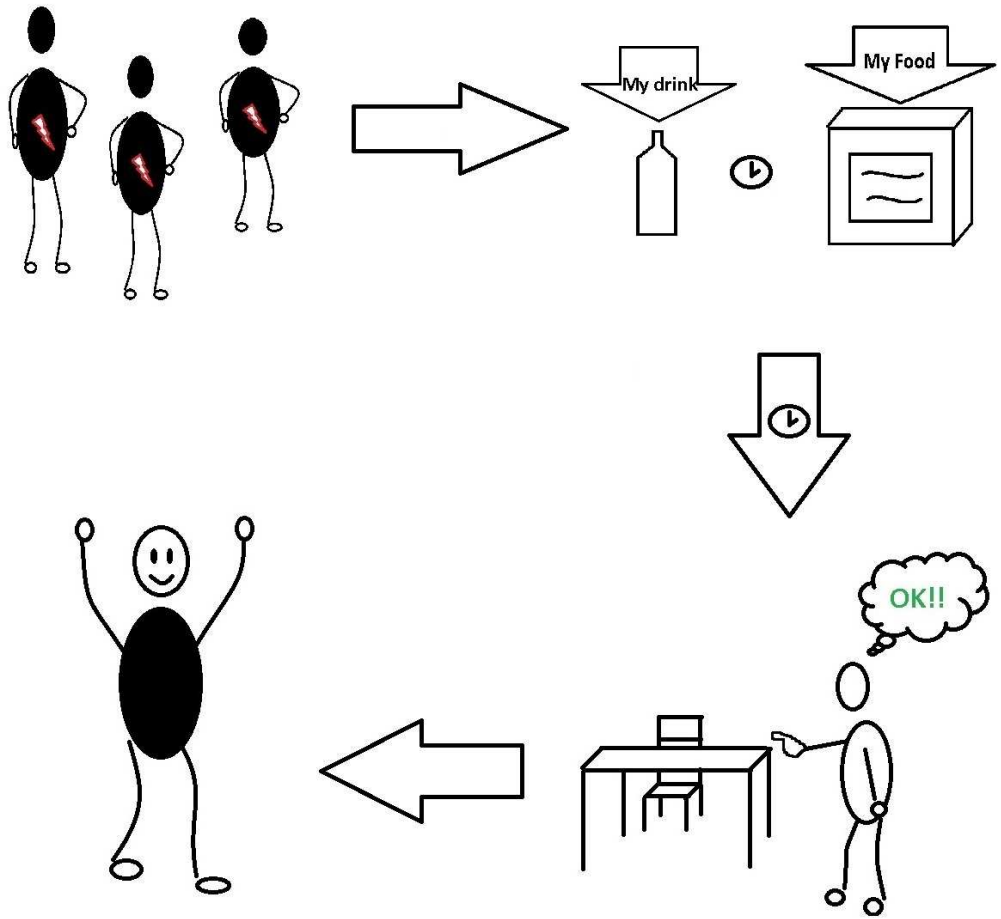
The *control-state reachability* problem is that of deciding, given a place p, whether p is marked in some reachable marking.

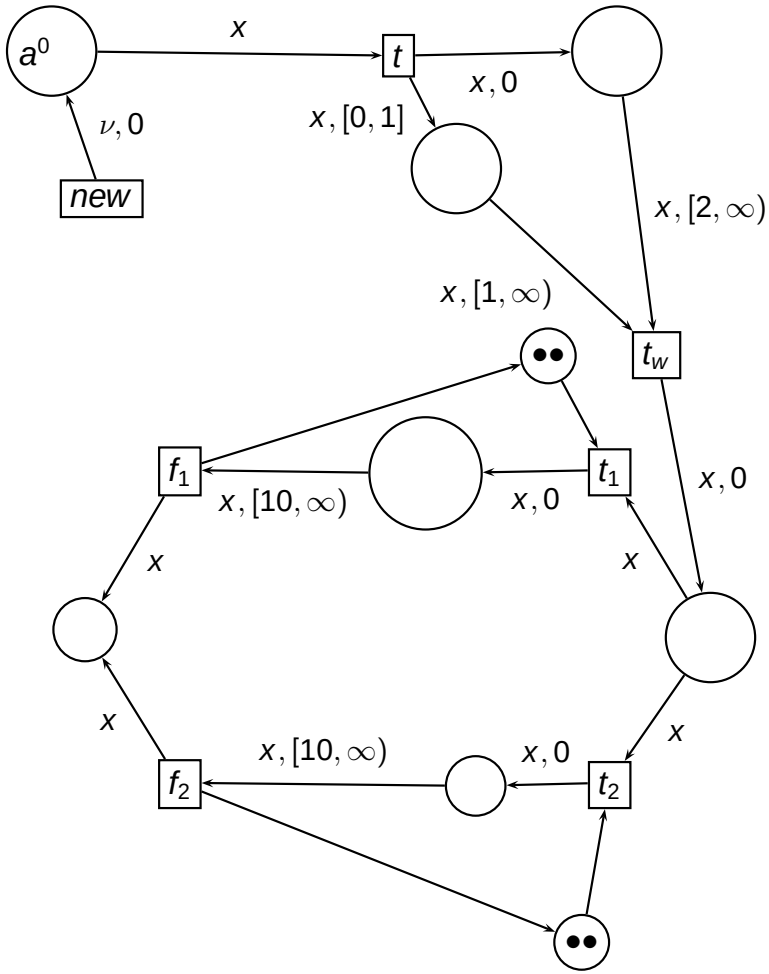
The control-state reachability problem is decidable for:

- Petri nets
- *TdPN*
- ν-*PN*

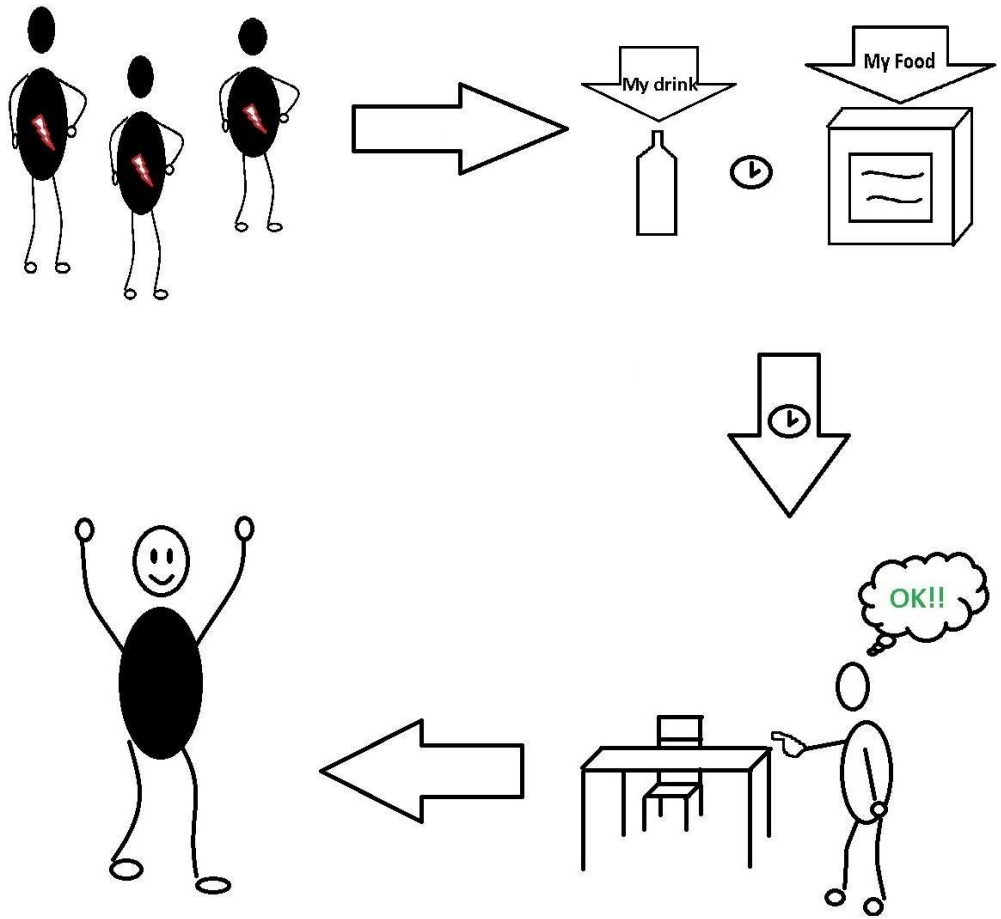
Outline

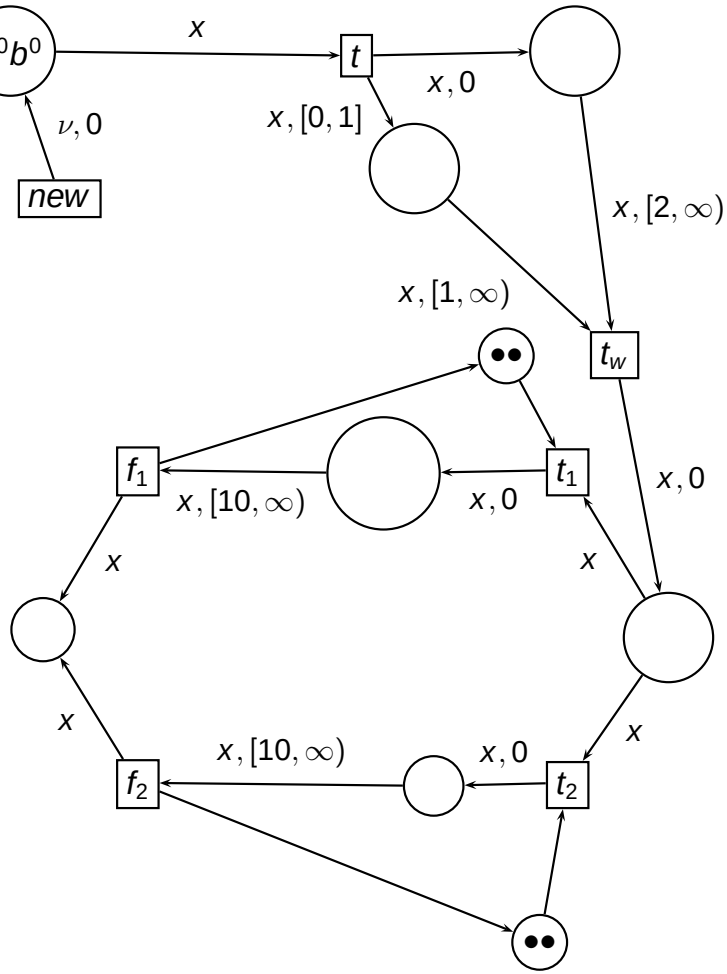
Timed ν-Petri Nets



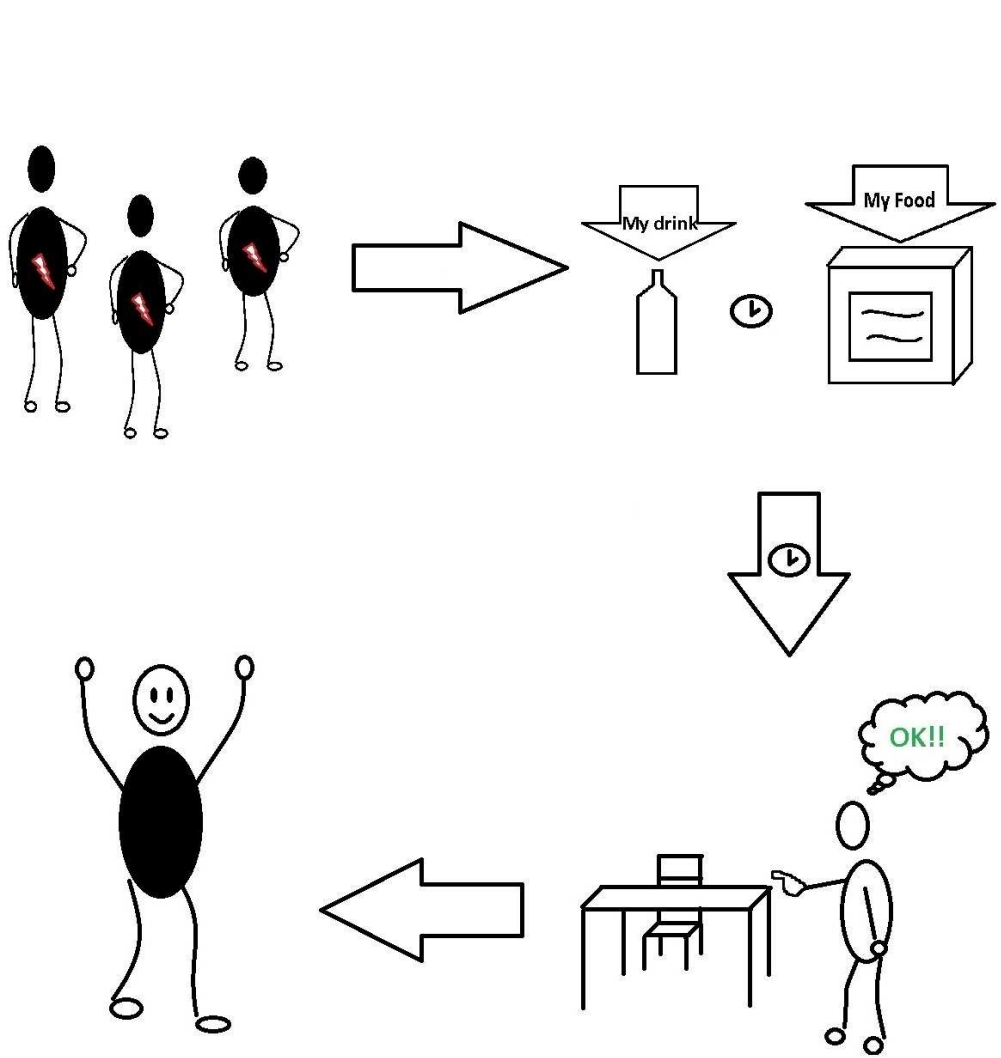


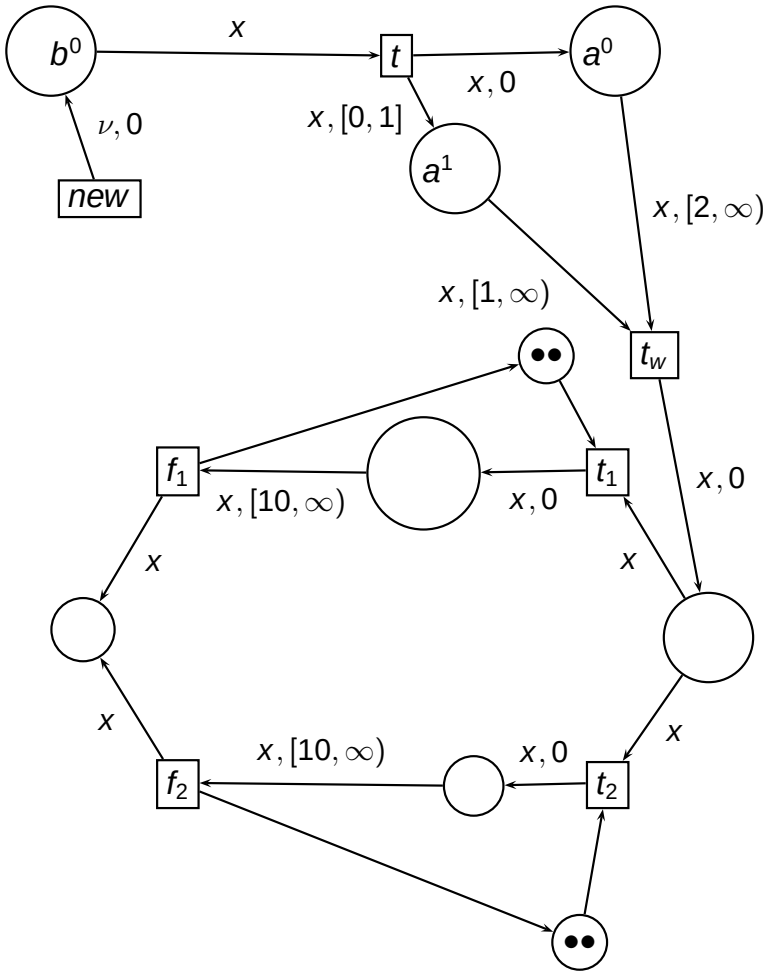
Timed ν-Petri Nets



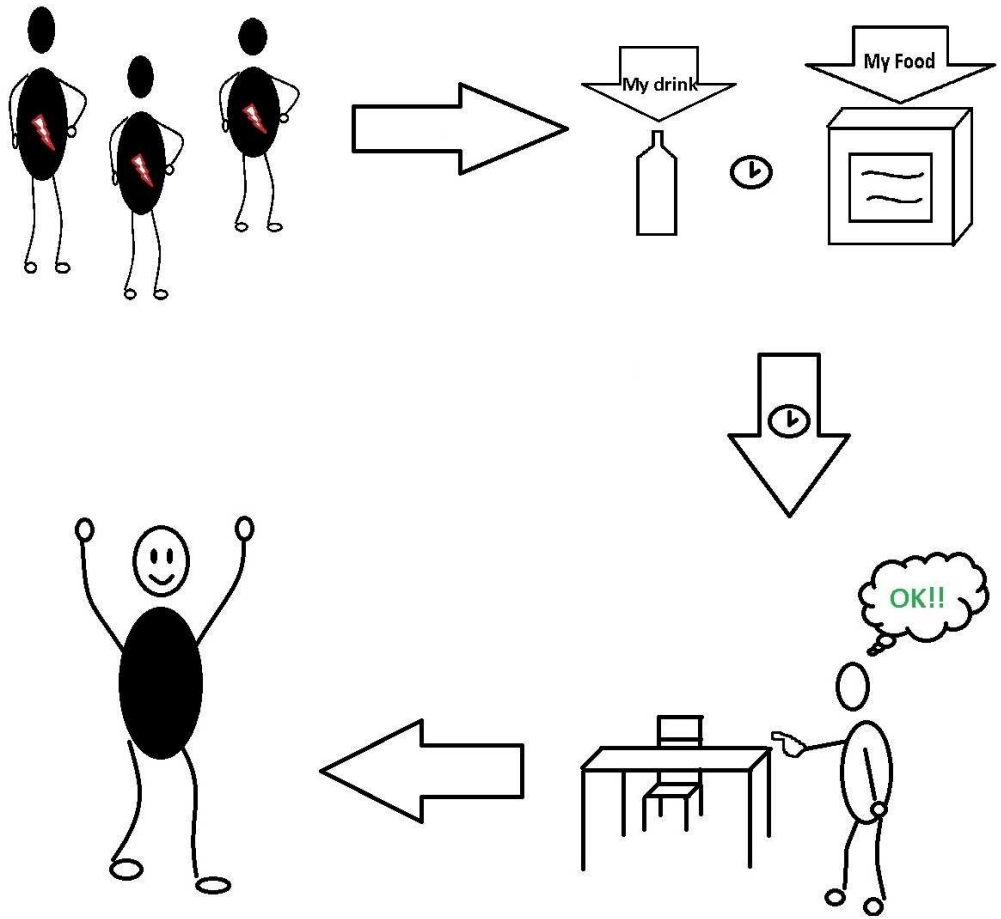


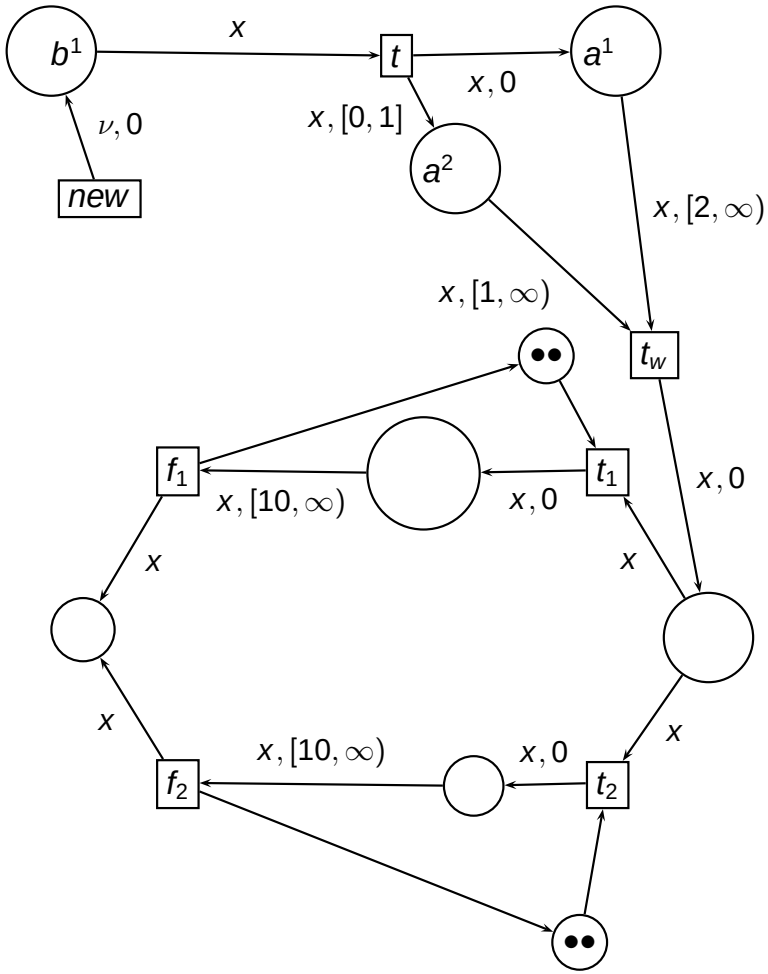
Timed ν-Petri Nets



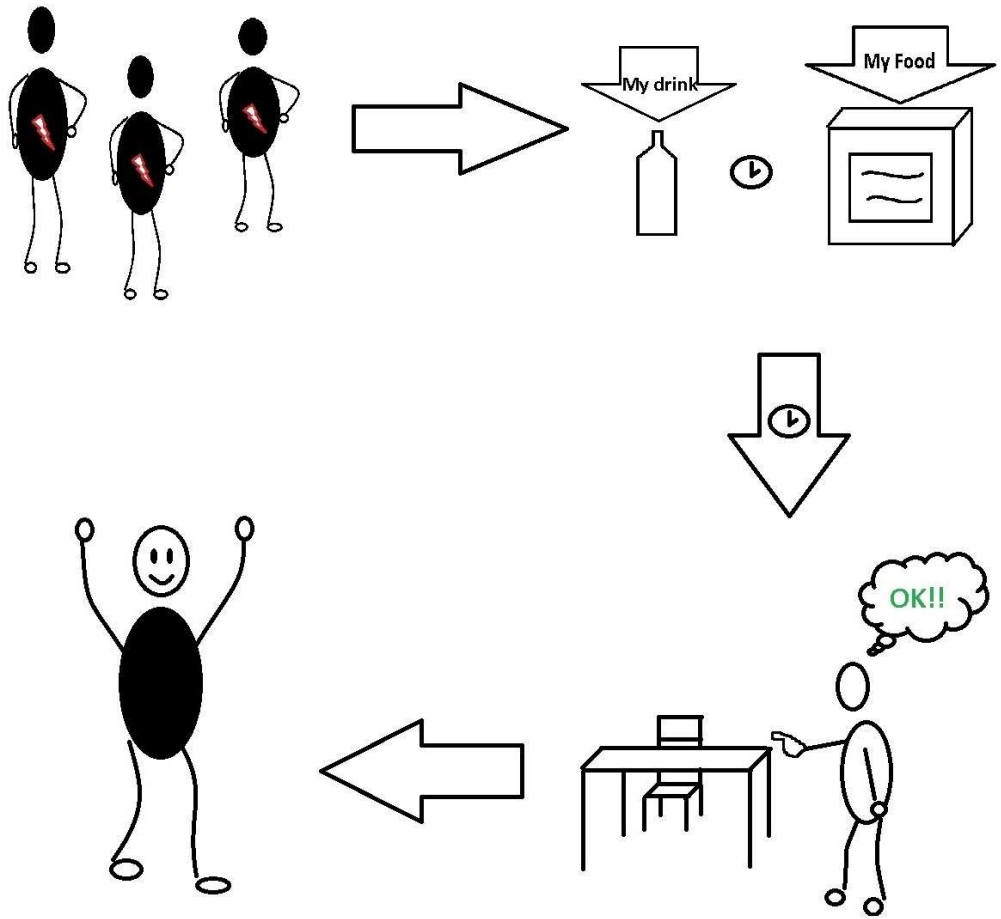


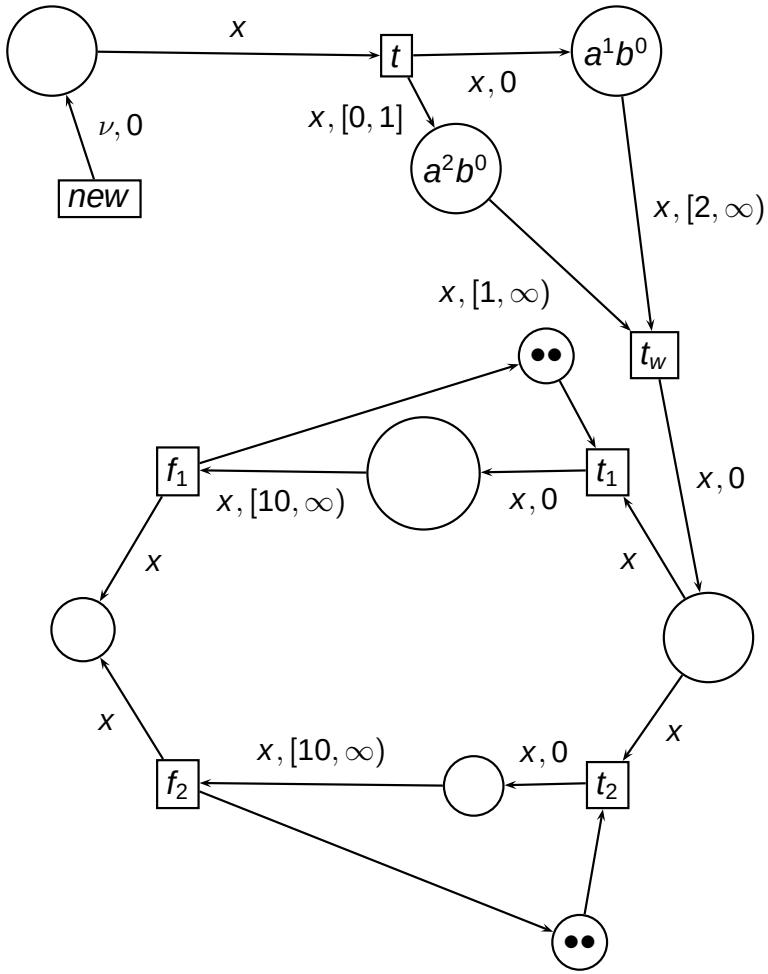
Timed ν-Petri Nets



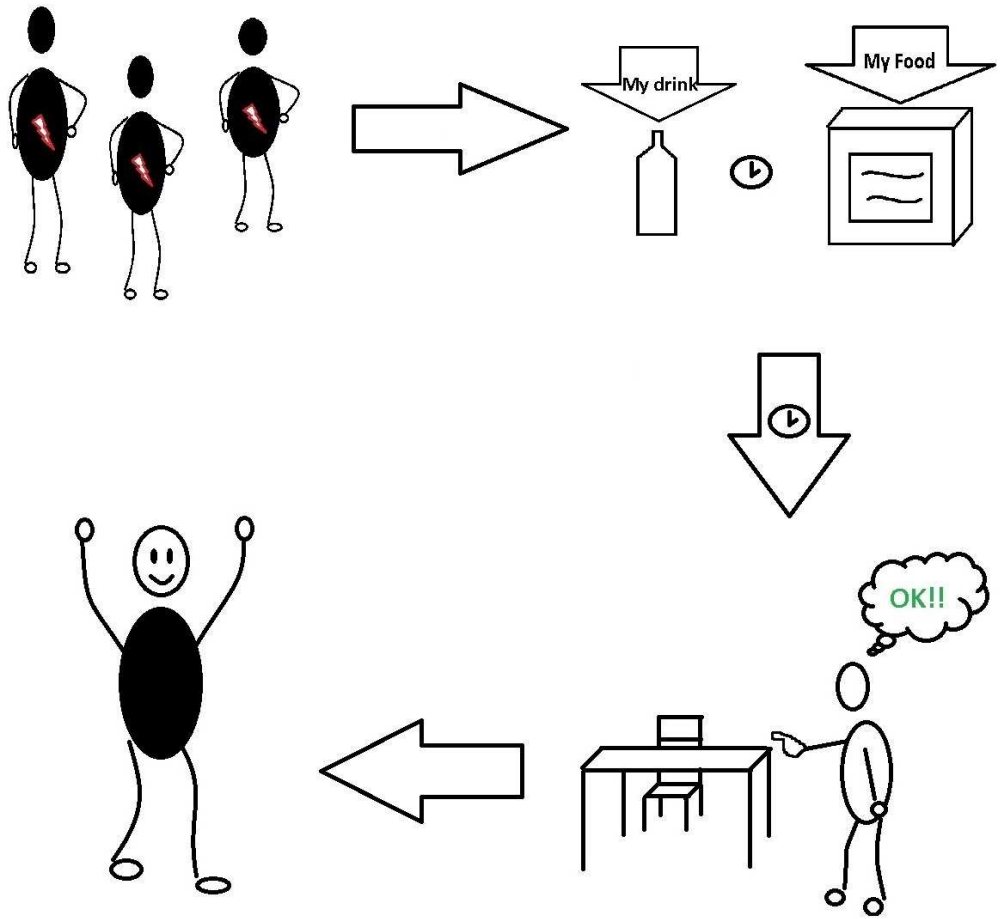


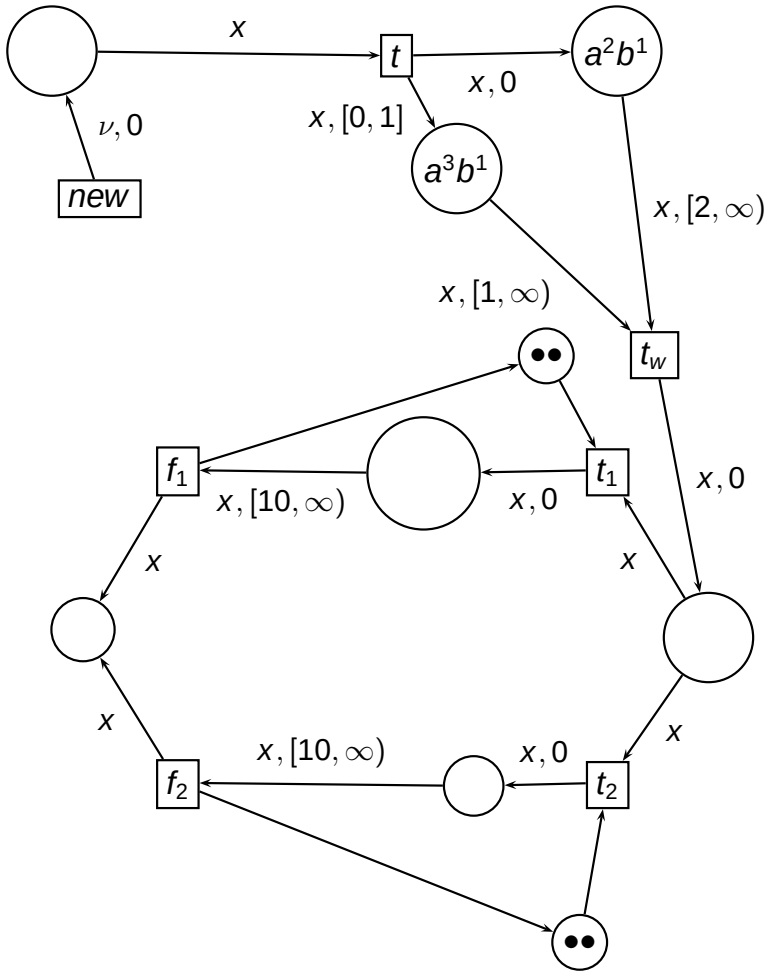
Timed ν-Petri Nets



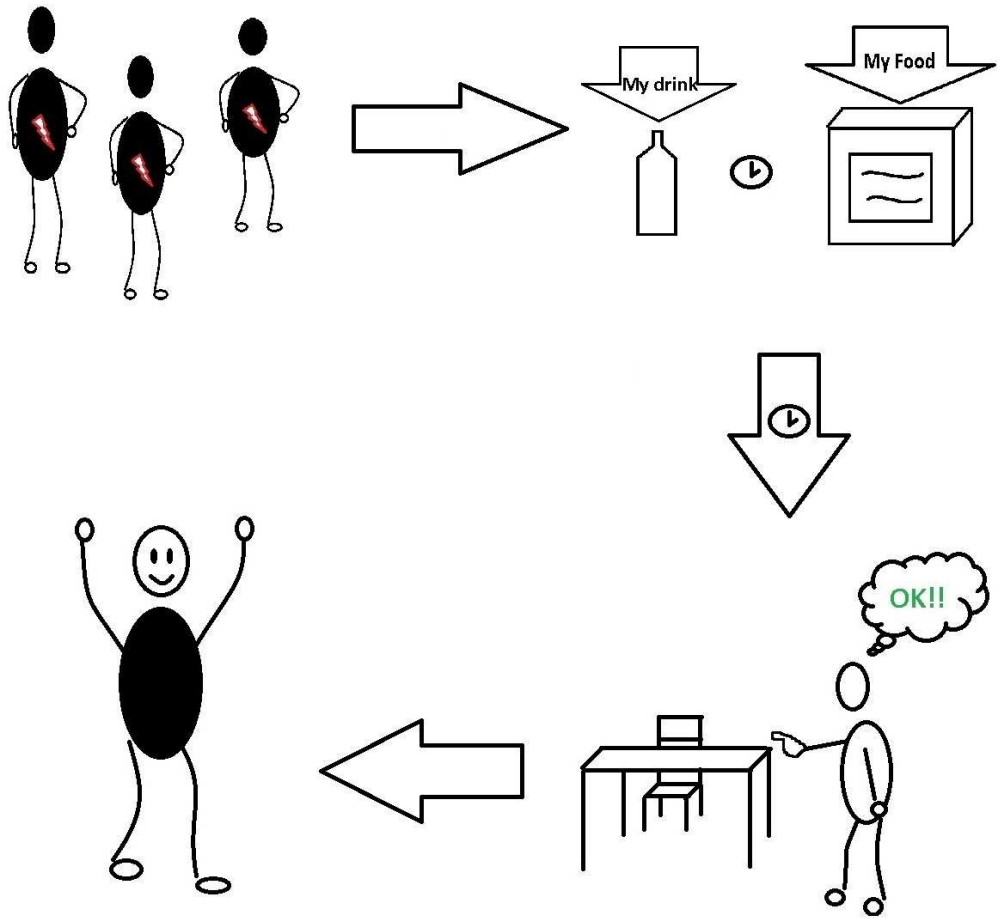


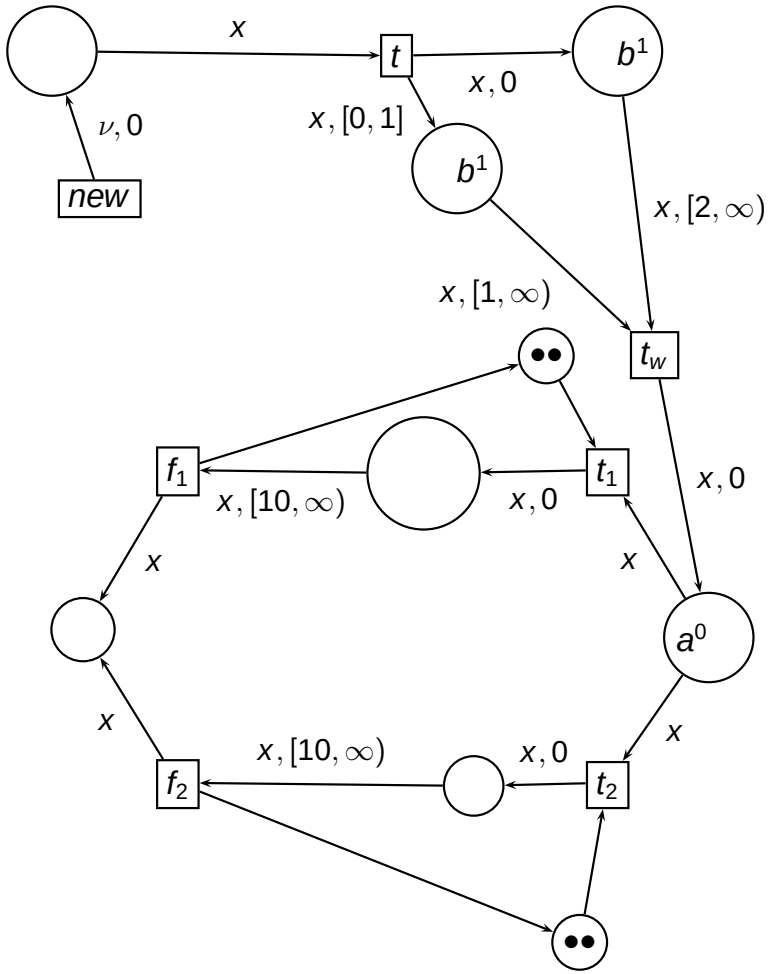
Timed ν-Petri Nets



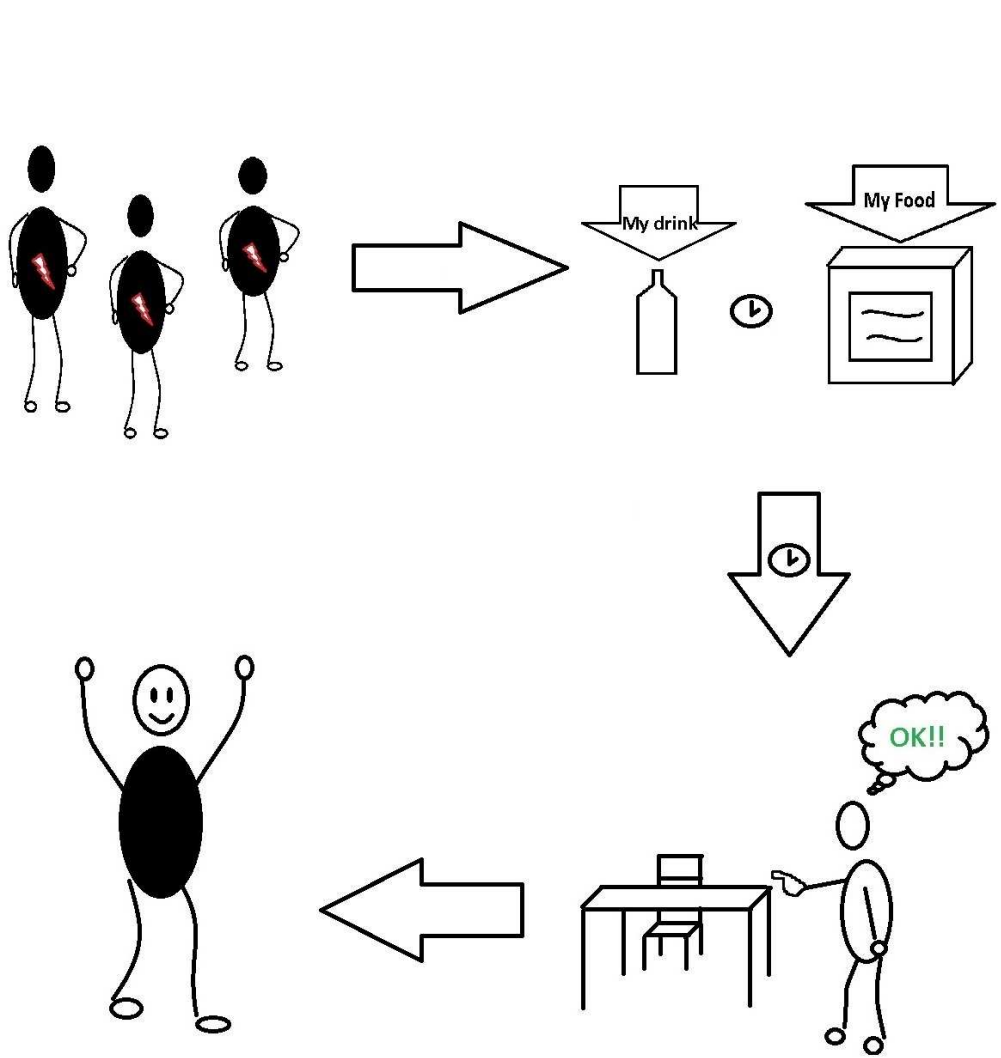


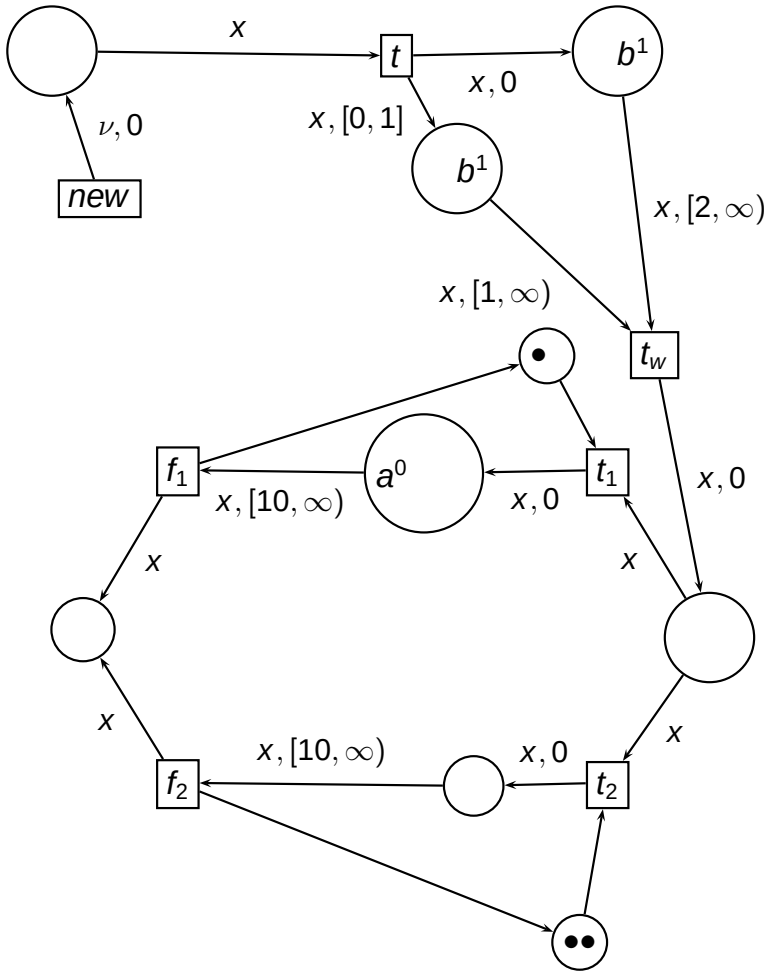
Timed ν-Petri Nets



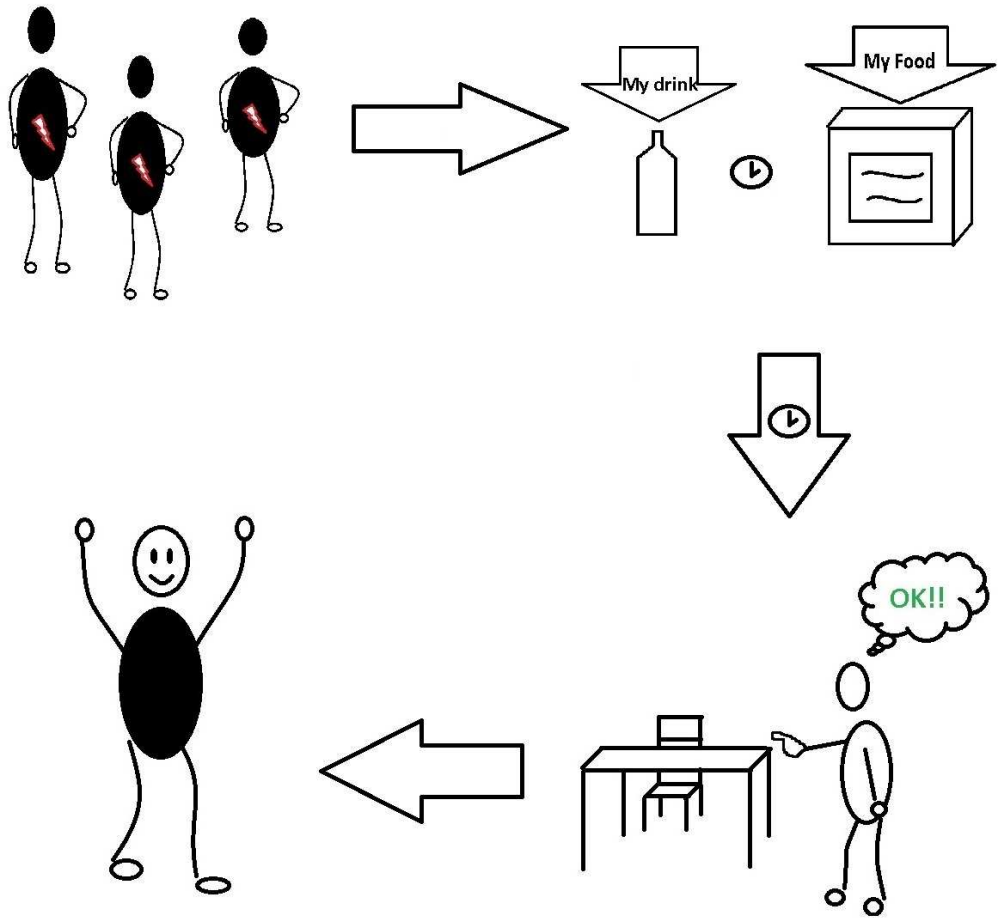


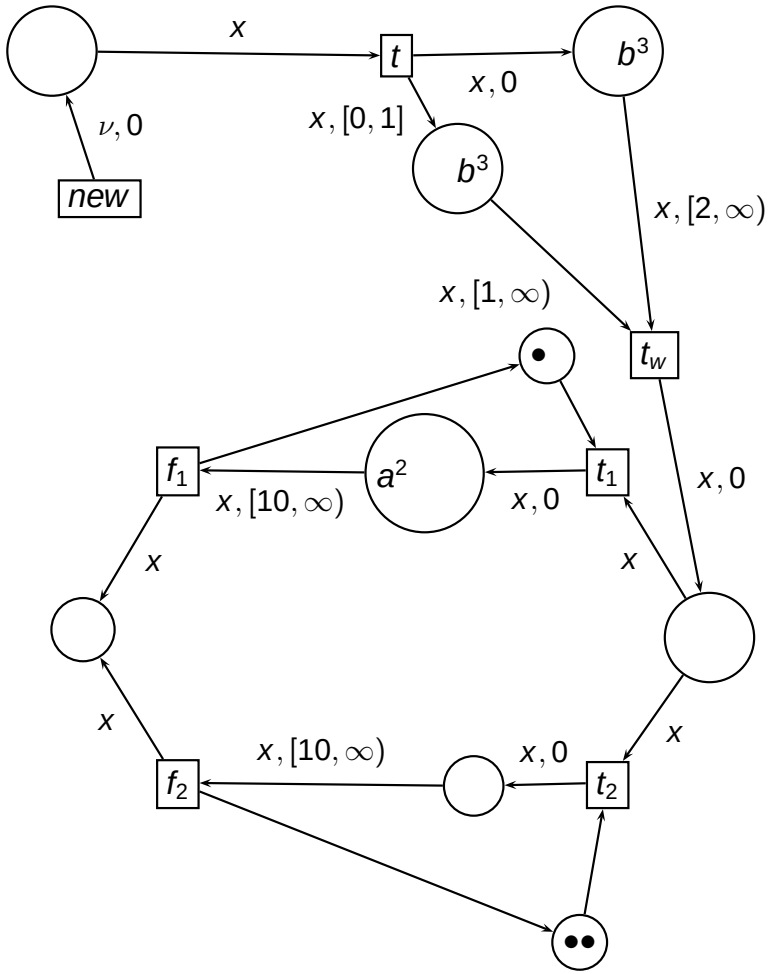
Timed ν-Petri Nets



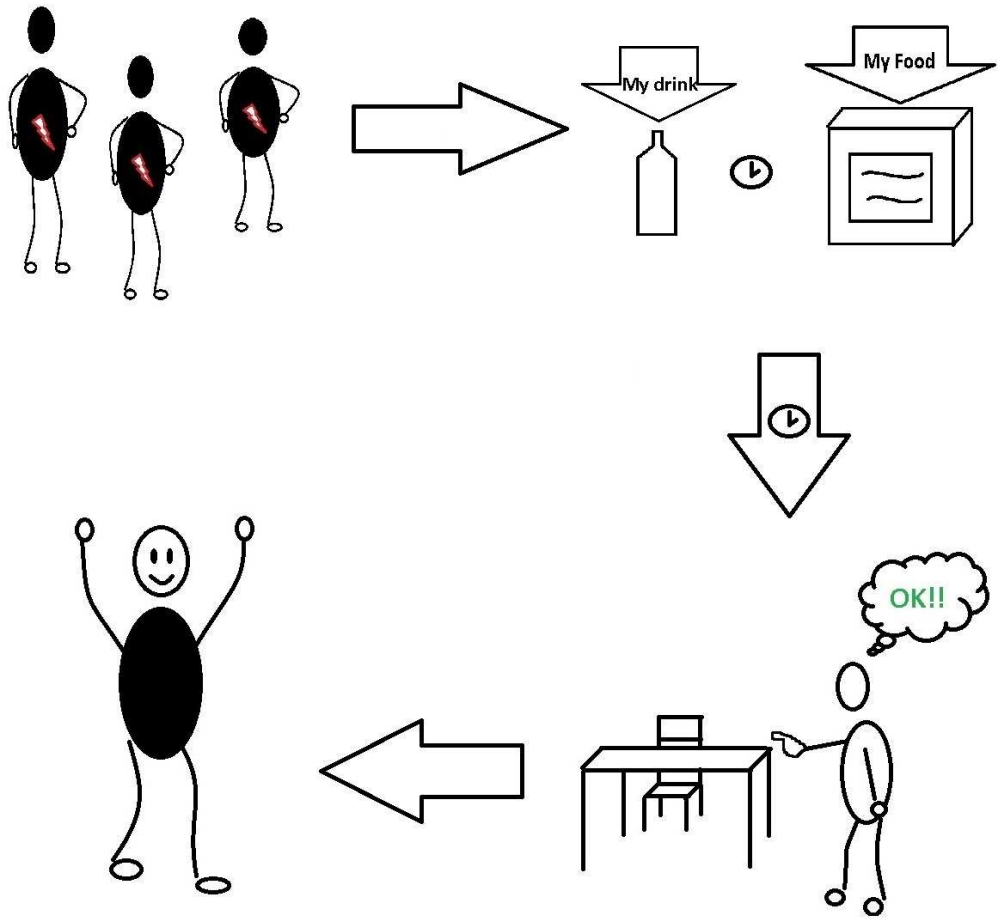


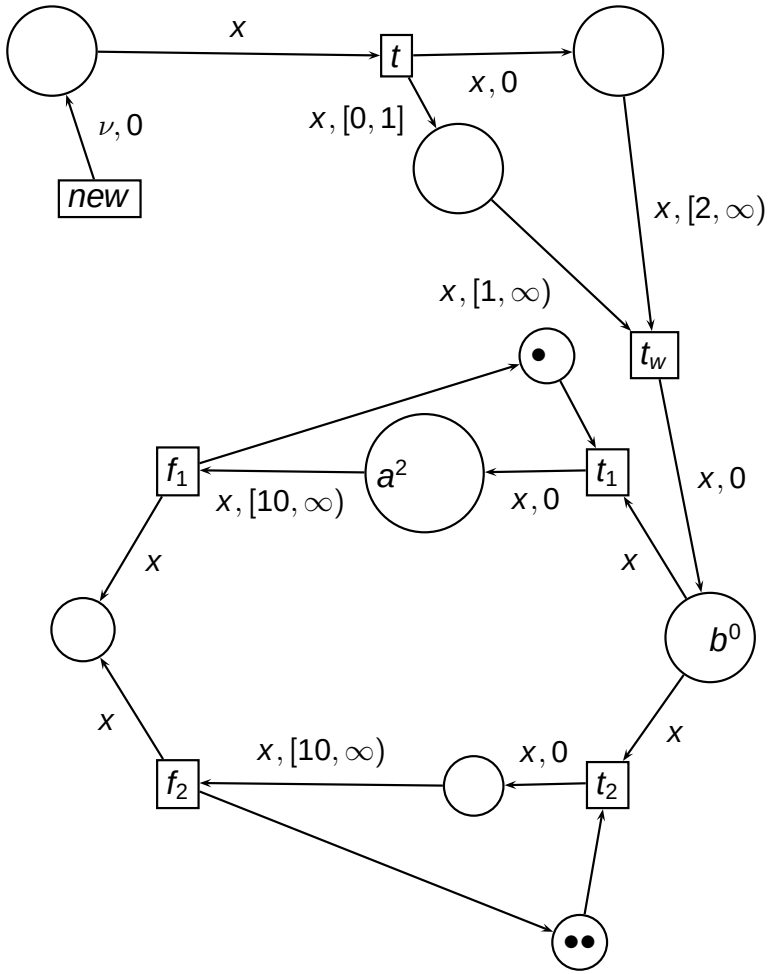
Timed ν-Petri Nets



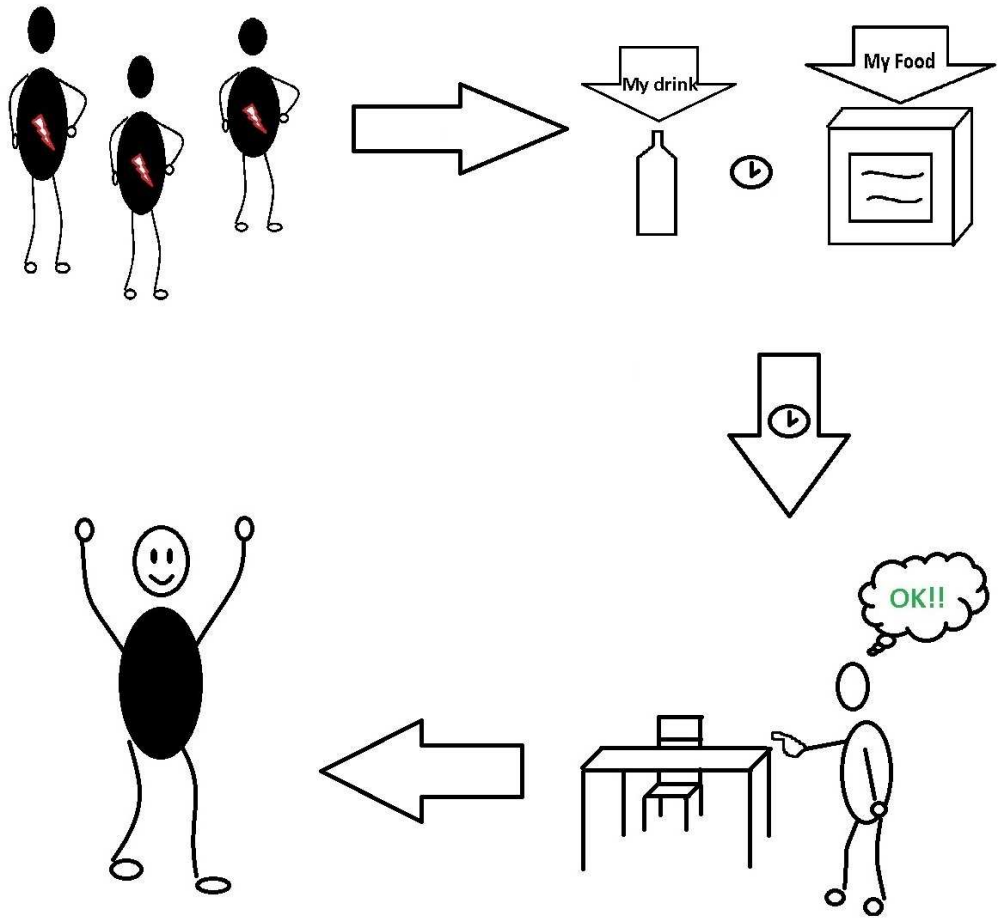


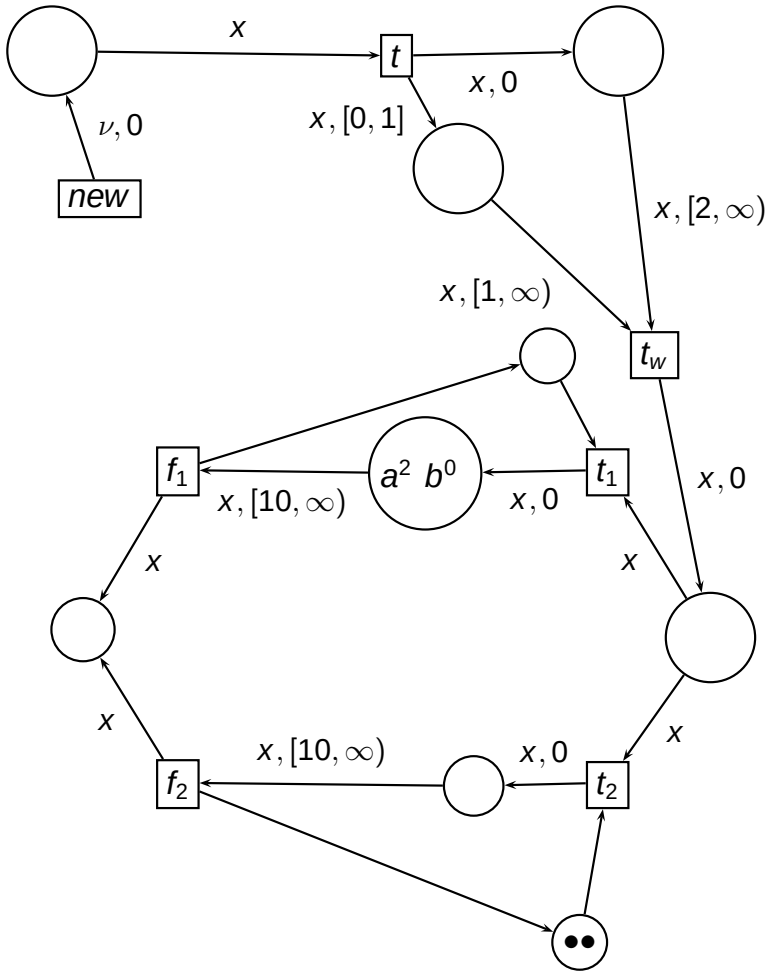
Timed ν-Petri Nets



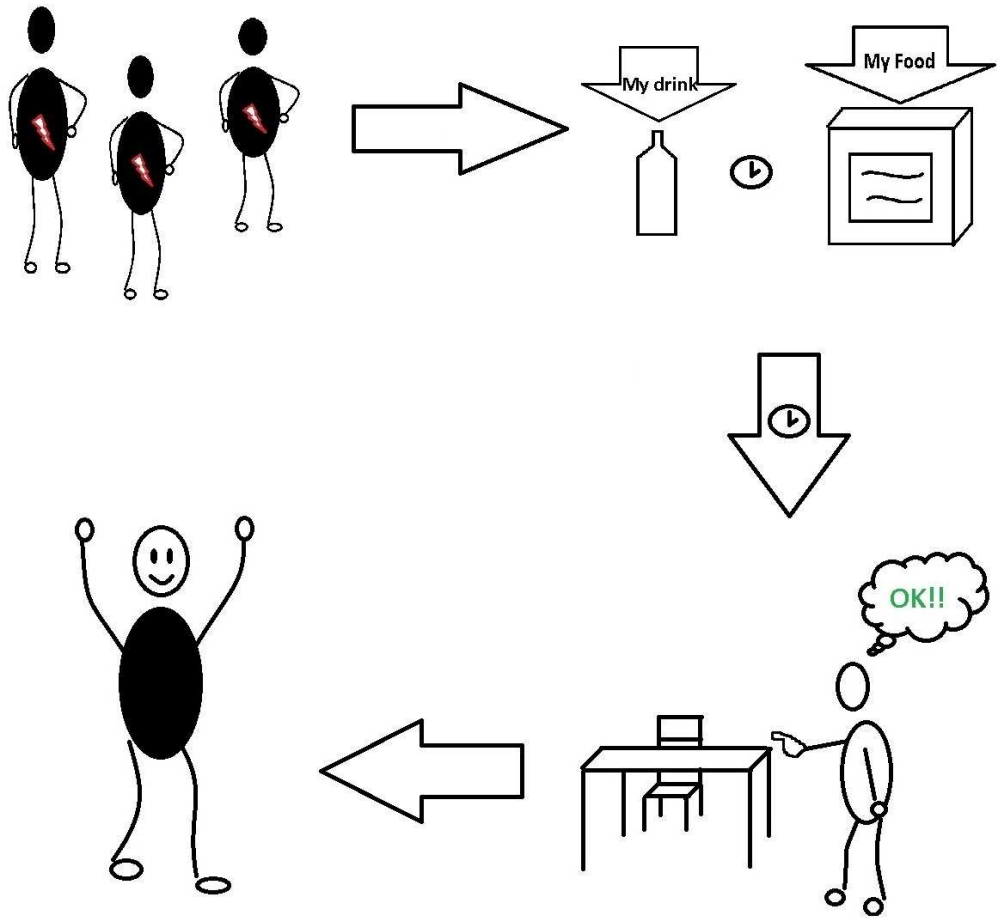


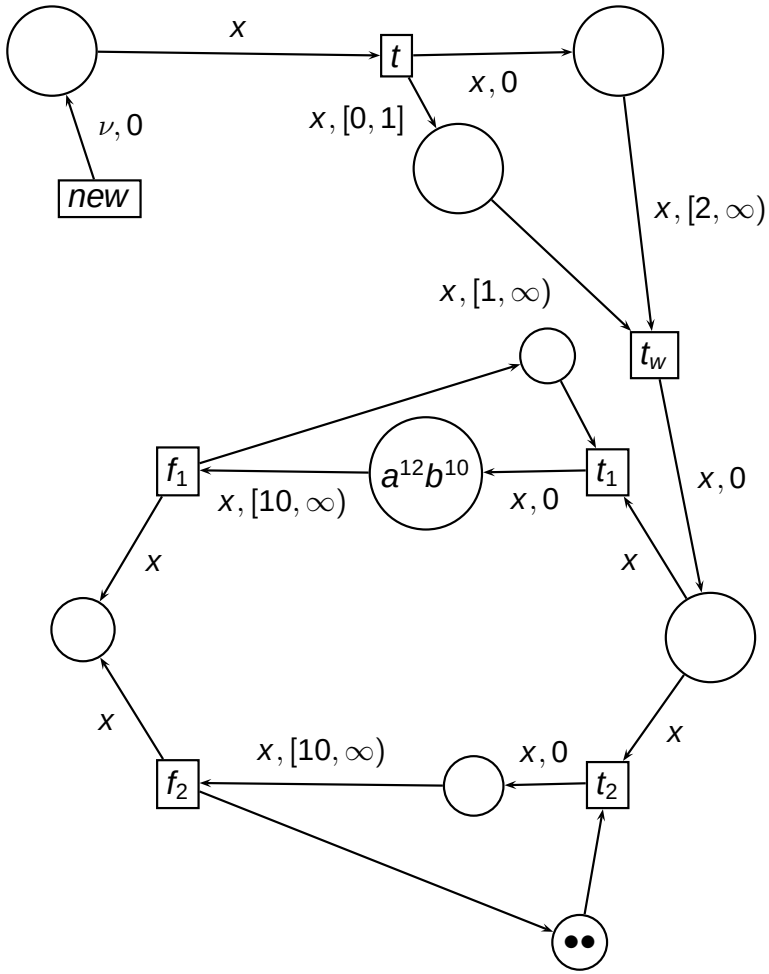
Timed ν-Petri Nets



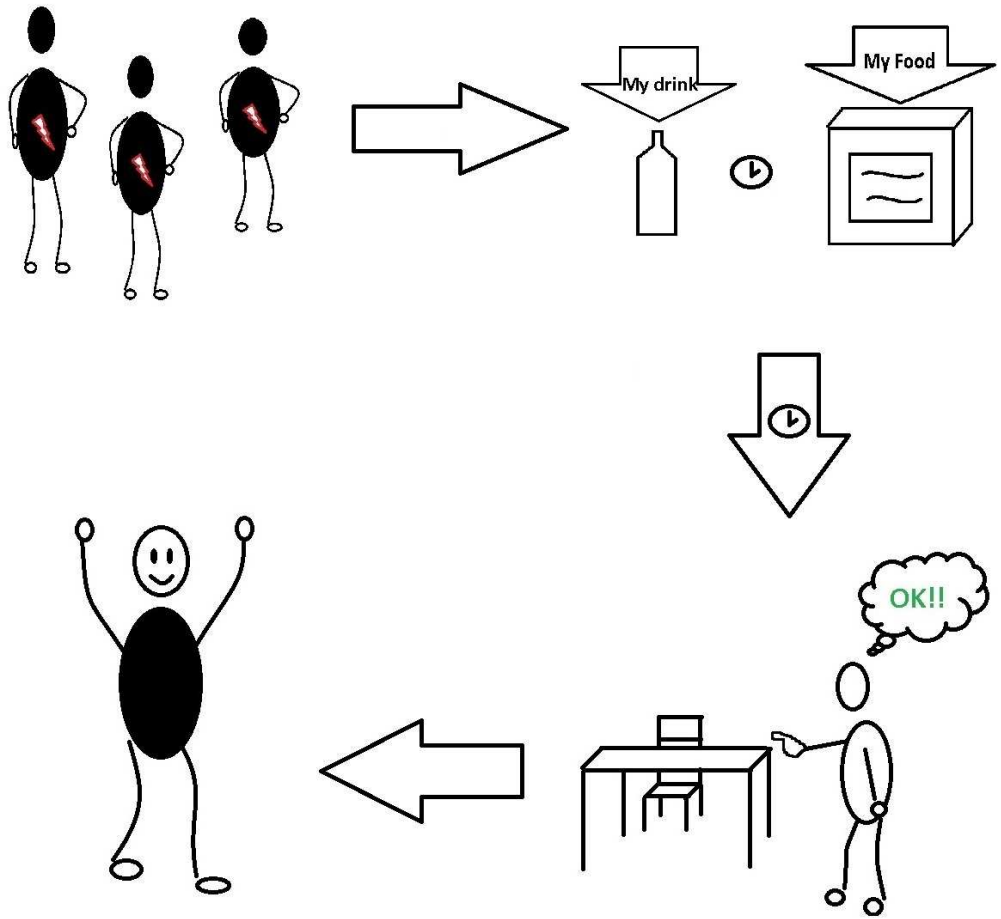


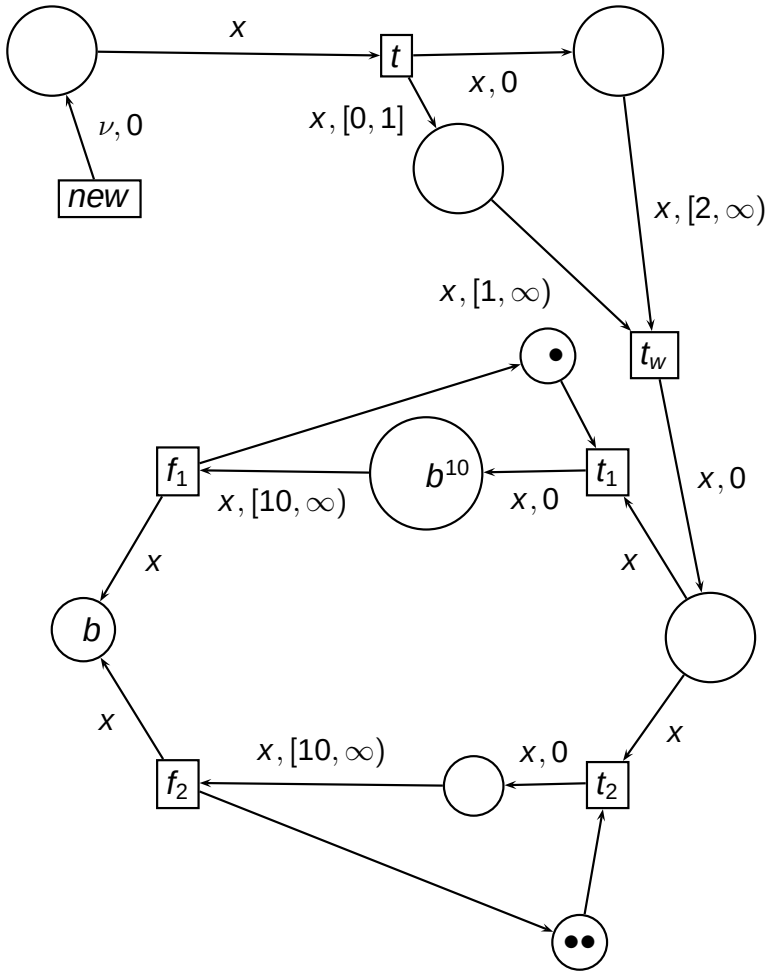
Timed ν-Petri Nets



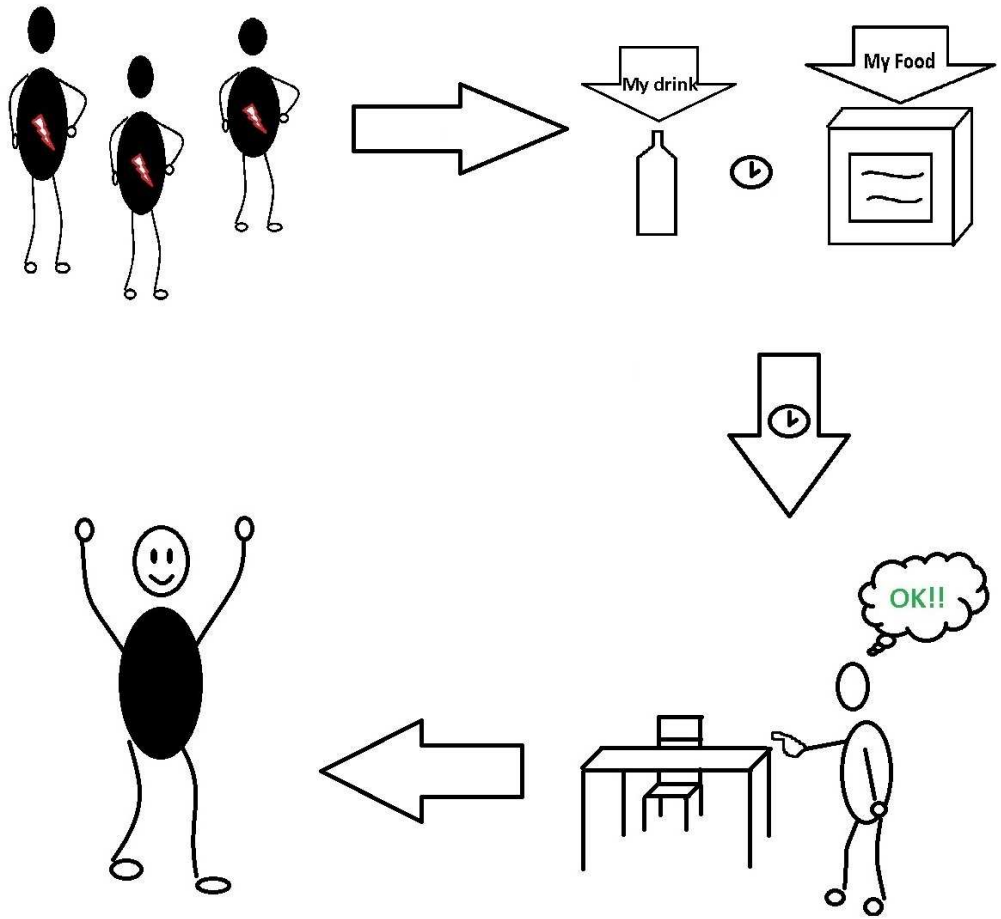


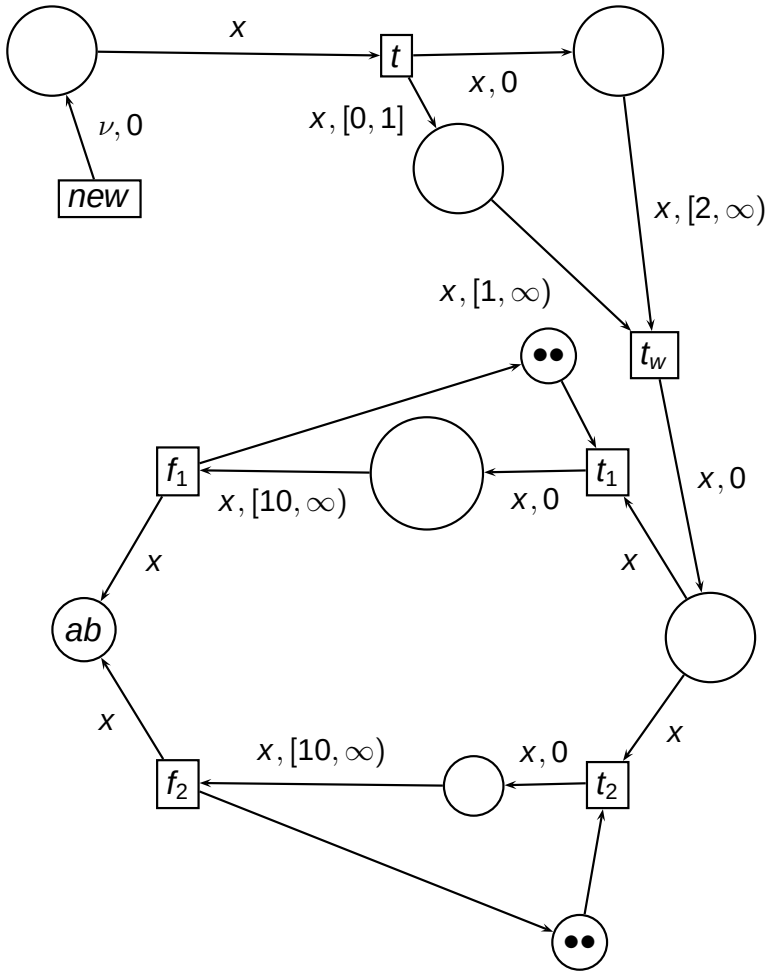
Timed ν-Petri Nets





Timed ν-Petri Nets





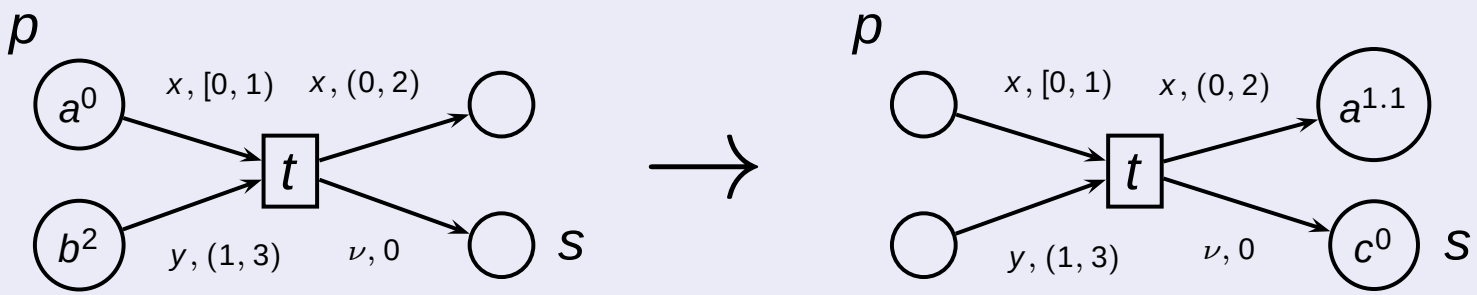
Timed ν-Petri Nets

Definition

A *Timed ν-Petri net* (ν-*TdPN*) is a tuple $N = \langle P, T, F, H \rangle$, where:

- P and T are finite disjoint sets,
- $F : T \to (P \times Var \times \mathcal{I})^{\oplus}$ is the input function,
- $H : T \to (P \times Var \times \mathcal{I})^{\oplus}$ is the output function.

A *token* of a ν-*TdPN* is an element of $P \times Id \times \mathbb{R}_{\geq 0}$. A *marking* is a finite multiset of tokens.



$F(t) = \{(p, x, [0, 1)), (q, y, (1, 3))\}$, $H(t) = \{(r, x, (0, 2)), (s, \nu, [0, 0])\}$.

Turing-completeness of Timed ν-Petri nets

- Timed ν-Petri nets are Turing complete
  - Control-state reachability is undecidable
- They simulate a Turing-complete formalism
  - ν-PN + replication

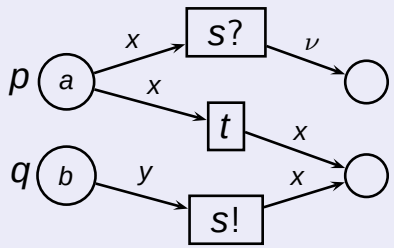
Turing-completeness of Timed ν-Petri nets

- Timed ν-Petri nets are Turing complete
  - Control-state reachability is undecidable
- They simulate a Turing-complete formalism
  - ν-PN + replication

Turing-completeness of Timed ν-Petri nets

ν-RN Systems

Collection of ν-*PN* that can
synchronize with each other, and that can create replicas of themselves.

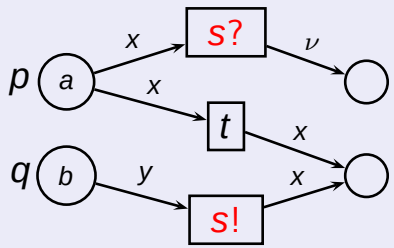


The control-state reachability problem for ν-*RN* is that of deciding whether some reachable marking marks a given place, and it is undecidable.

Turing-completeness of Timed ν-Petri nets

ν-RN Systems

Collection of ν-*PN* that can
synchronize with each other, and that can create replicas of themselves.

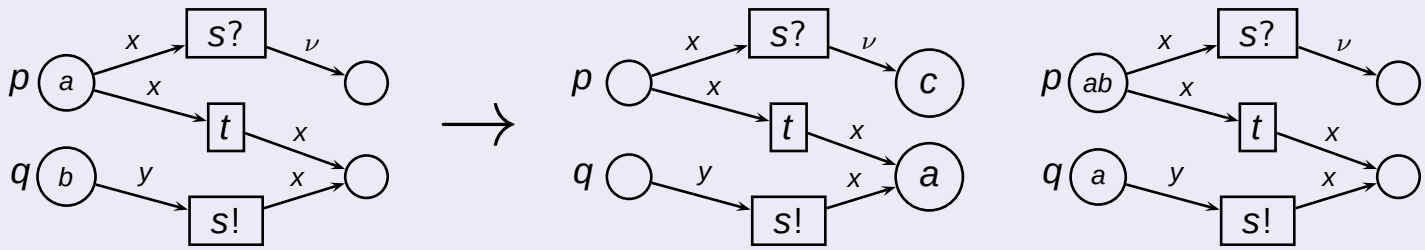


The control-state reachability problem for ν-*RN* is that of deciding whether some reachable marking marks a given place, and it is undecidable.

Turing-completeness of Timed ν-Petri nets

ν-RN Systems

Collection of ν-*PN* that can synchronize with each other, and that can create replicas of themselves.



The control-state reachability problem for ν-*RN* is that of deciding whether some reachable marking marks a given place, and it is undecidable.

Simulation of ν-*RN* systems

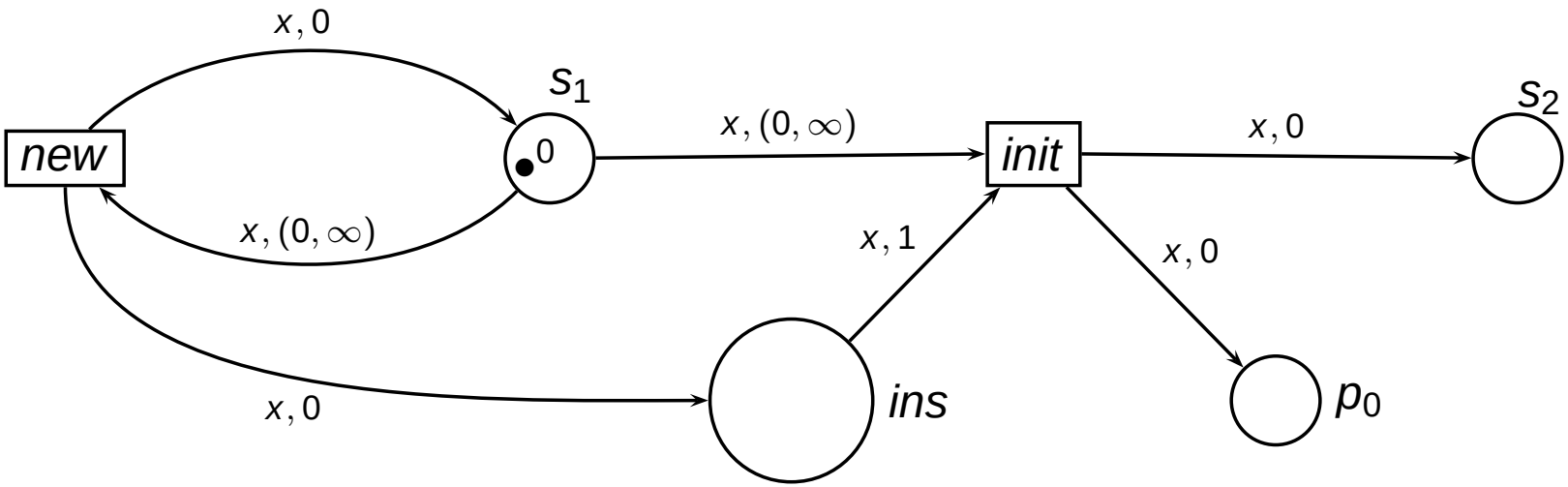
We represent each instance of the ν-*RN* system by a multiset of tokens with the **same clock value**.

$$\begin{matrix}(p,a),(q,a),(p,b)\\(p,a),(q,c)\end{matrix} \rightsquigarrow (p,a,0),(q,a,0),(p,b,0),(p,a,0.5),(q,c,0.5)$$

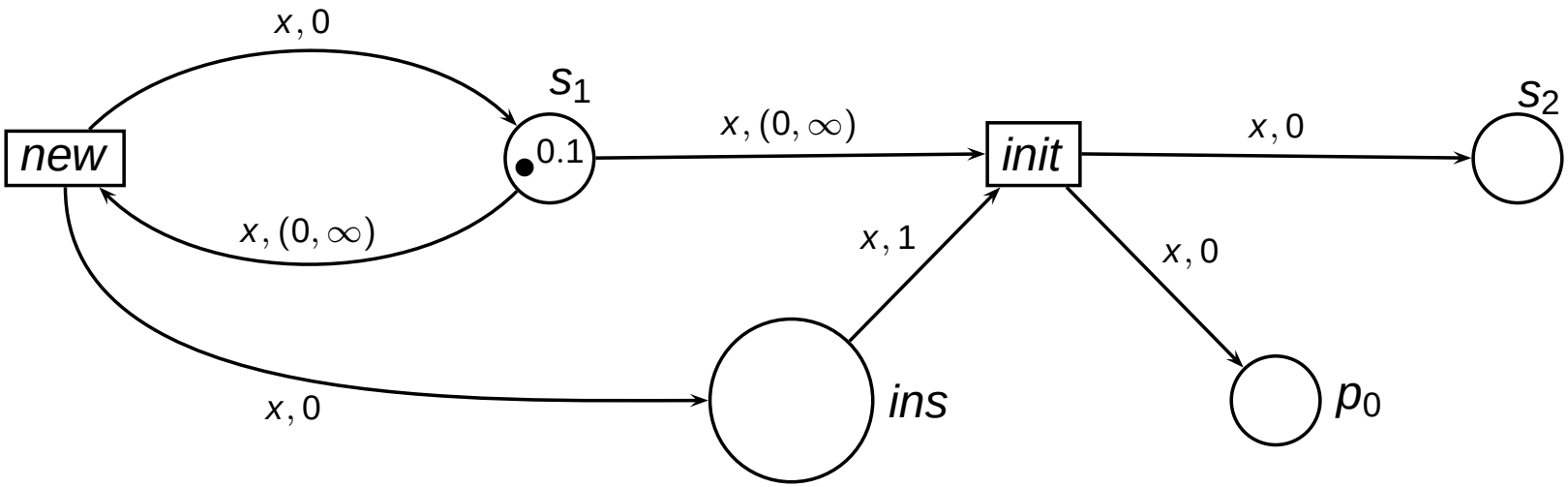
We perform the simulation in two steps:

- Step1: Creation of instances.
- Step2: Simulation.
  - Simulation of the firing of autonomous transitions.
  - Simulation of the firing of synchronizing transitions.

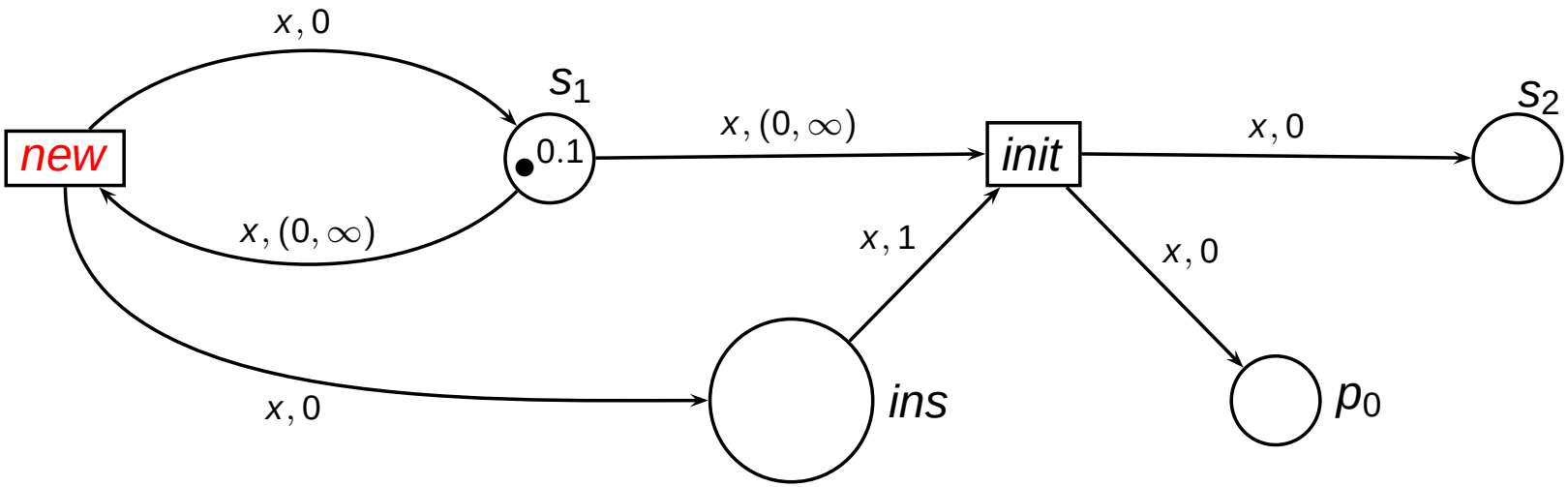
Undecidability-Step1: Creation of Instances



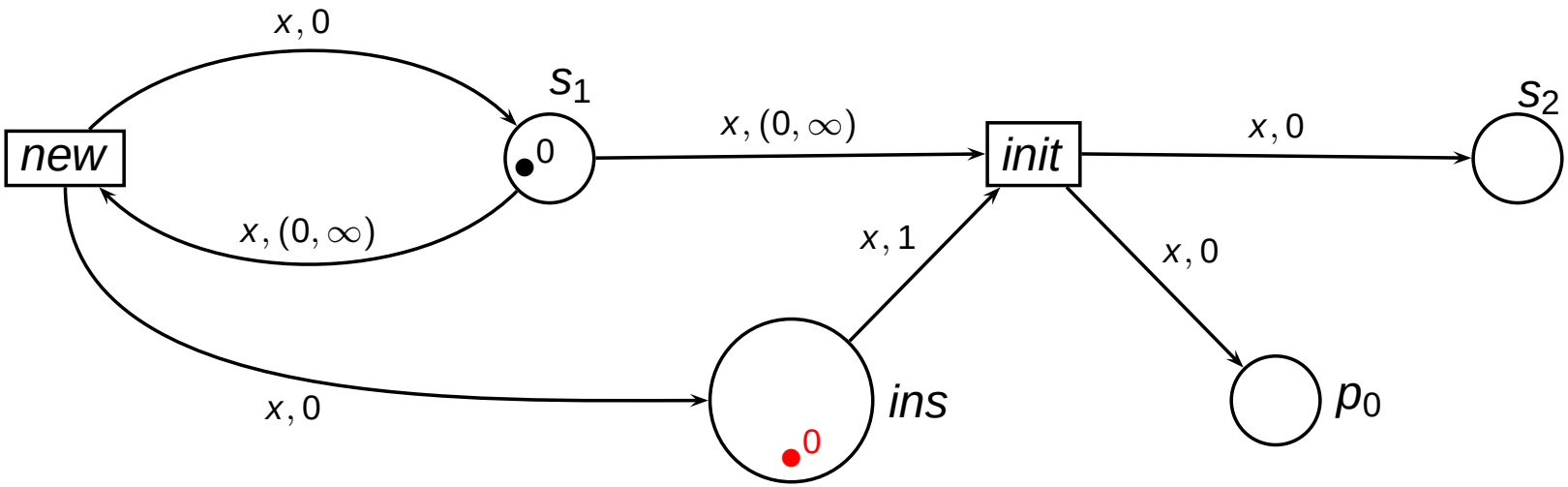
Undecidability-Step1: Creation of Instances



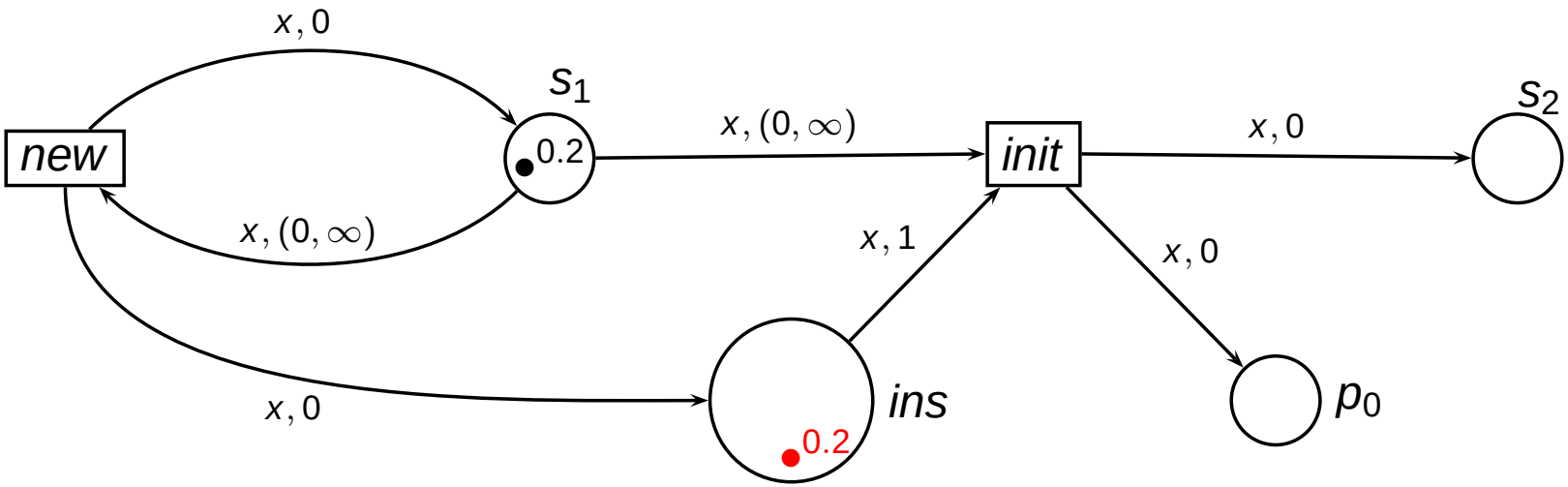
Undecidability-Step1: Creation of Instances



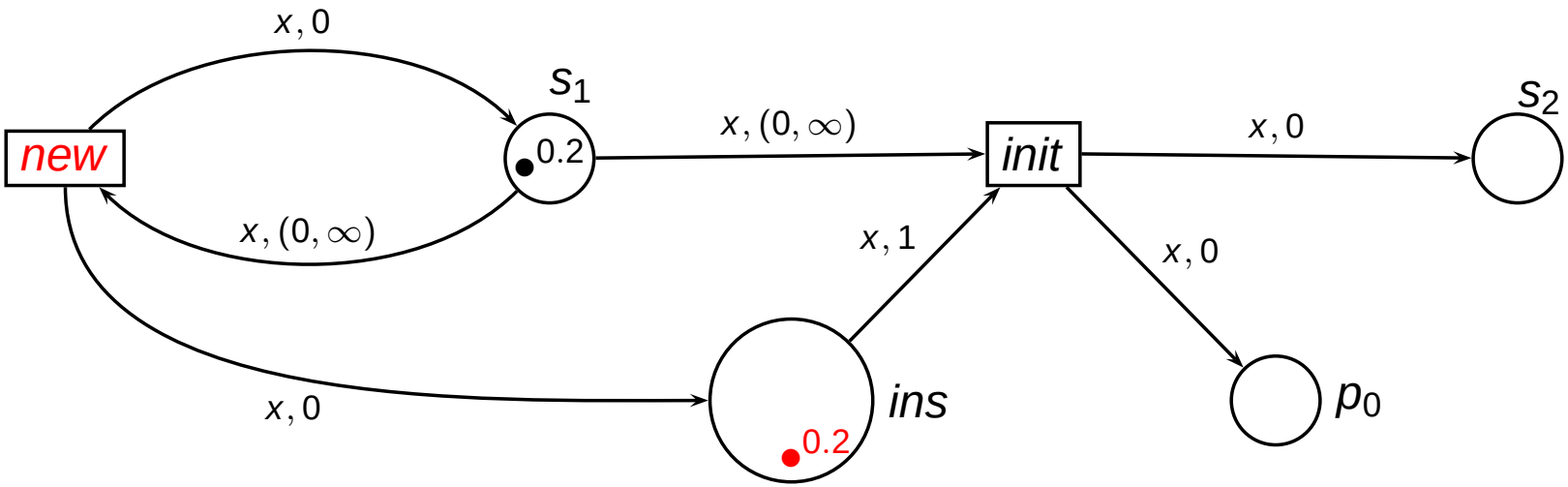
Undecidability-Step1: Creation of Instances



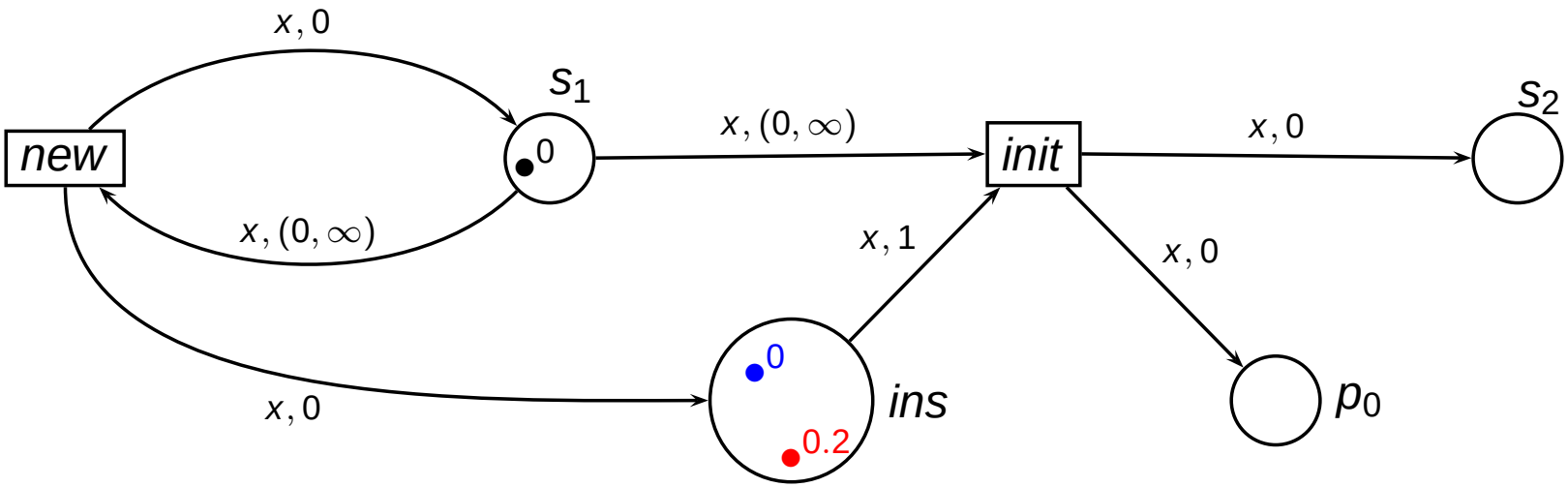
Undecidability-Step1: Creation of Instances



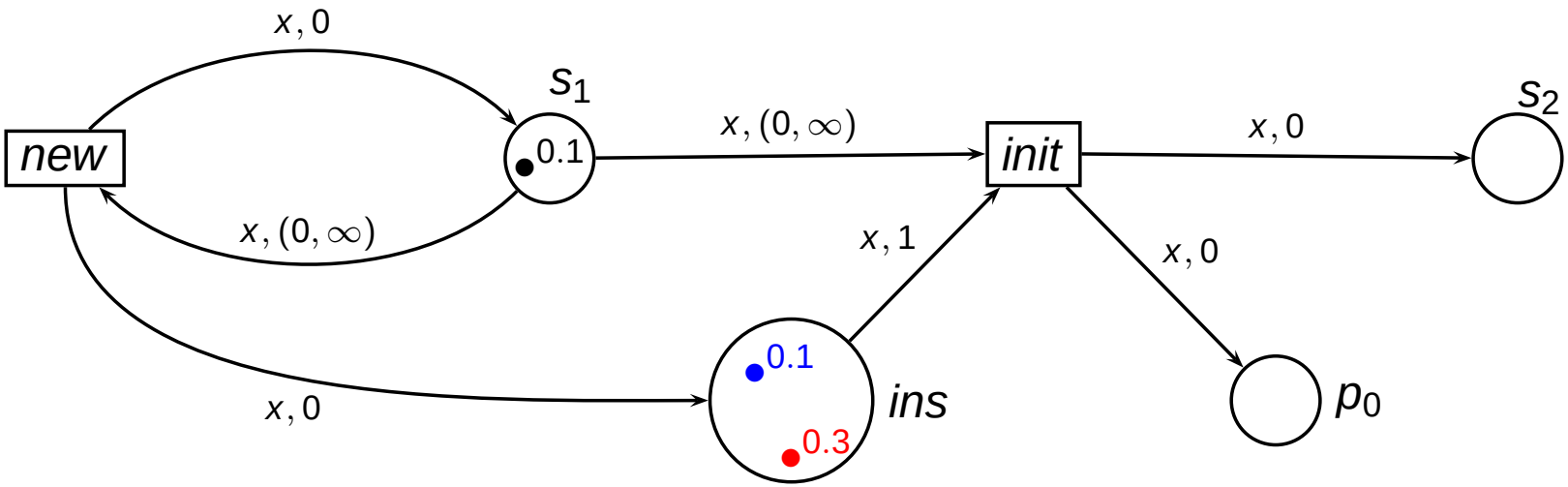
Undecidability-Step1: Creation of Instances



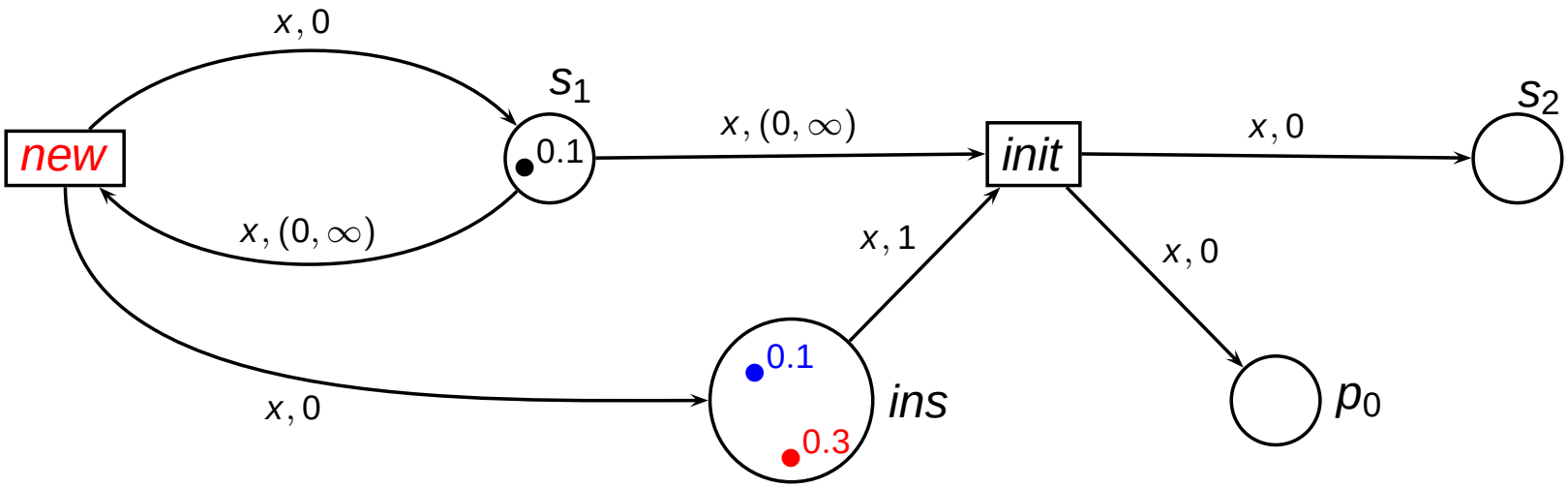
Undecidability-Step1: Creation of Instances



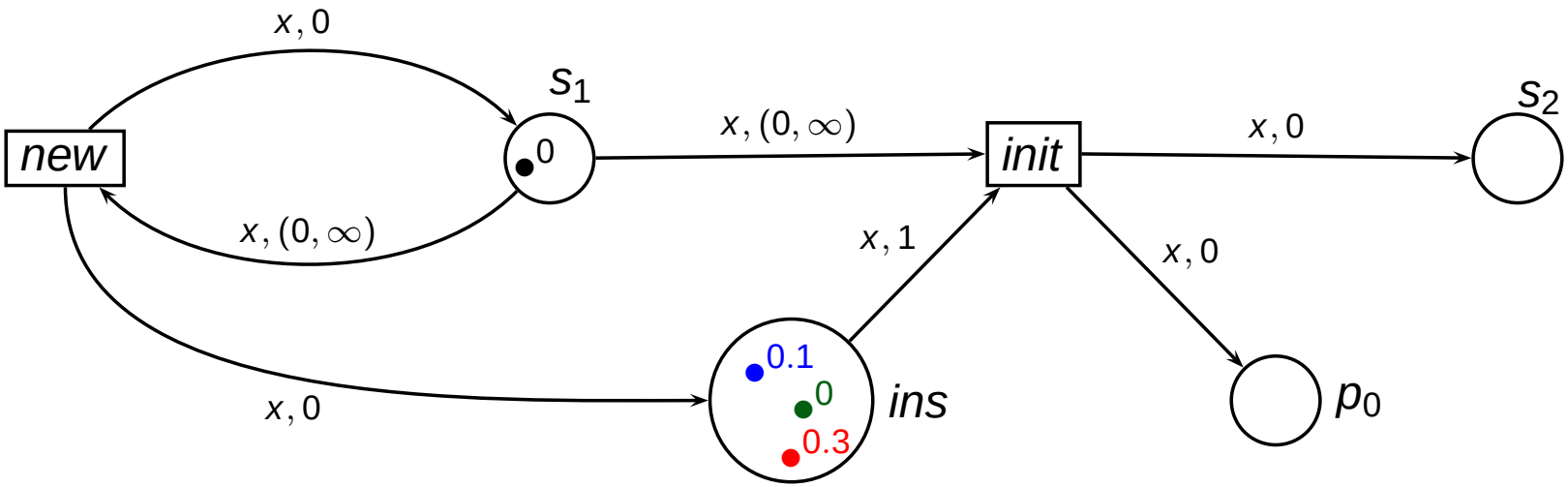
Undecidability-Step1: Creation of Instances



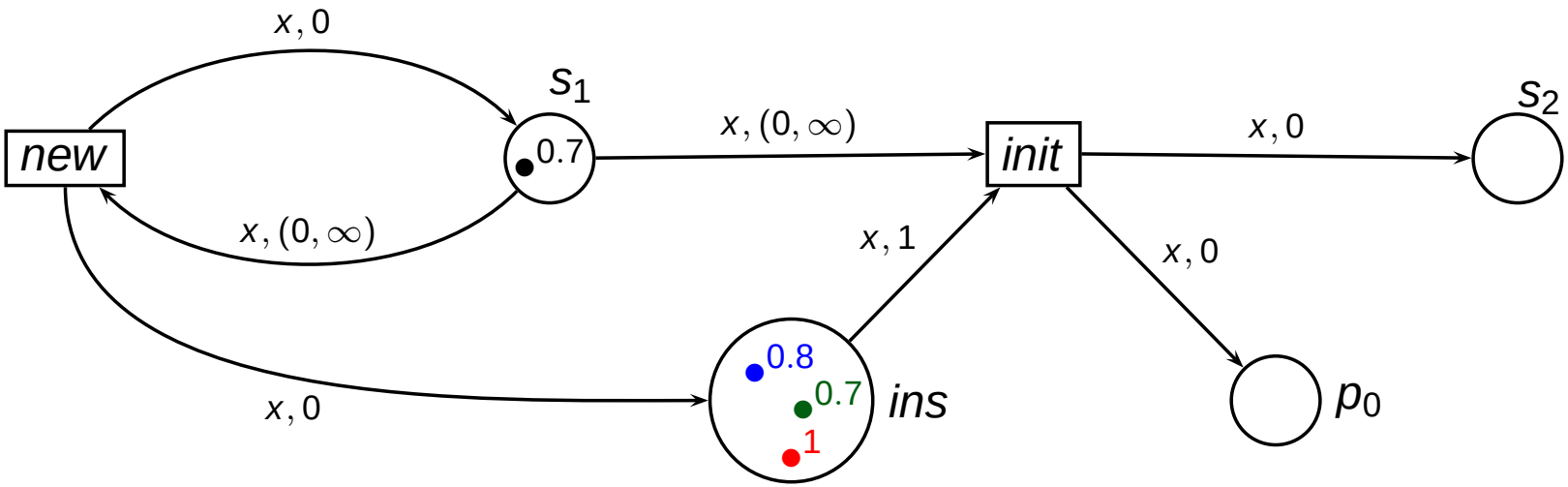
Undecidability-Step1: Creation of Instances



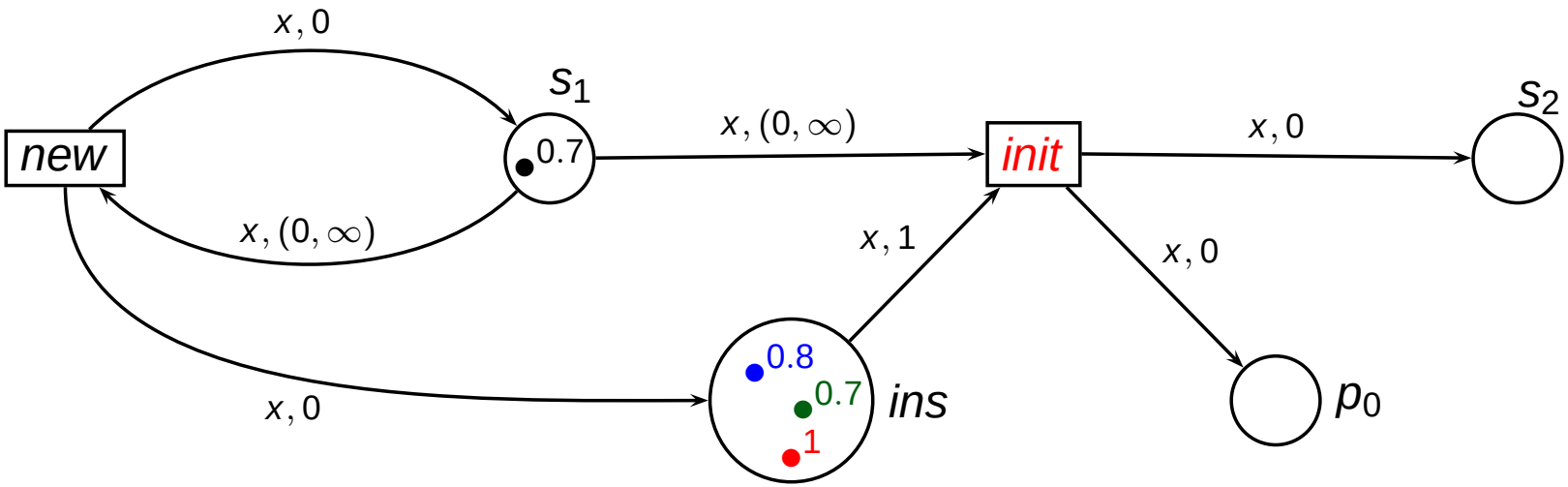
Undecidability-Step1: Creation of Instances



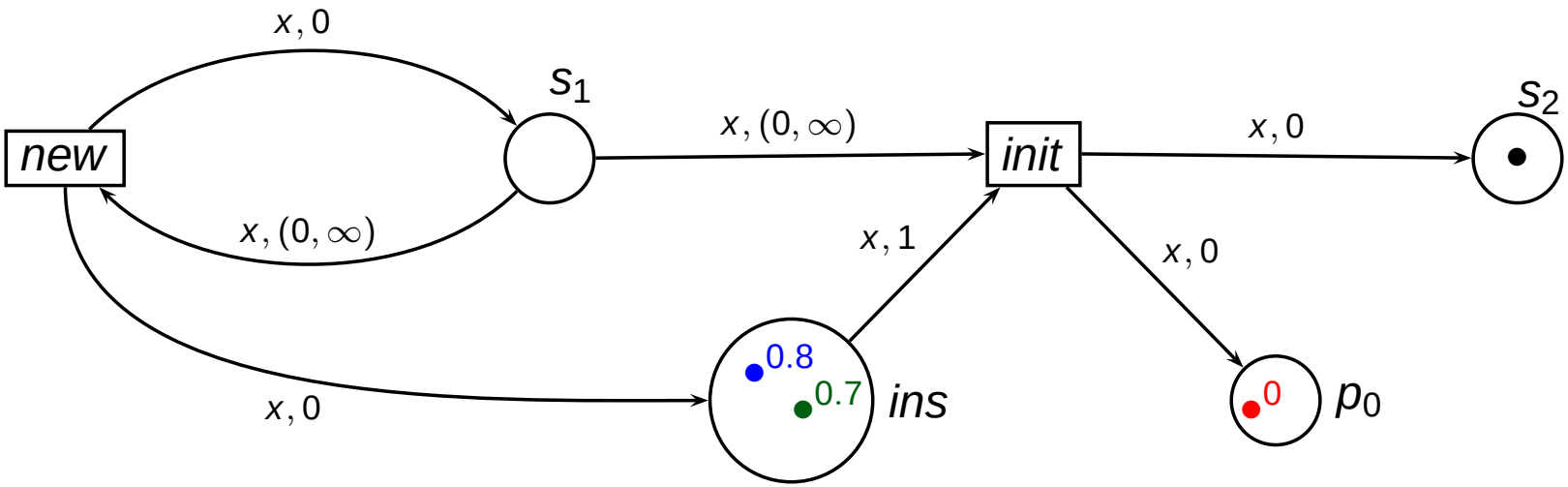
Undecidability-Step1: Creation of Instances



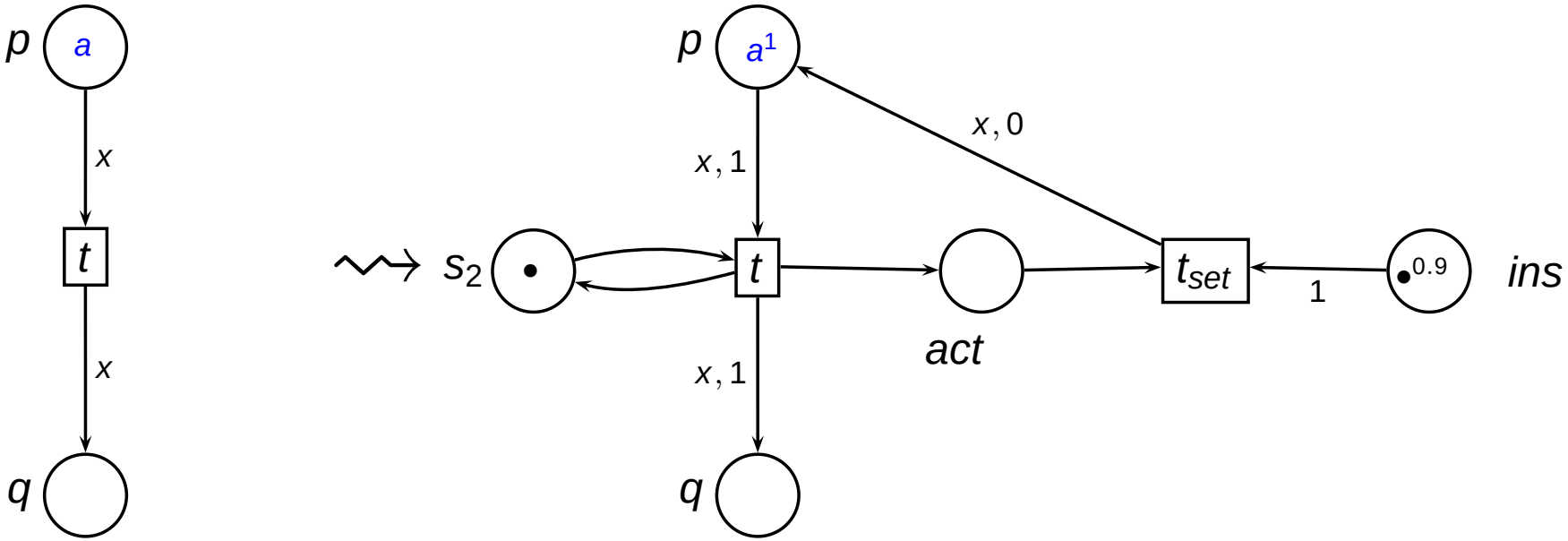
Undecidability-Step1: Creation of Instances



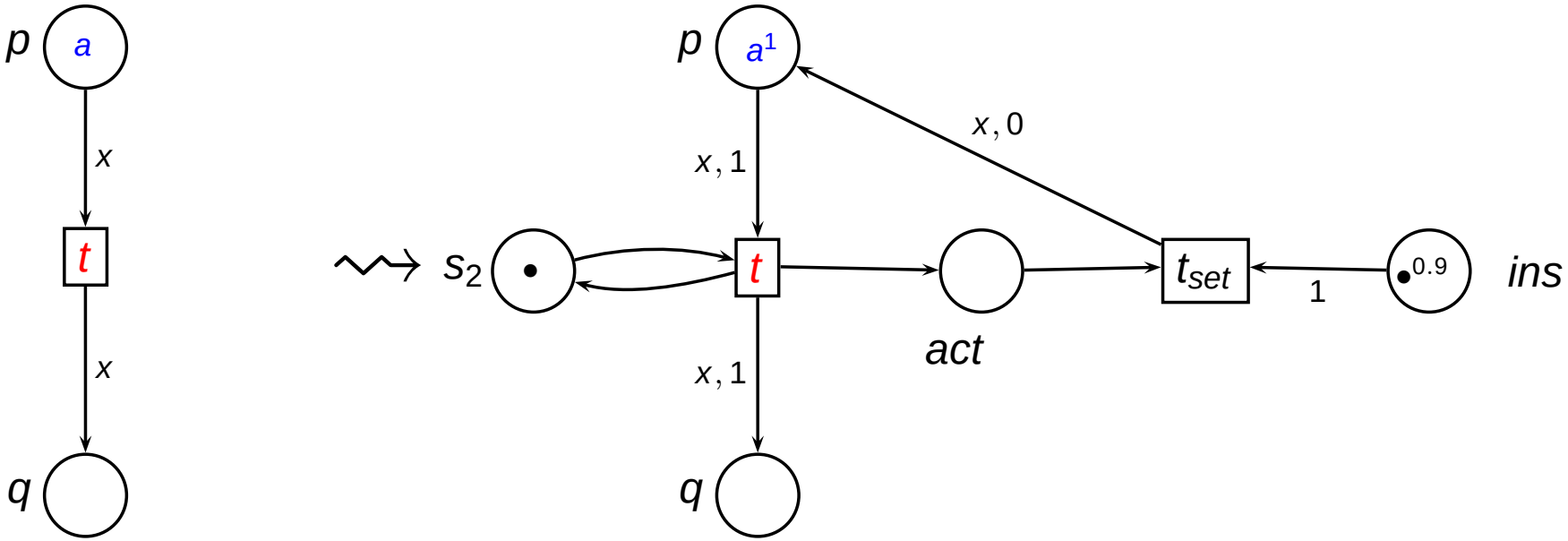
Undecidability-Step1: Creation of Instances



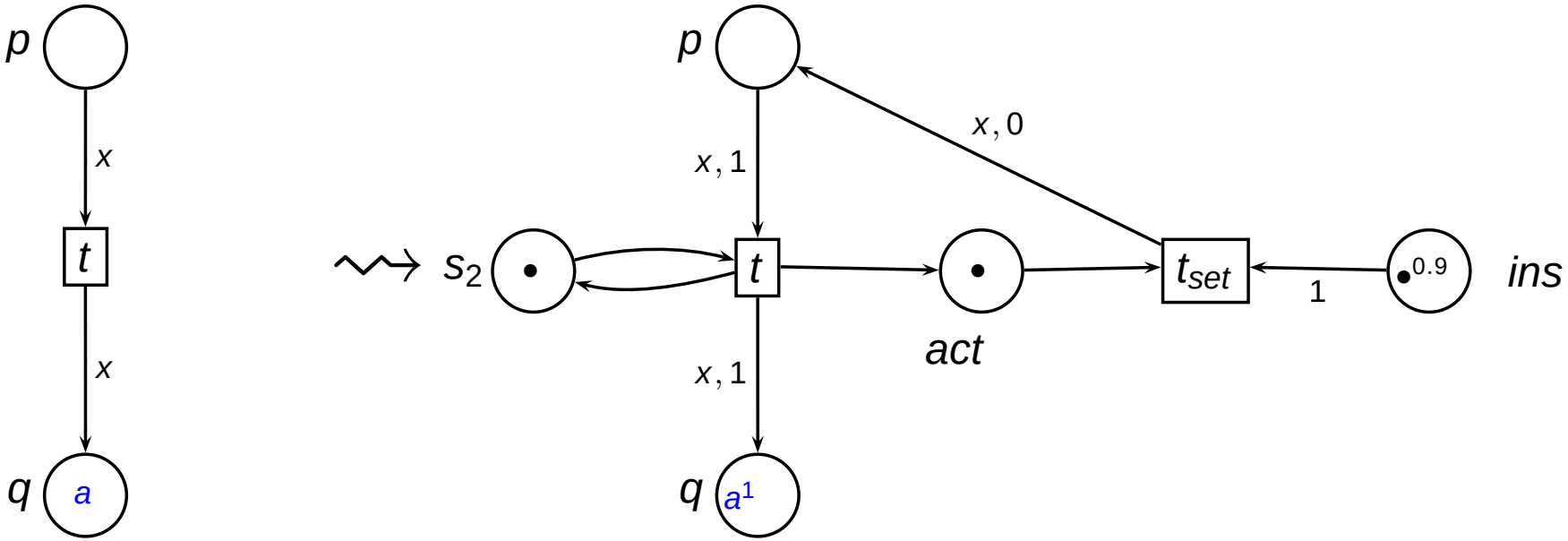
Undecidability-Step2.1: Firing of an autonomous transition



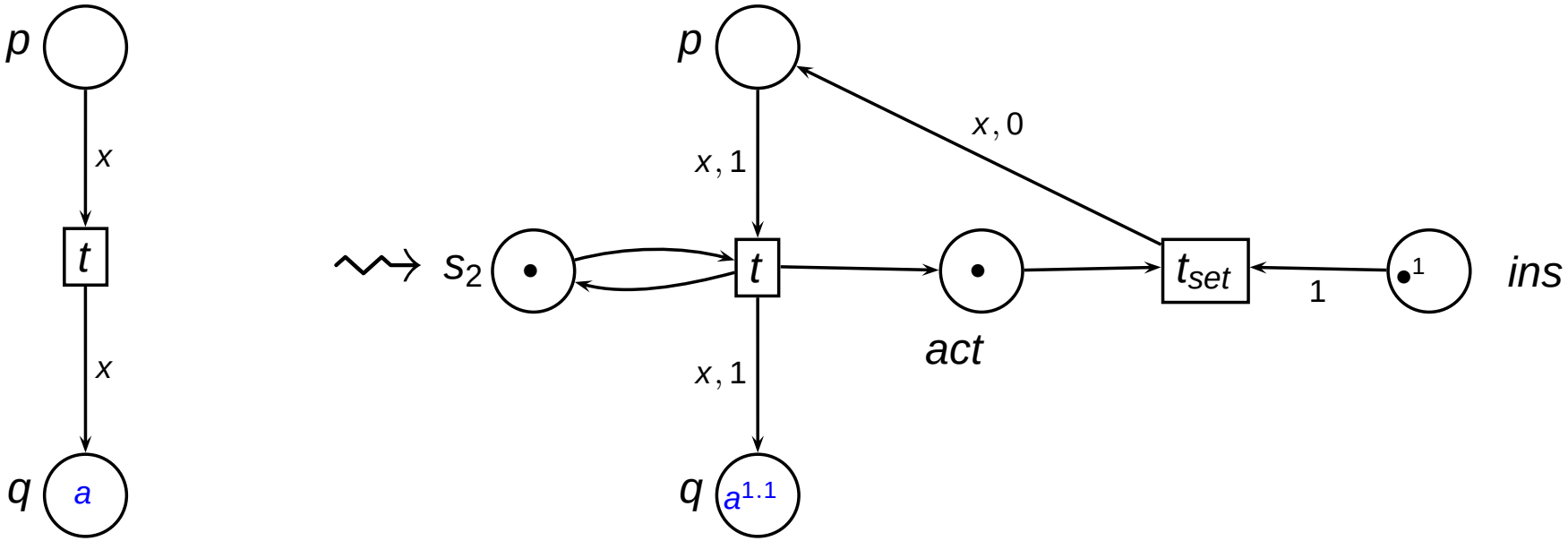
Undecidability-Step2.1: Firing of an autonomous transition



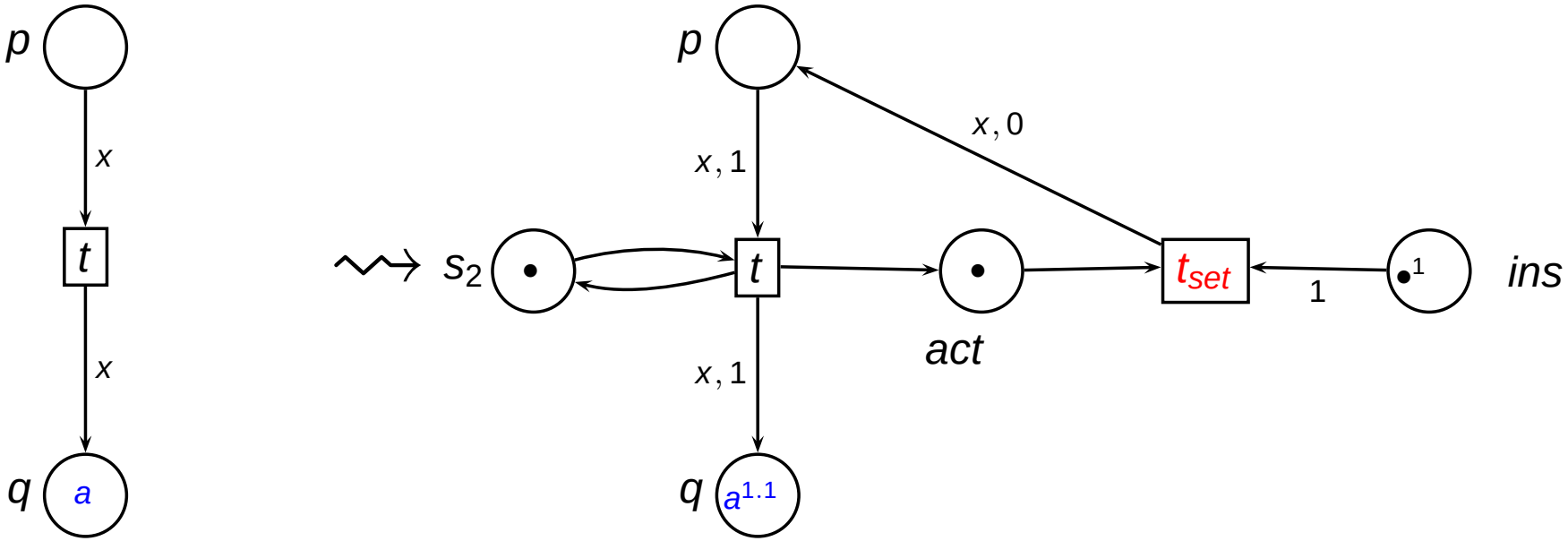
Undecidability-Step2.1: Firing of an autonomous transition



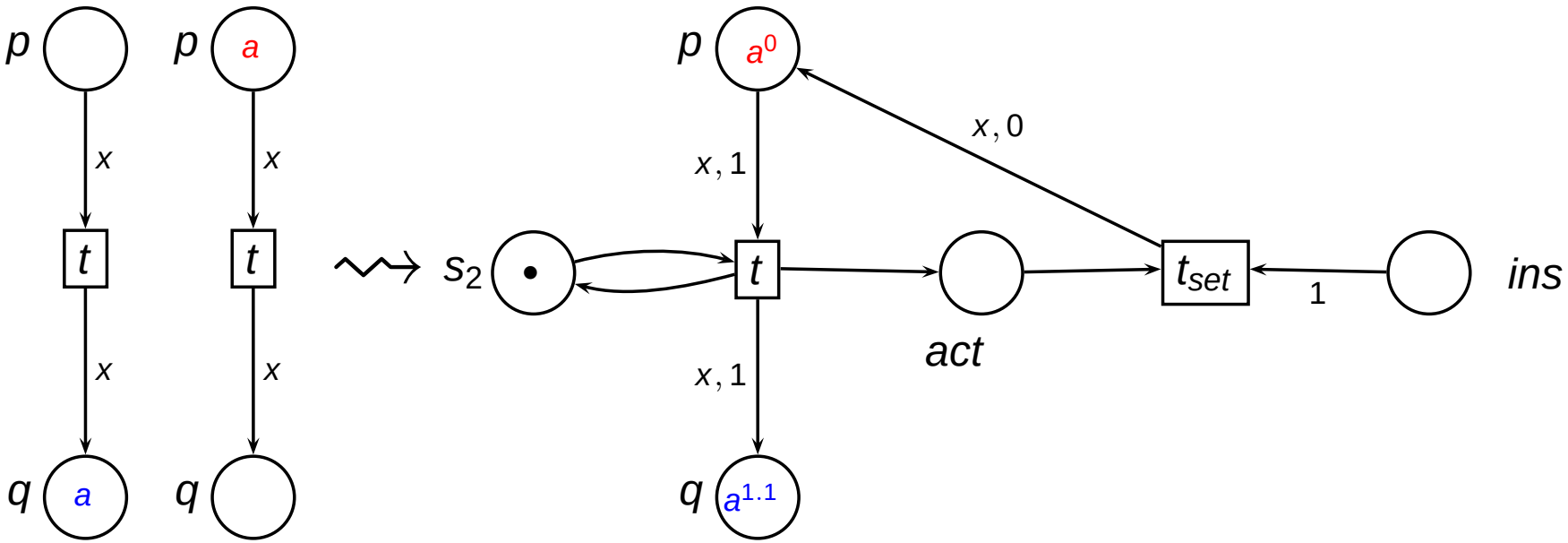
Undecidability-Step2.1: Firing of an autonomous transition



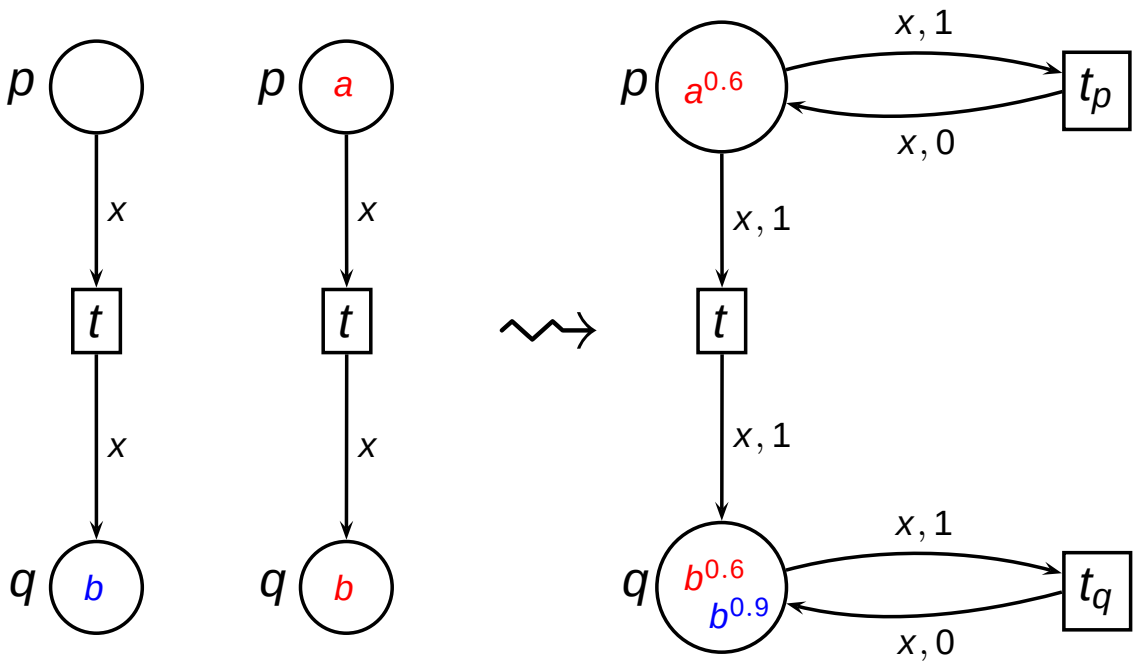
Undecidability-Step2.1: Firing of an autonomous transition



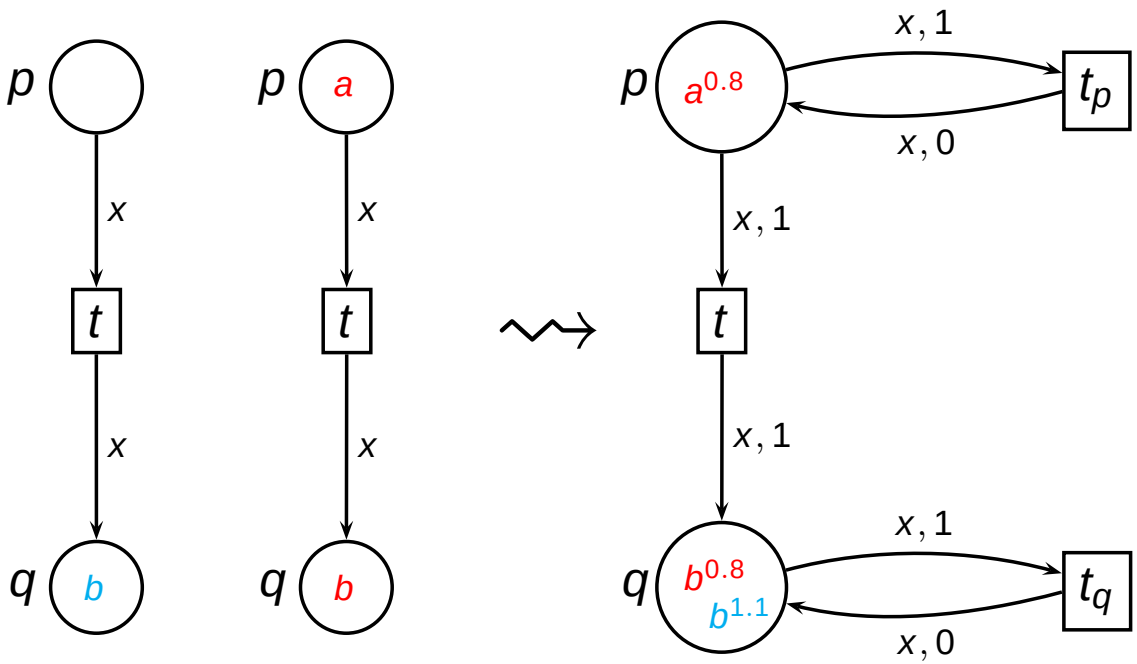
Undecidability-Step2.1: Firing of an autonomous transition



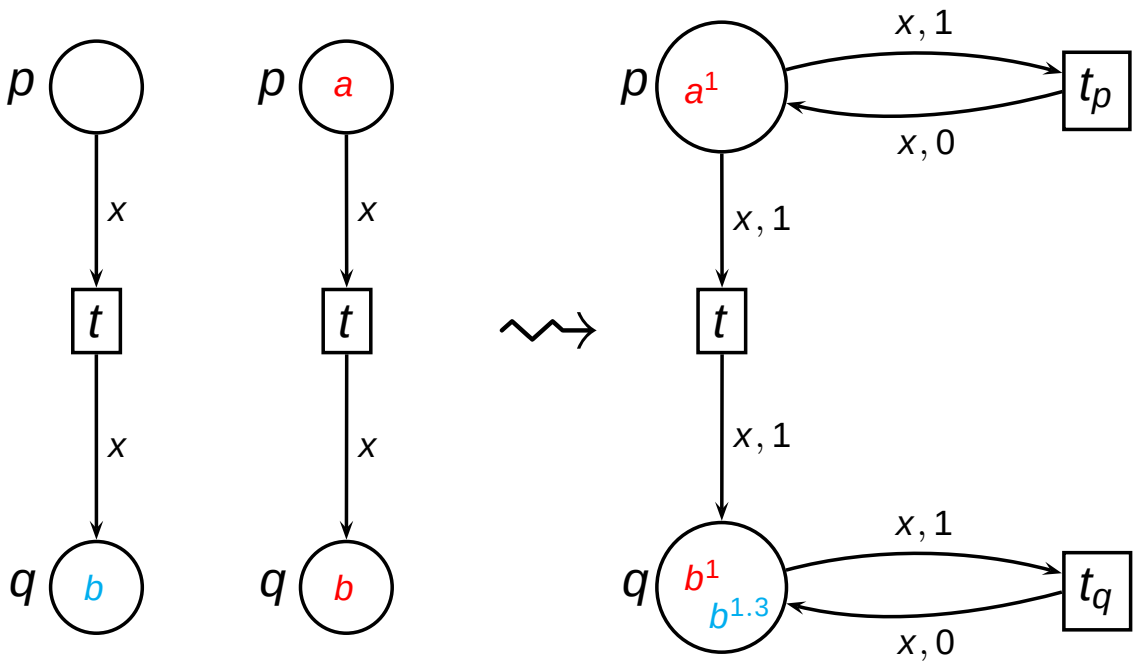
Undecidability-Step2.2: Reseting tokens



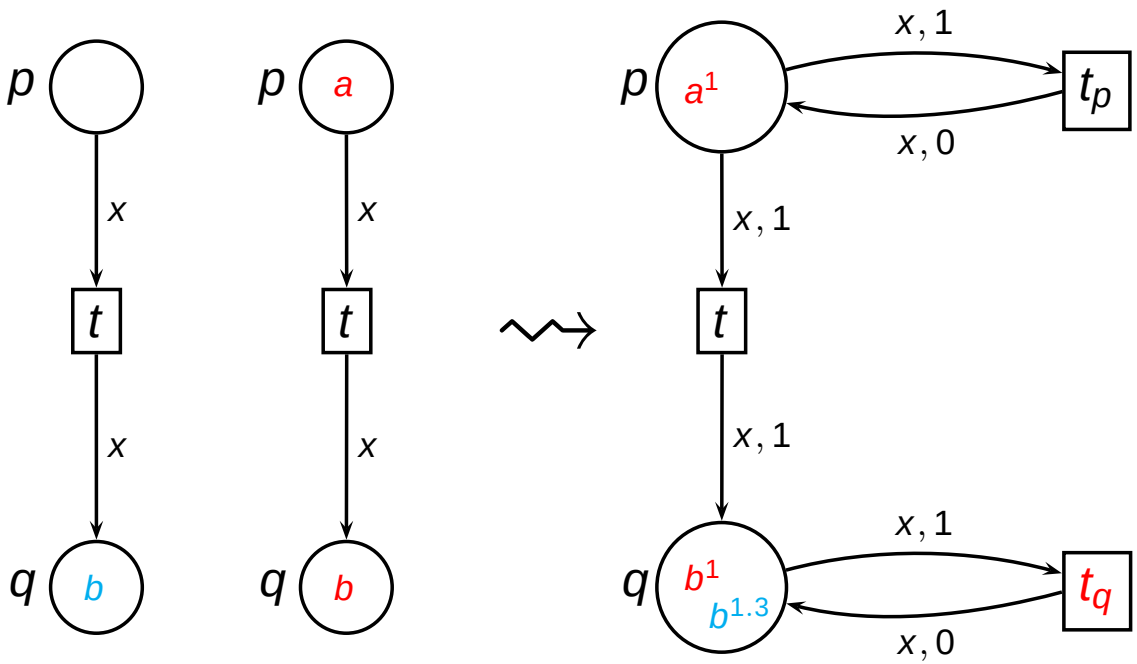
Undecidability-Step2.2: Reseting tokens



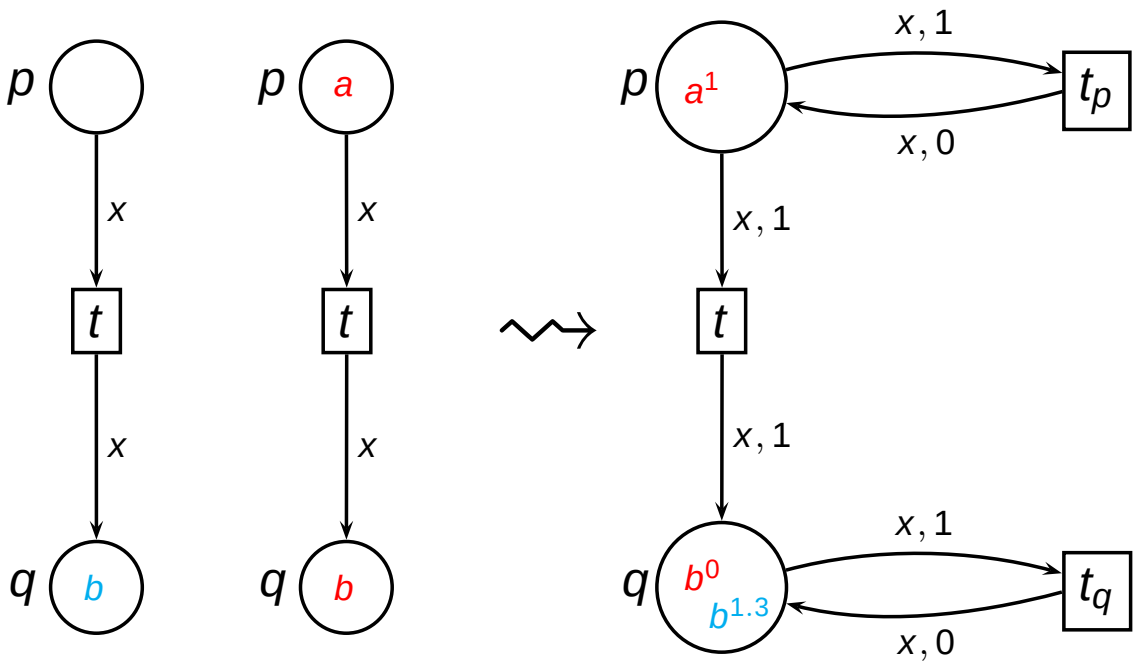
Undecidability-Step2.2: Reseting tokens



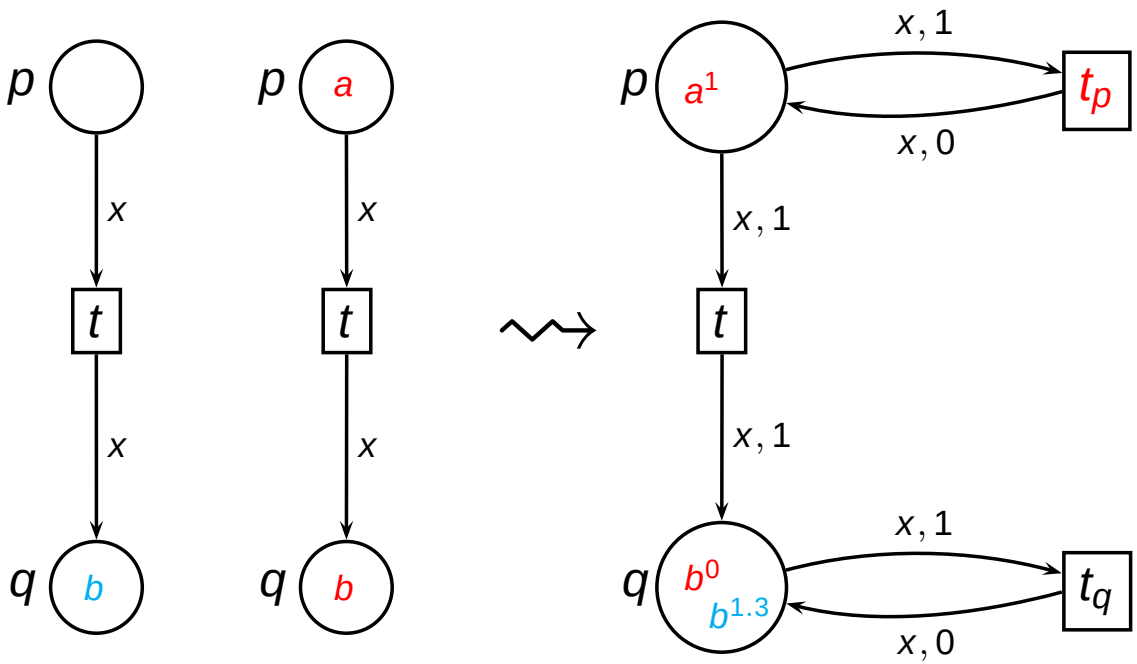
Undecidability-Step2.2: Reseting tokens



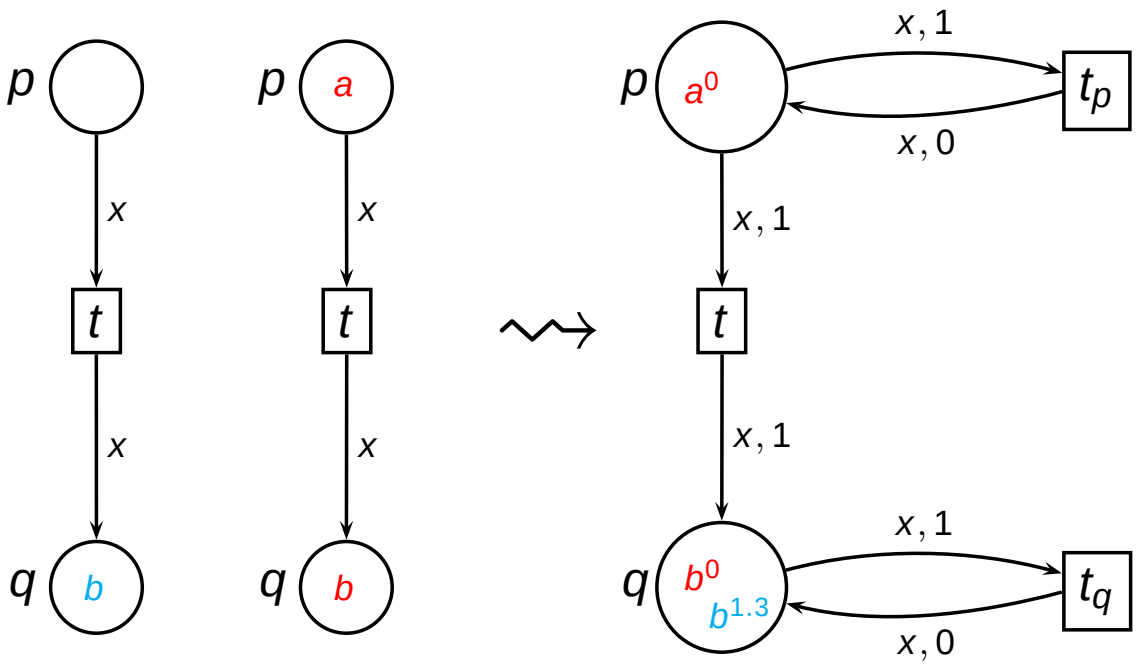
Undecidability-Step2.2: Reseting tokens



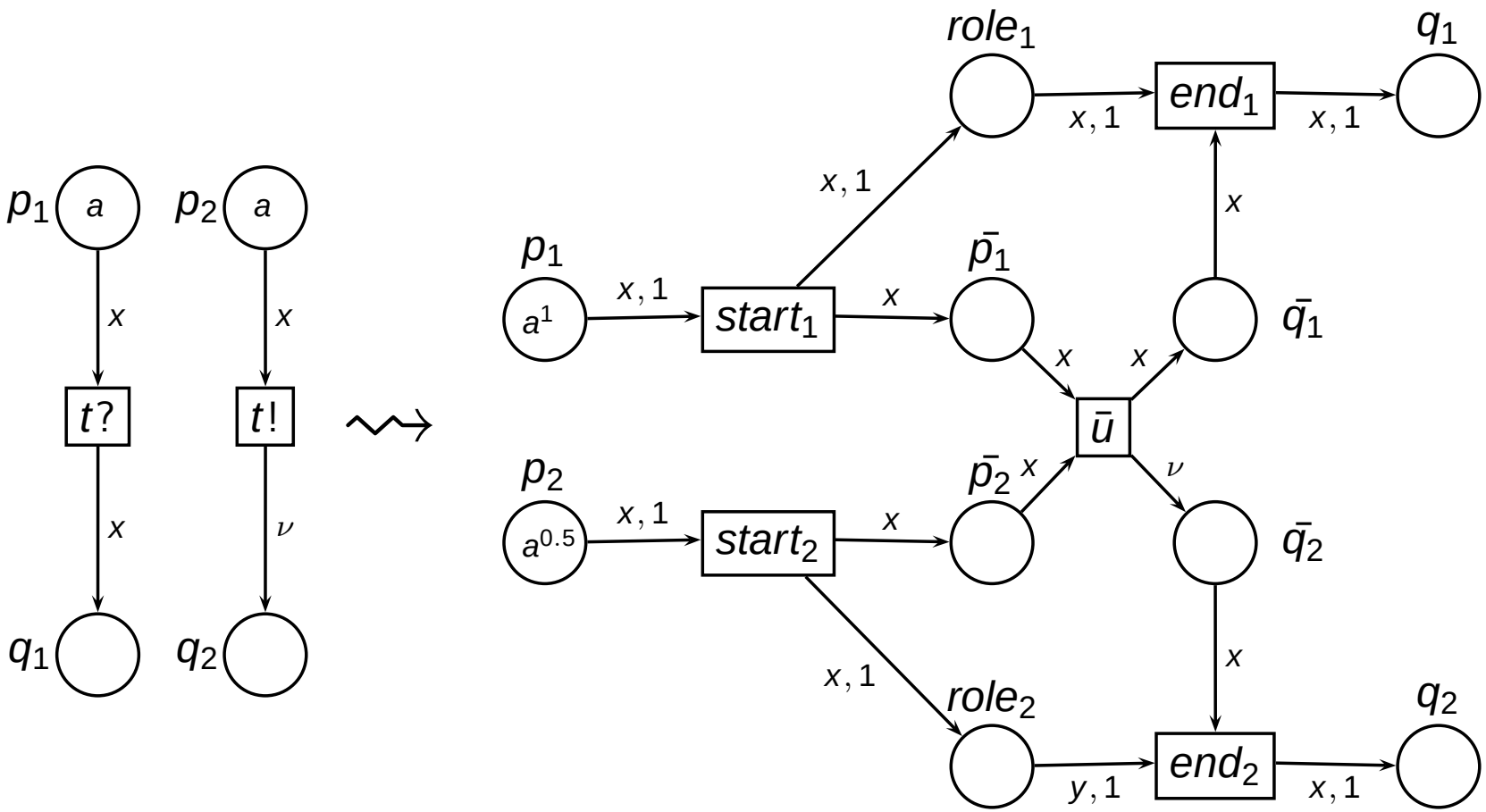
Undecidability-Step2.2: Reseting tokens



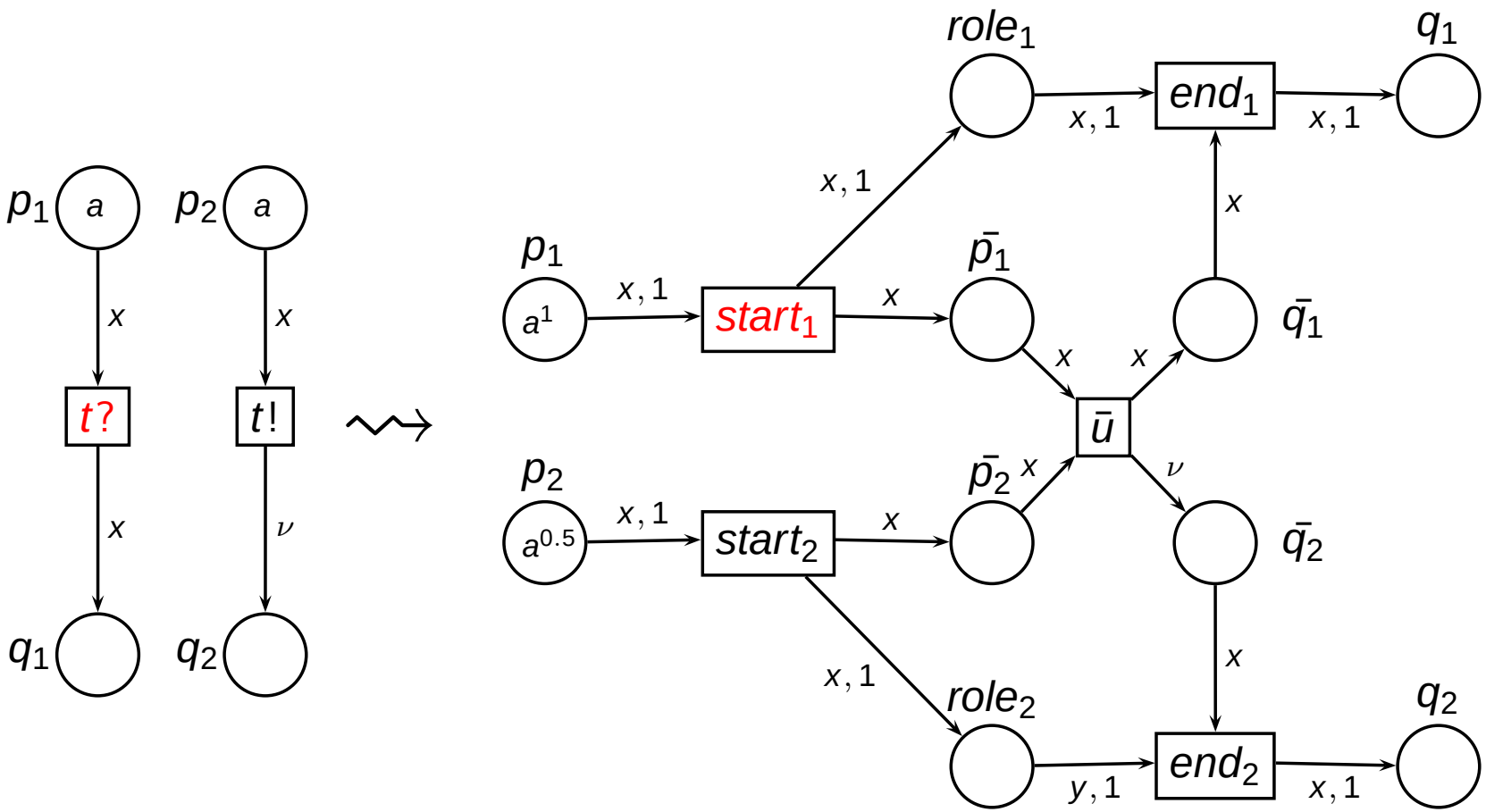
Undecidability-Step2.2: Reseting tokens



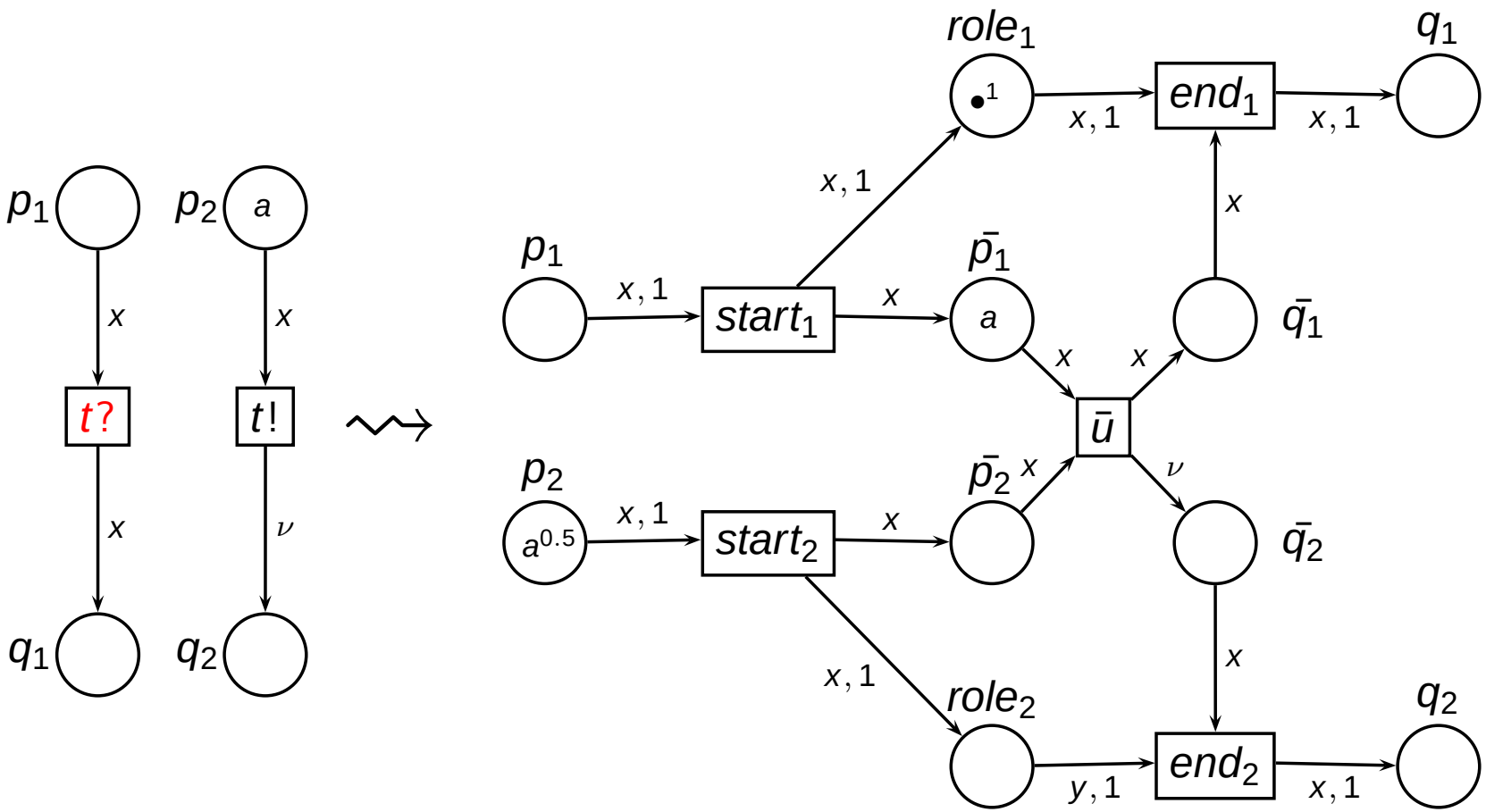
Undecidability-Step2.3: Firing of a Synchronizing transition



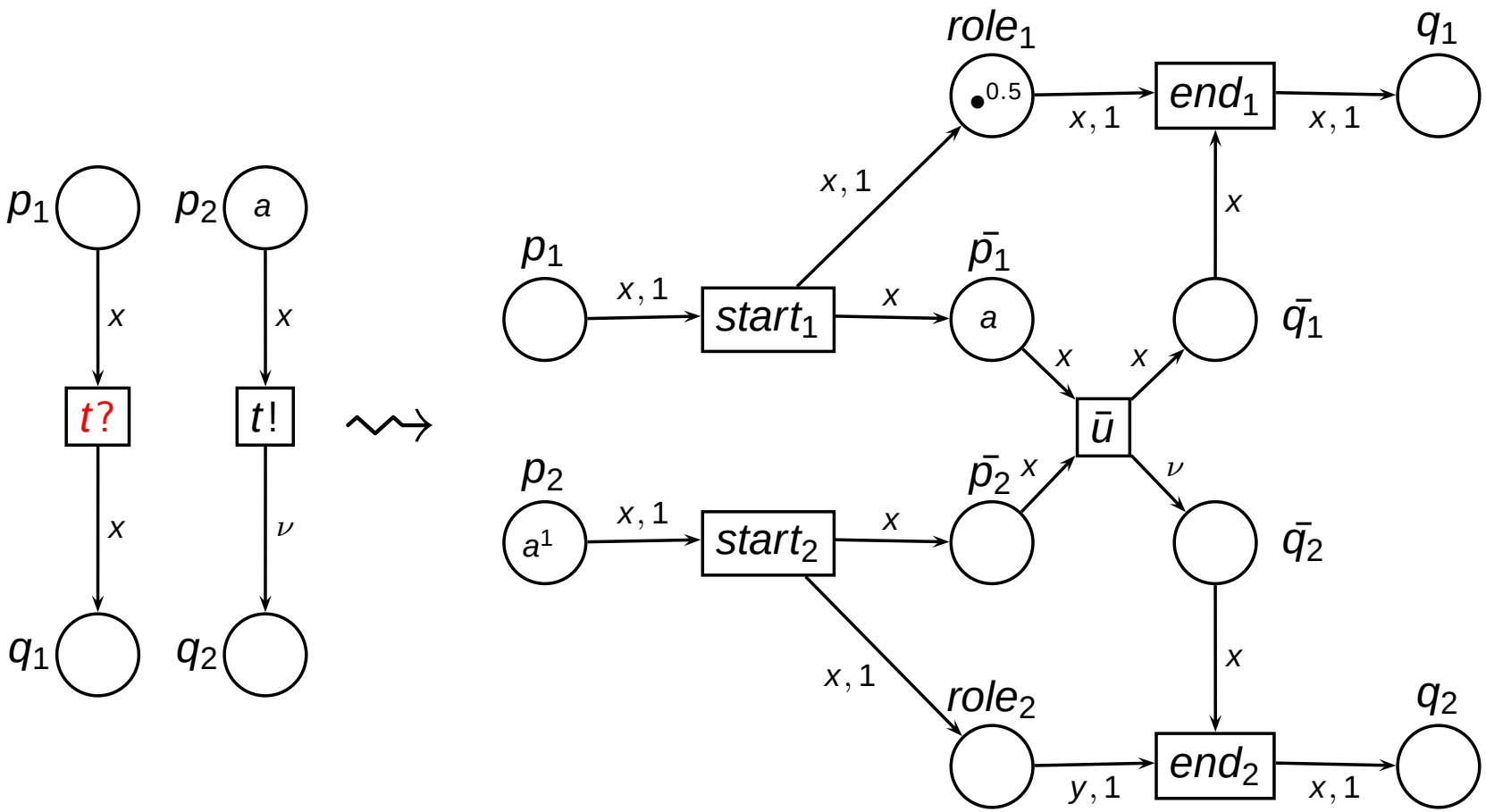
Undecidability-Step2.3: Firing of a Synchronizing transition



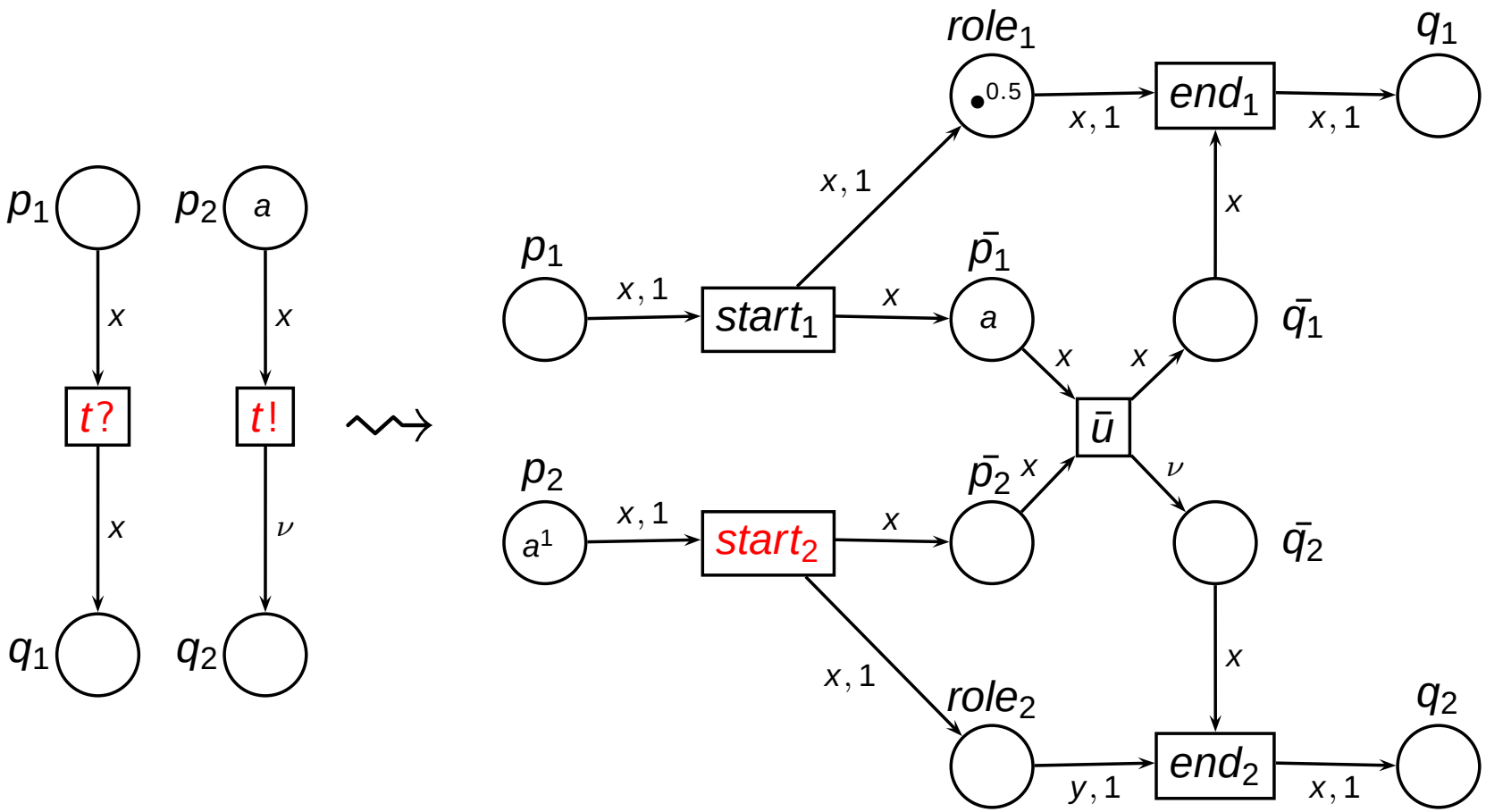
Undecidability-Step2.3: Firing of a Synchronizing transition



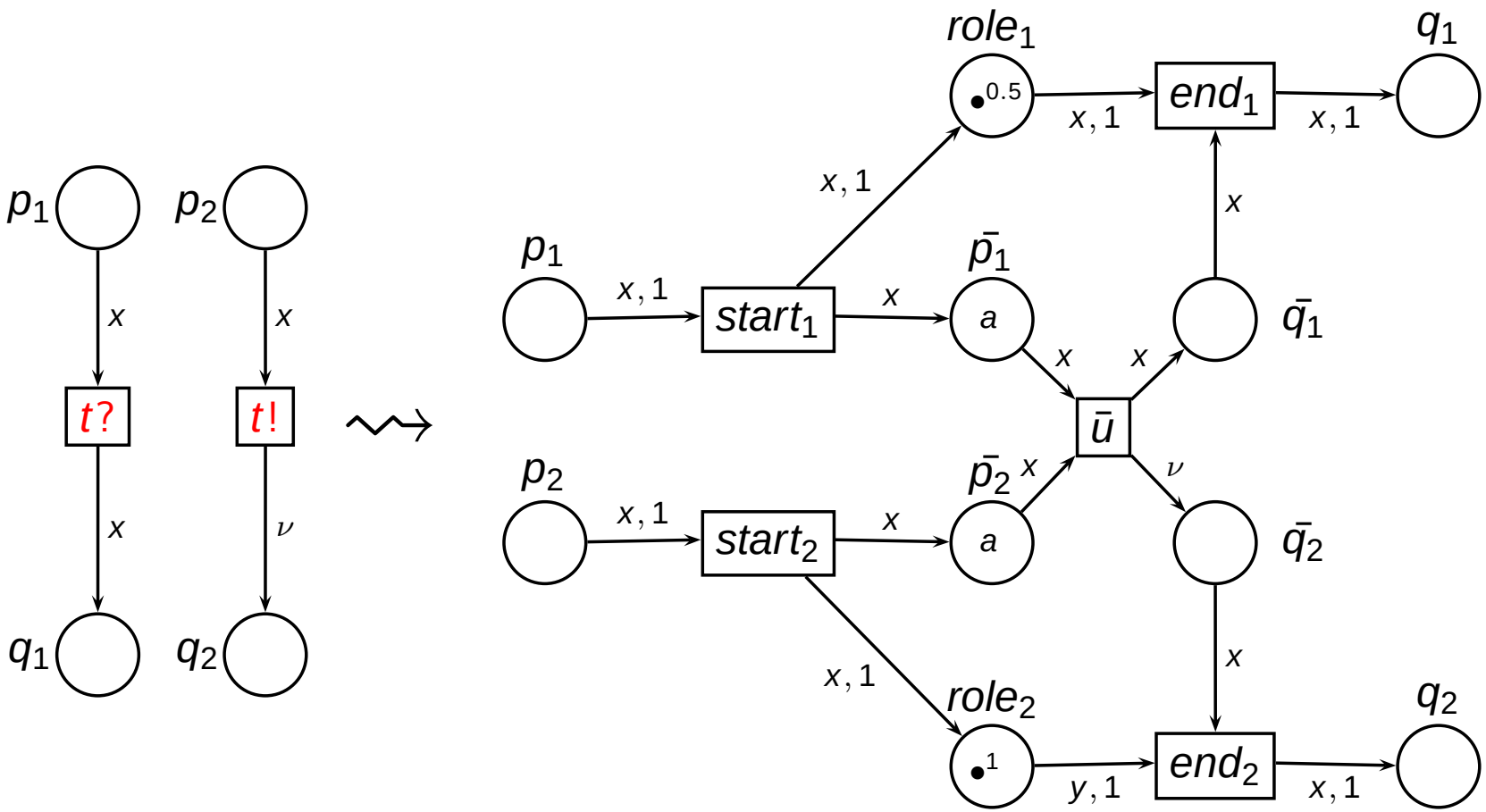
Undecidability-Step2.3: Firing of a Synchronizing transition



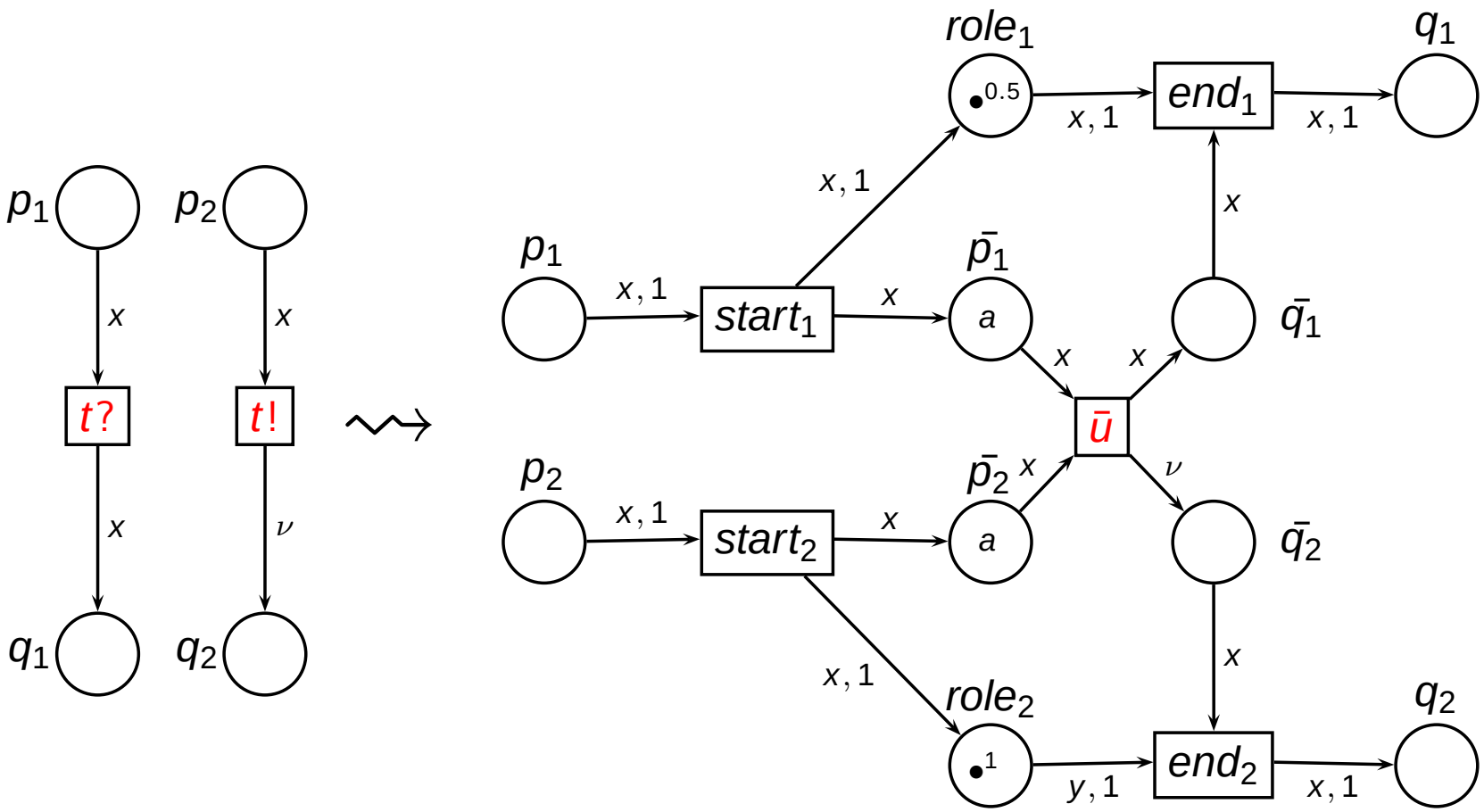
Undecidability-Step2.3: Firing of a Synchronizing transition



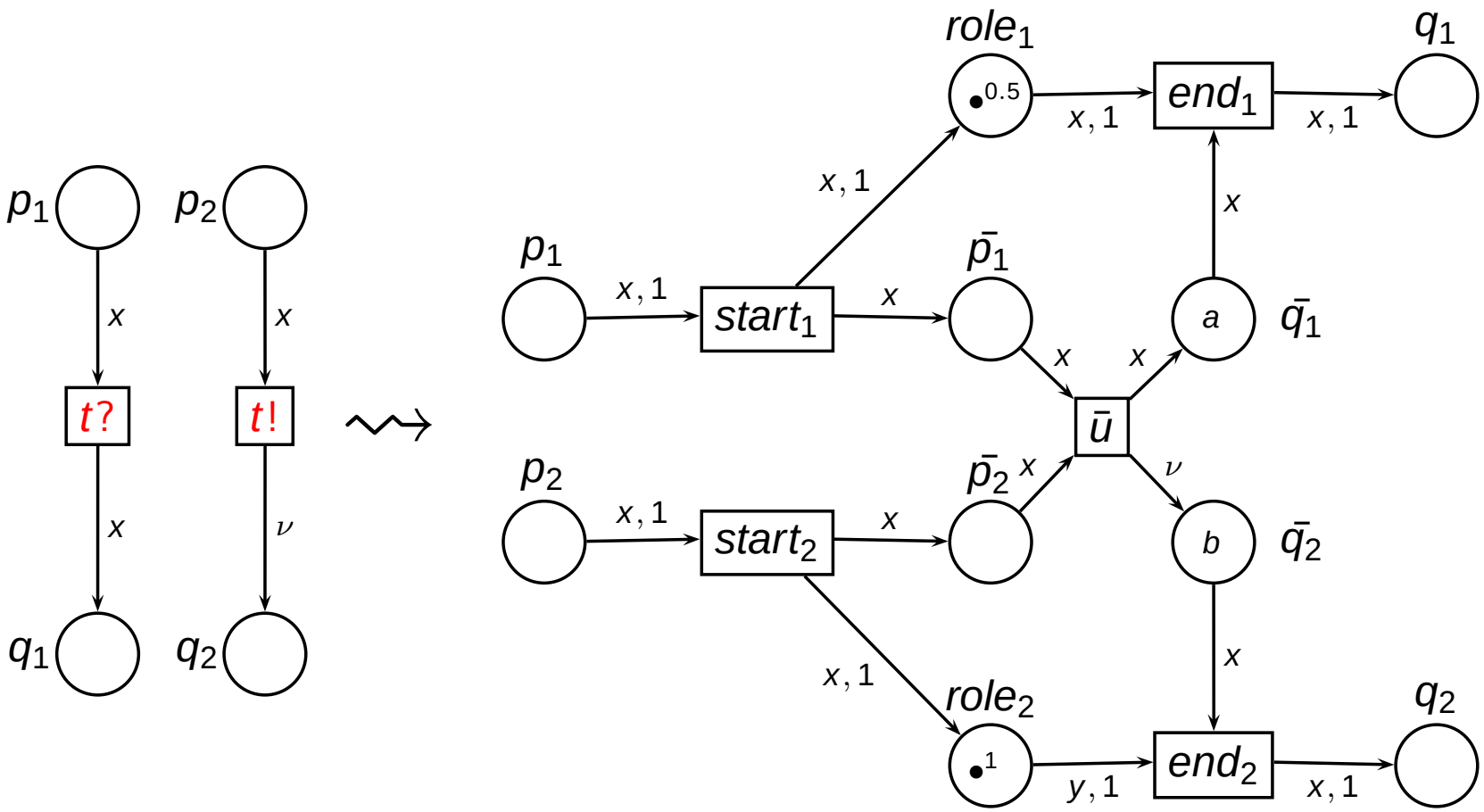
Undecidability-Step2.3: Firing of a Synchronizing transition



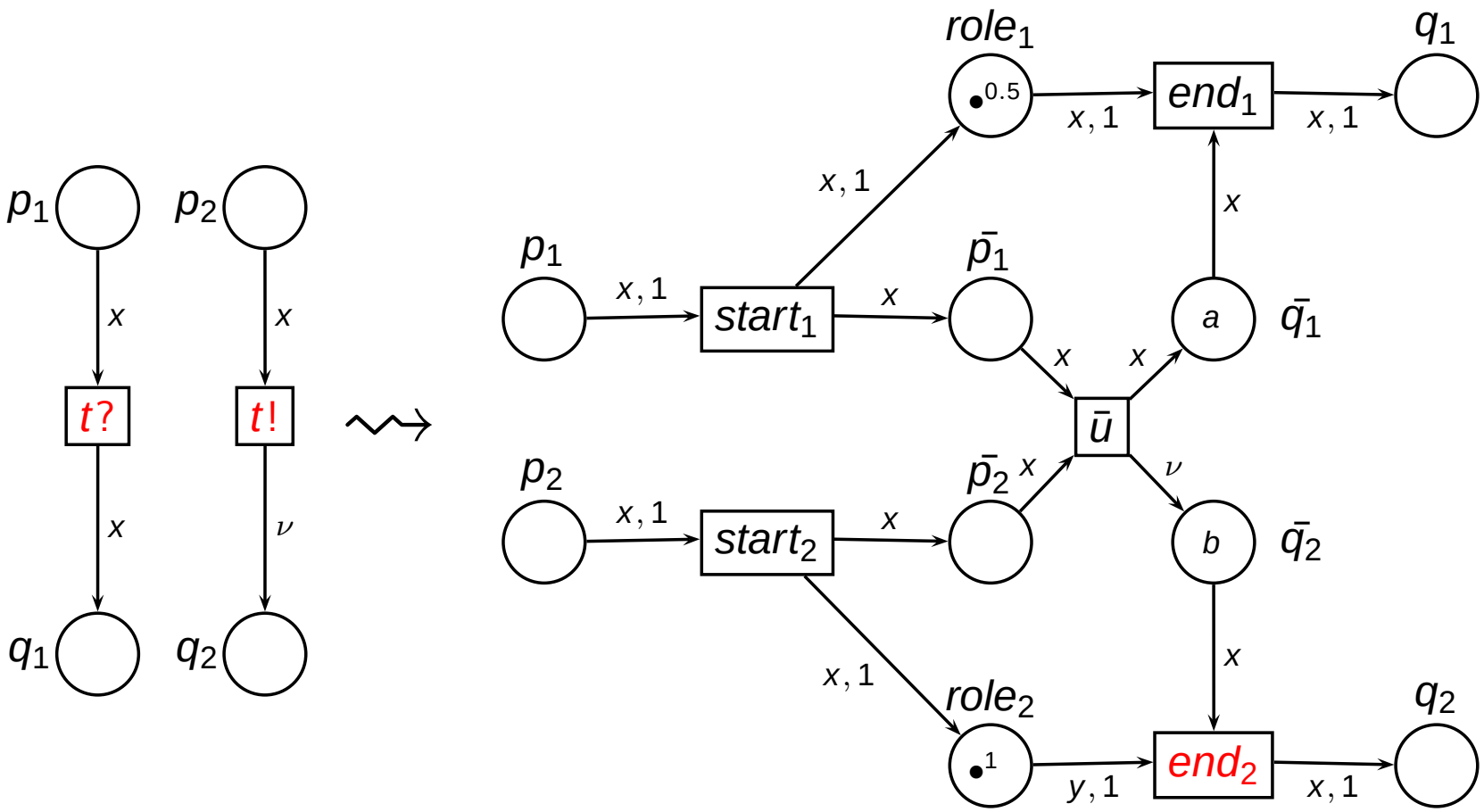
Undecidability-Step2.3: Firing of a Synchronizing transition



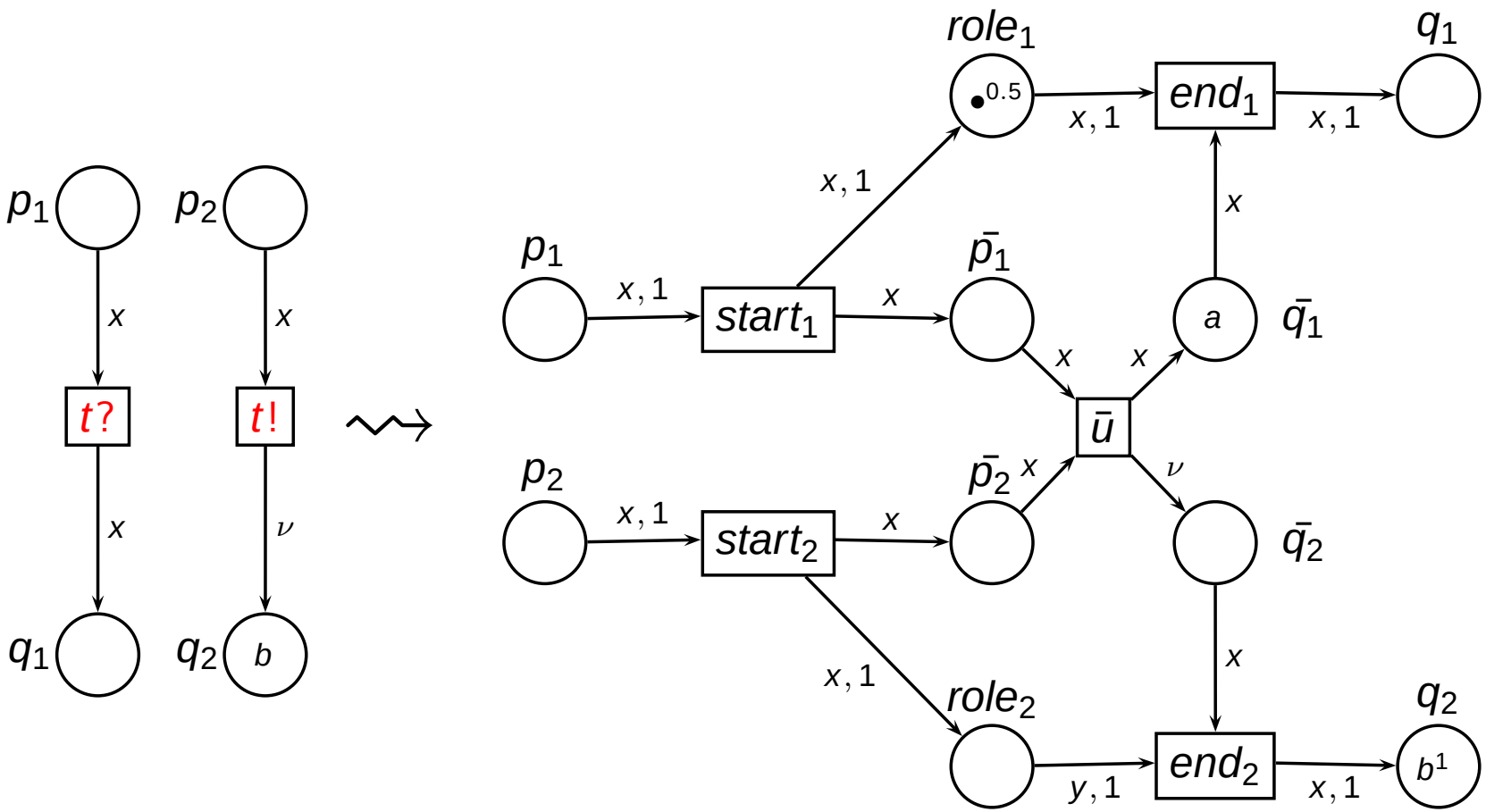
Undecidability-Step2.3: Firing of a Synchronizing transition



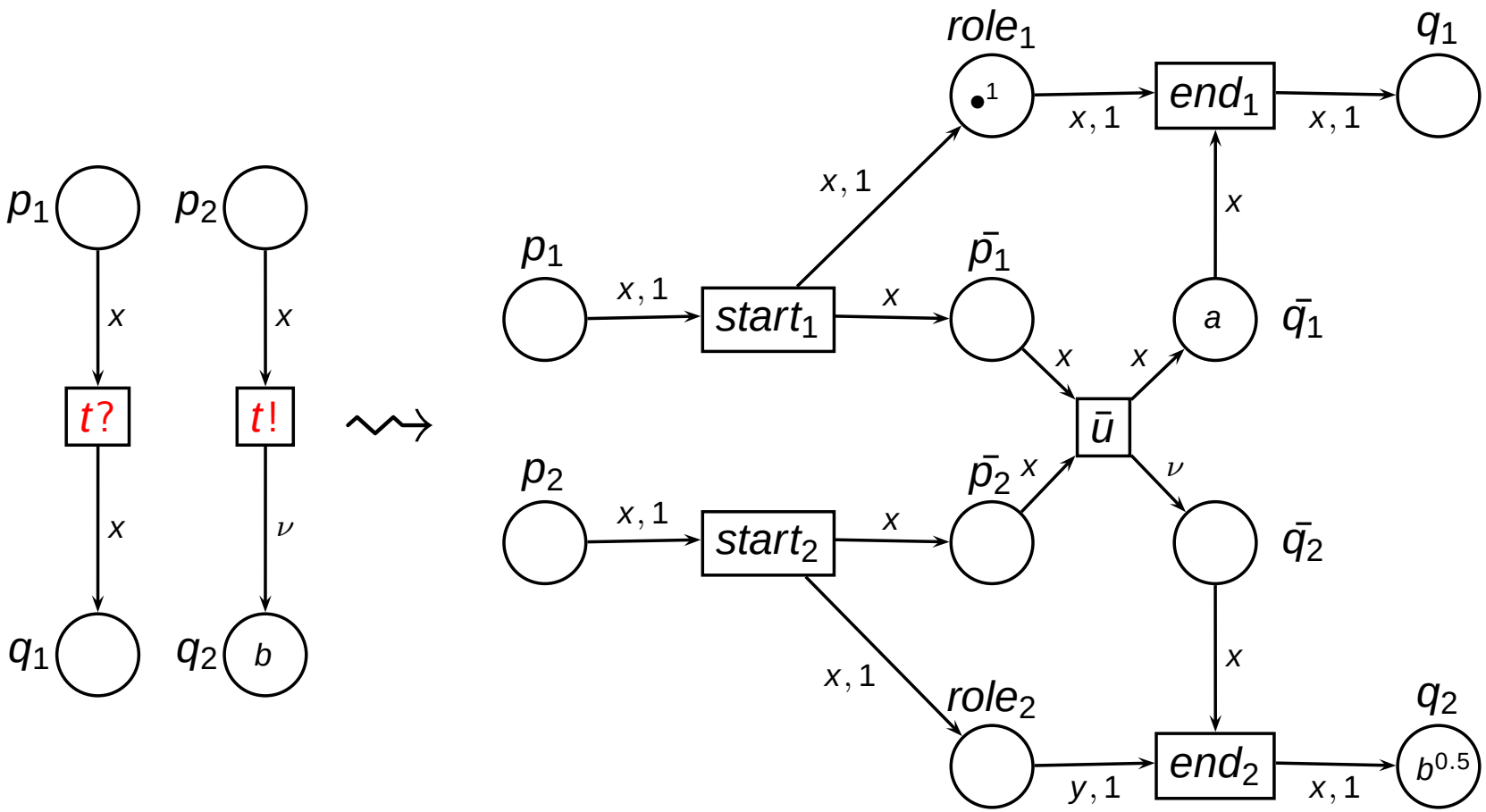
Undecidability-Step2.3: Firing of a Synchronizing transition



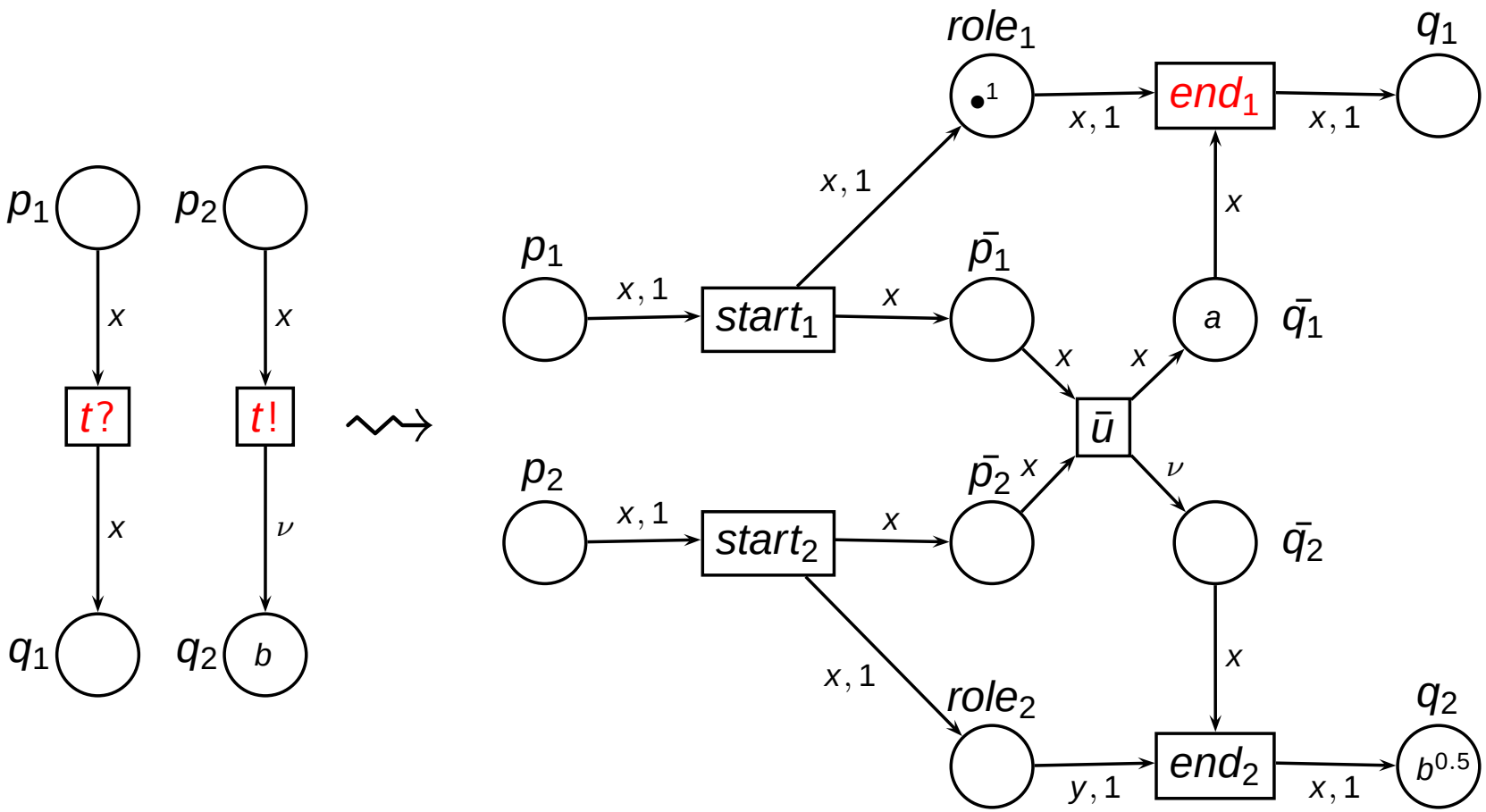
Undecidability-Step2.3: Firing of a Synchronizing transition



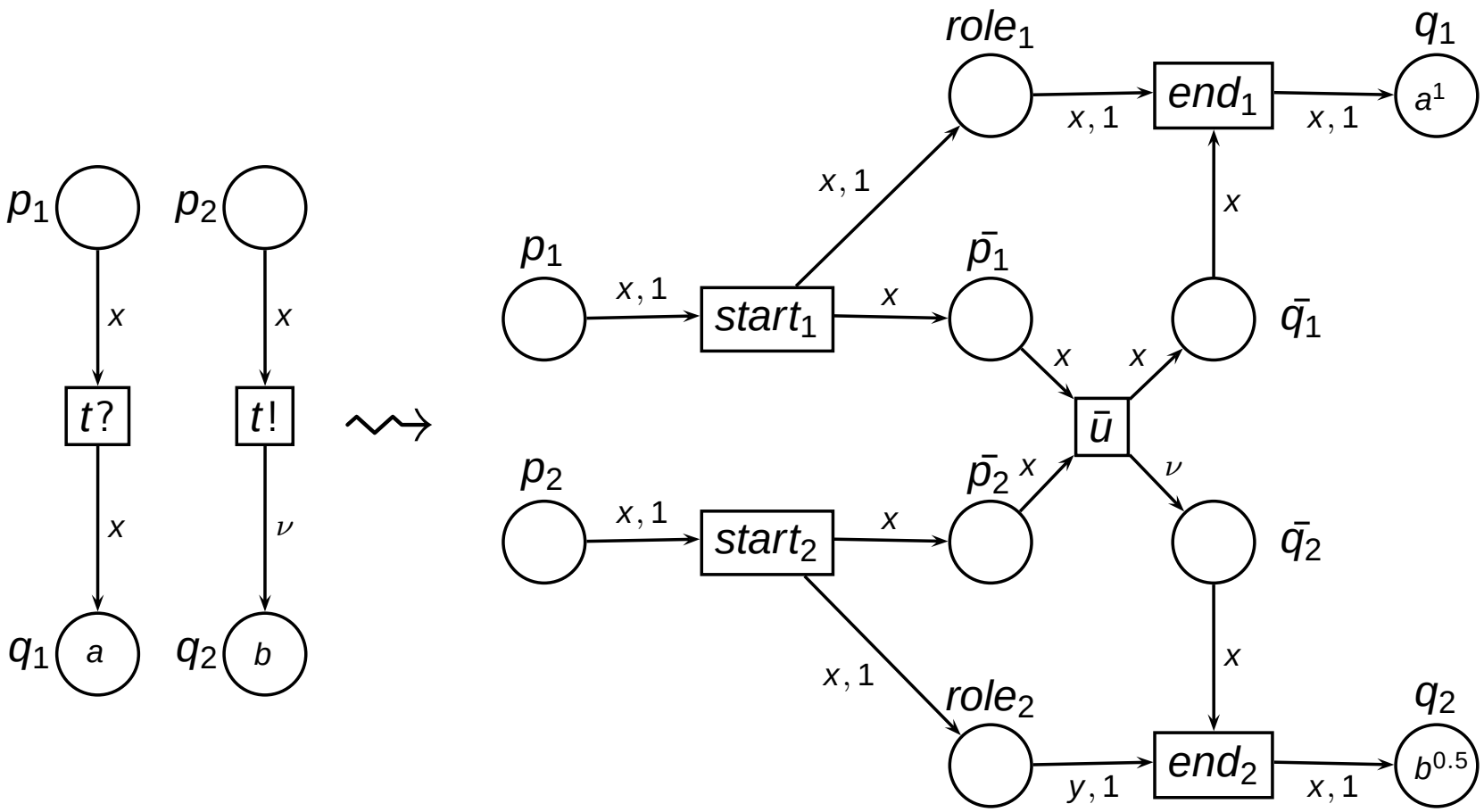
Undecidability-Step2.3: Firing of a Synchronizing transition



Undecidability-Step2.3: Firing of a Synchronizing transition



Undecidability-Step2.3: Firing of a Synchronizing transition



Undecidability

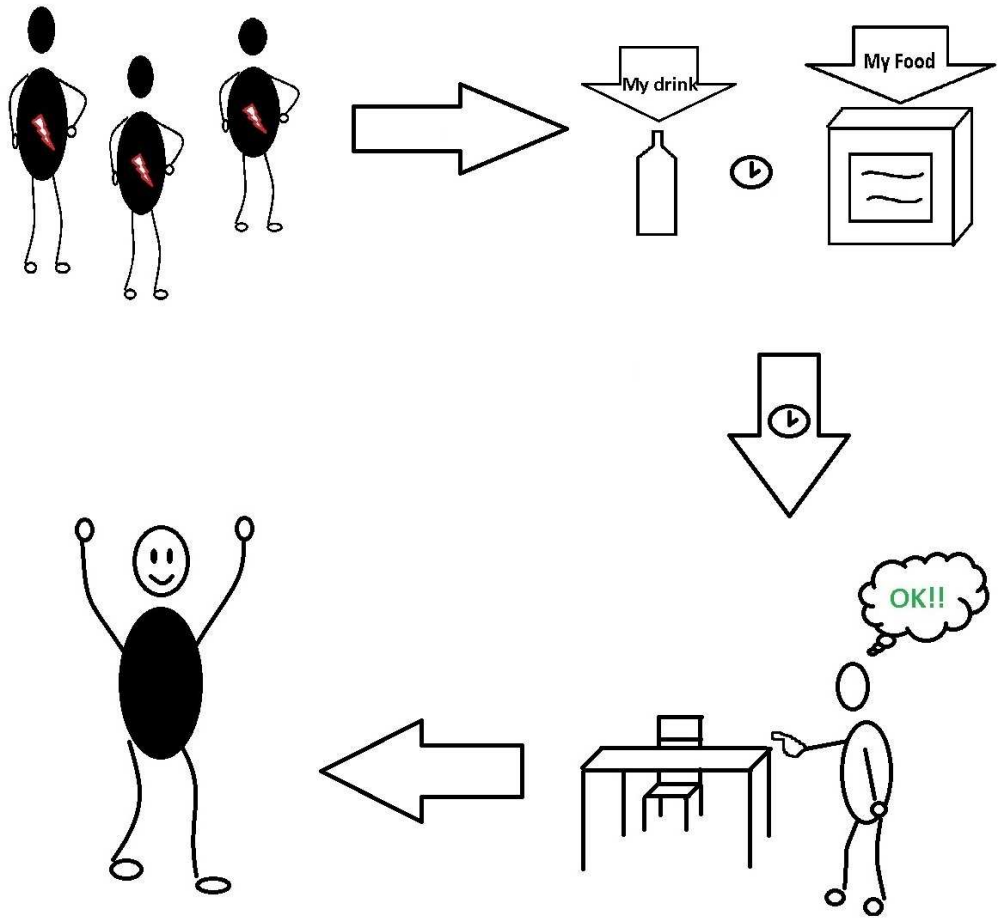
We have reduced the control-state reachability problem for ν-*RN* systems (undecidable) to control-state reachability for Timed ν-Petri Nets. Therefore:

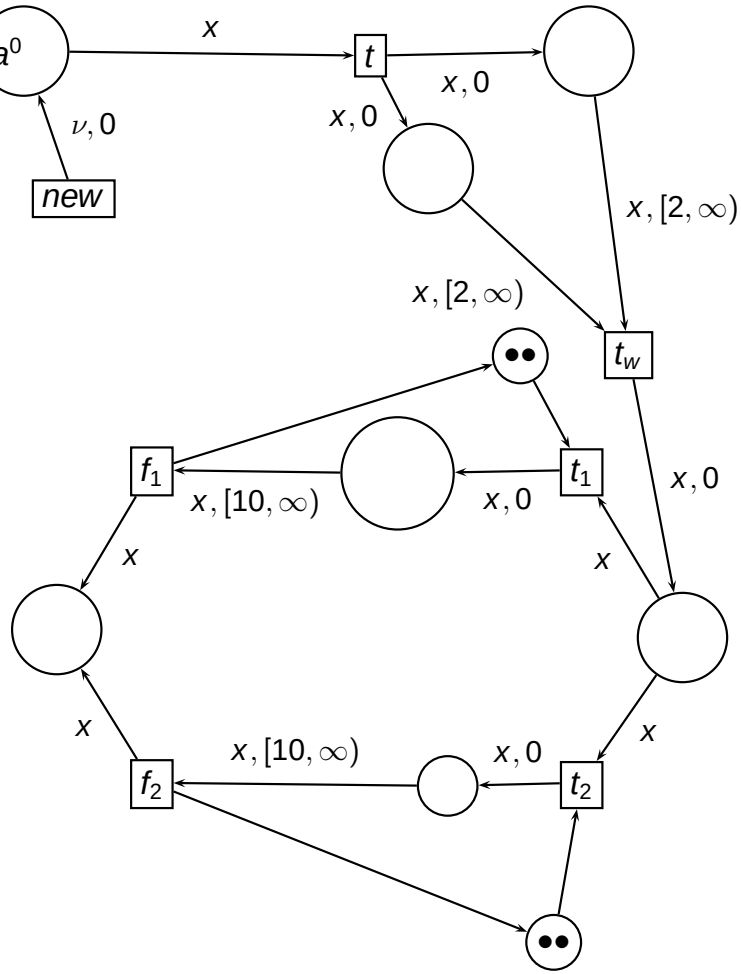
Proposition

Control-state reachability is undecidable for Timed ν-Petri Nets.

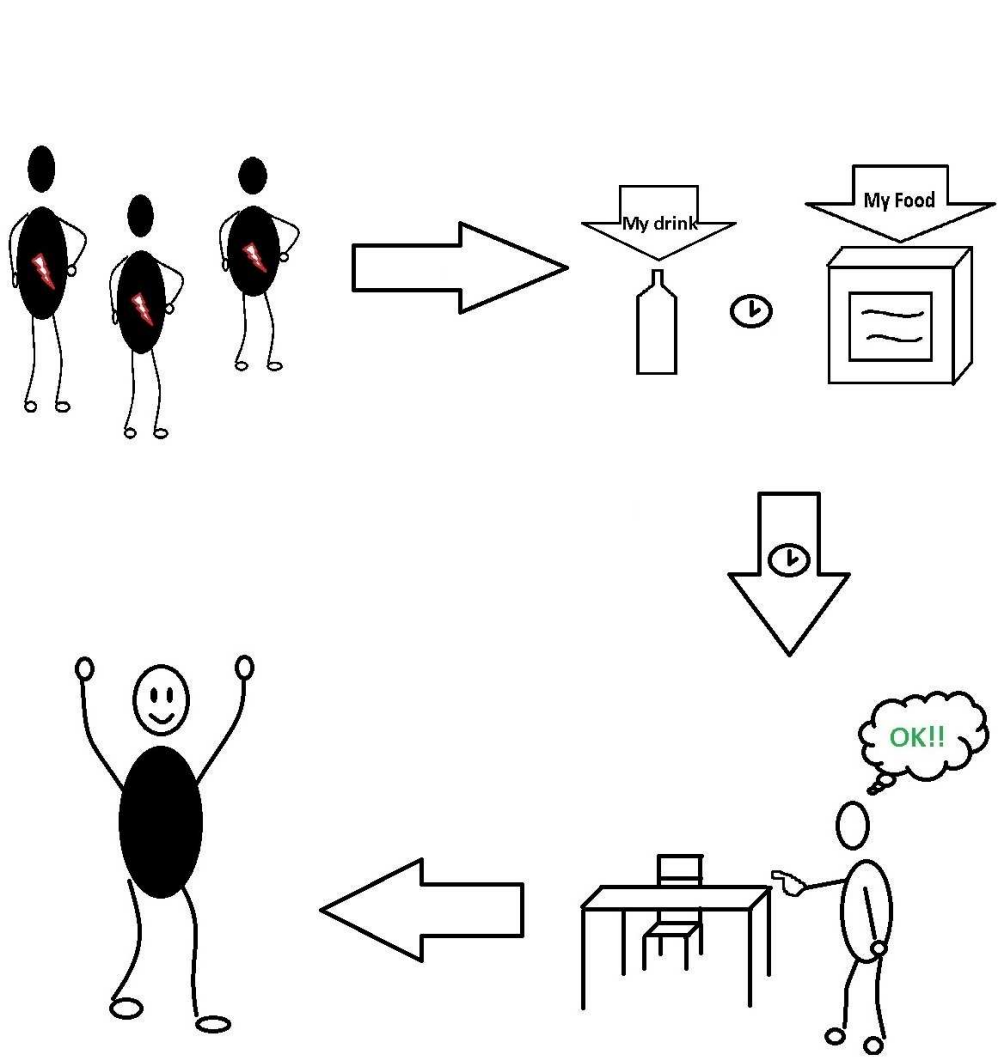
Outline

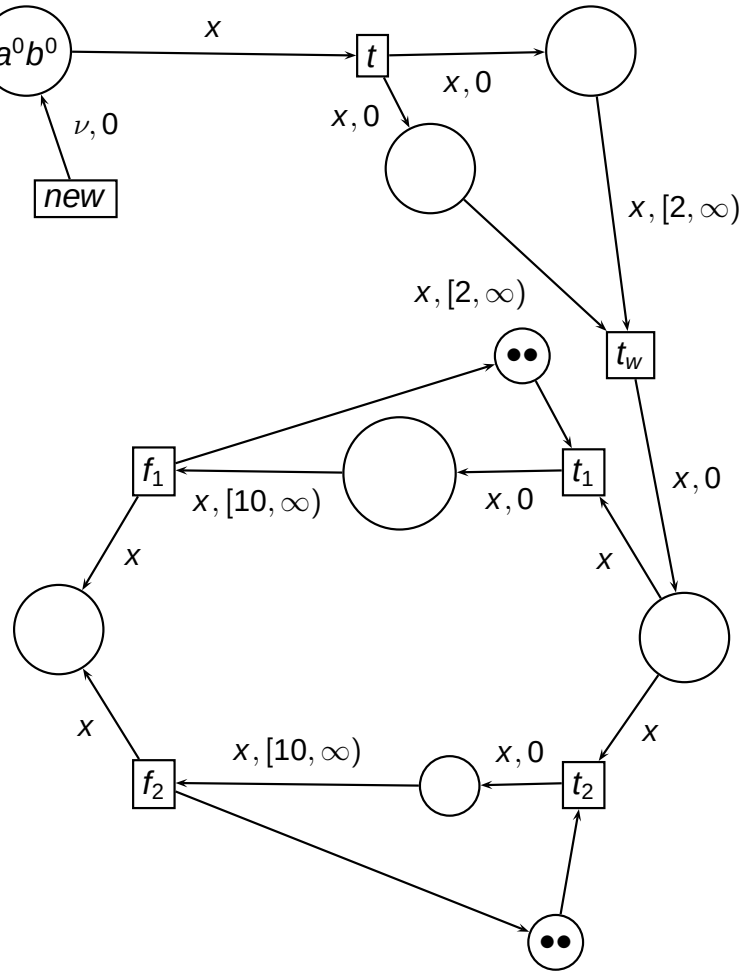
Locally-timed ν-PN



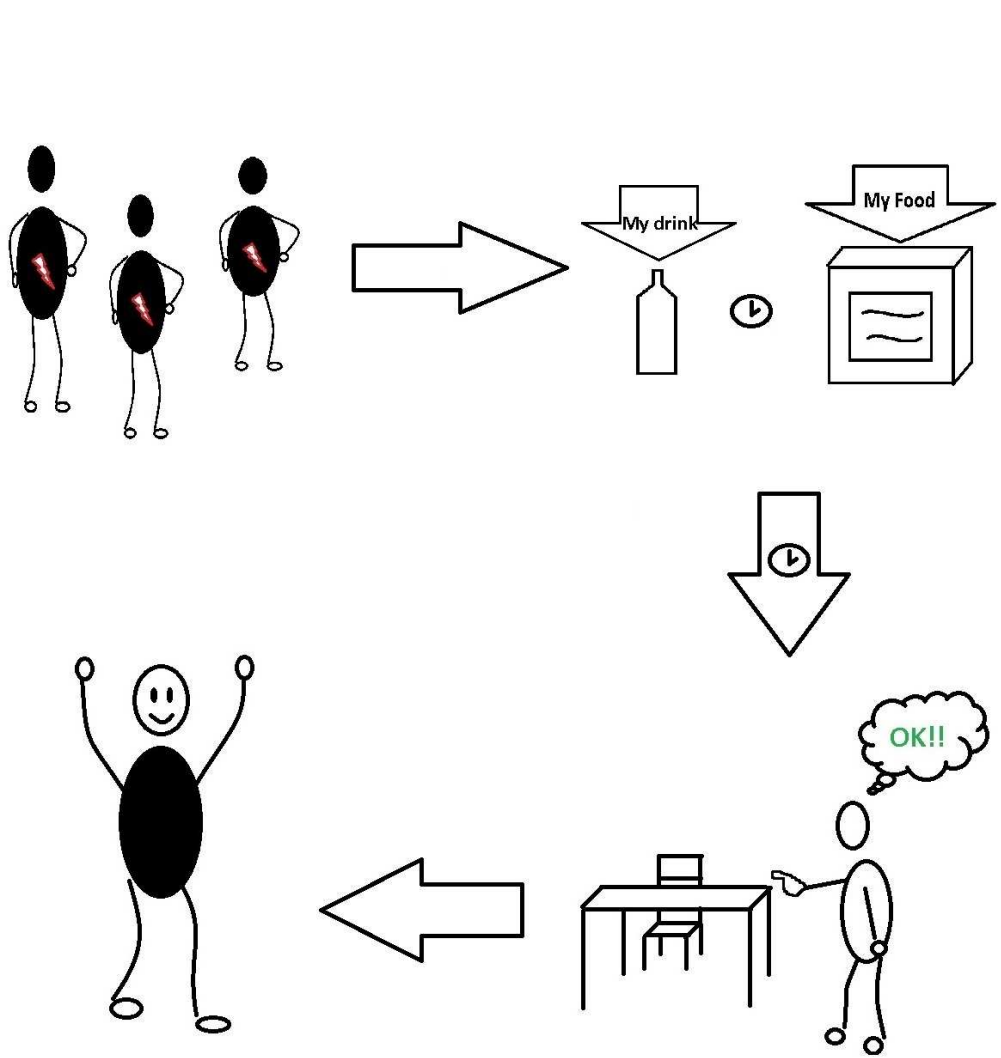


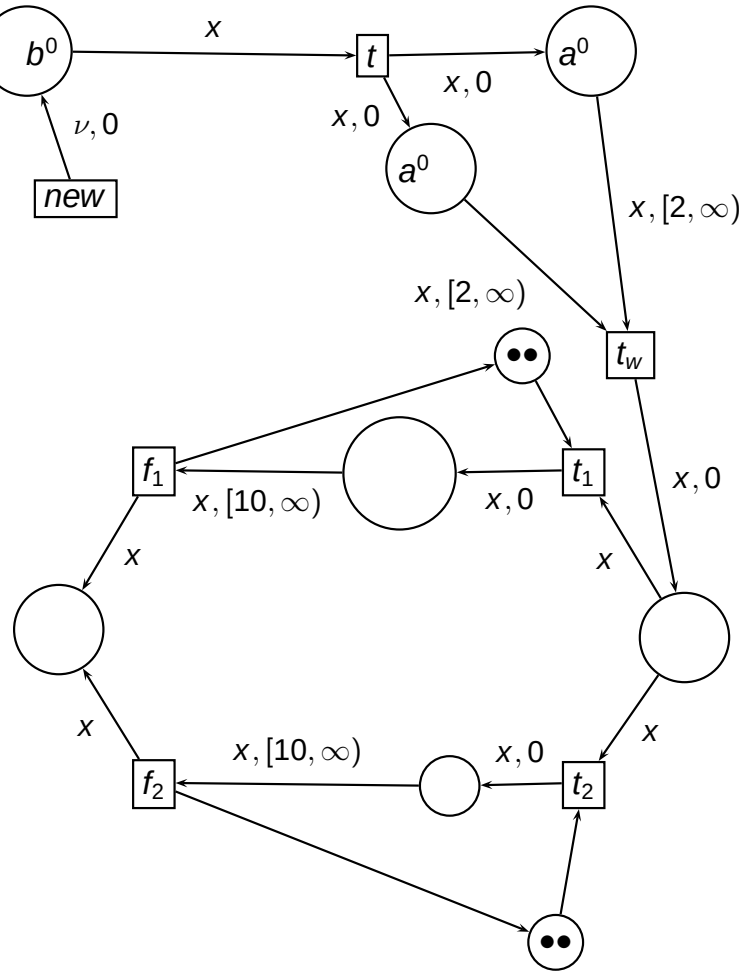
Locally-timed ν-PN



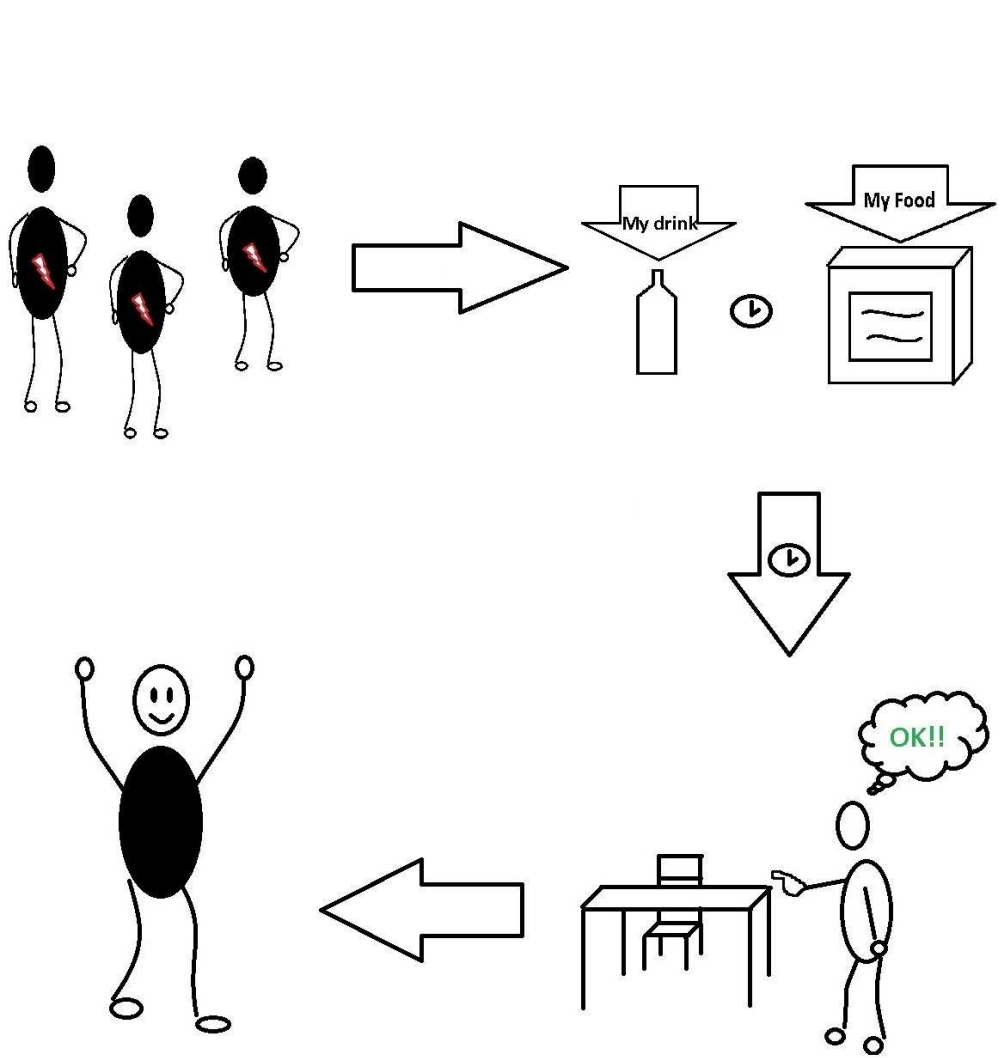


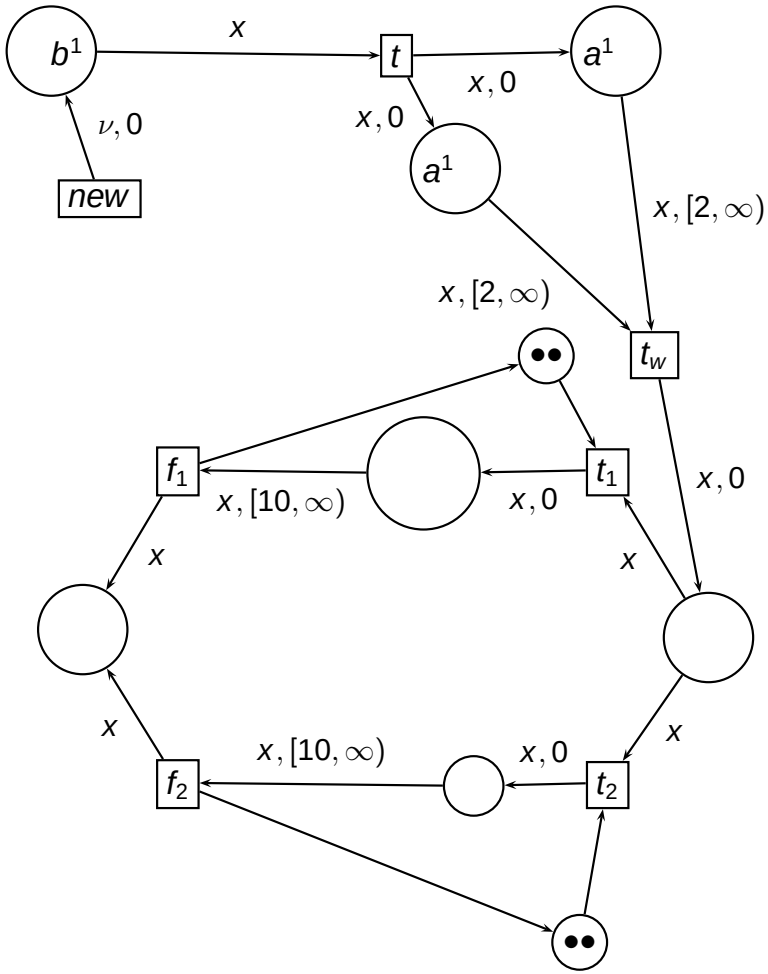
Locally-timed ν-PN



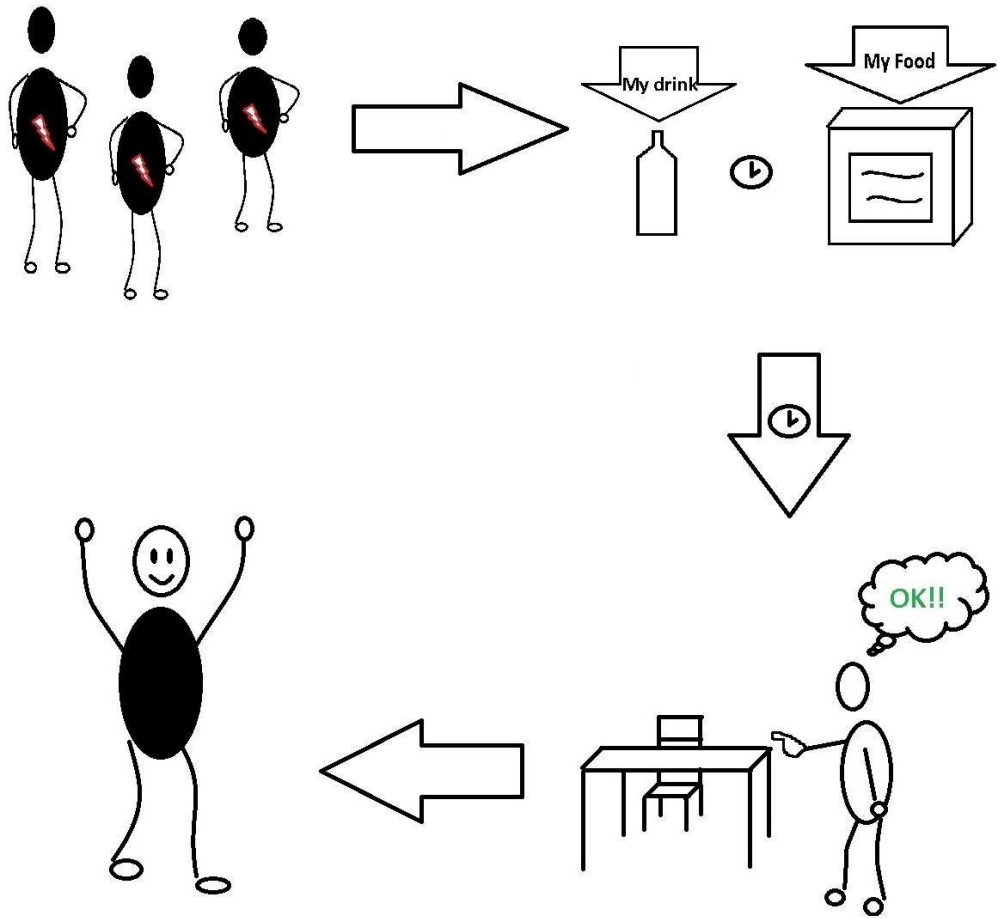


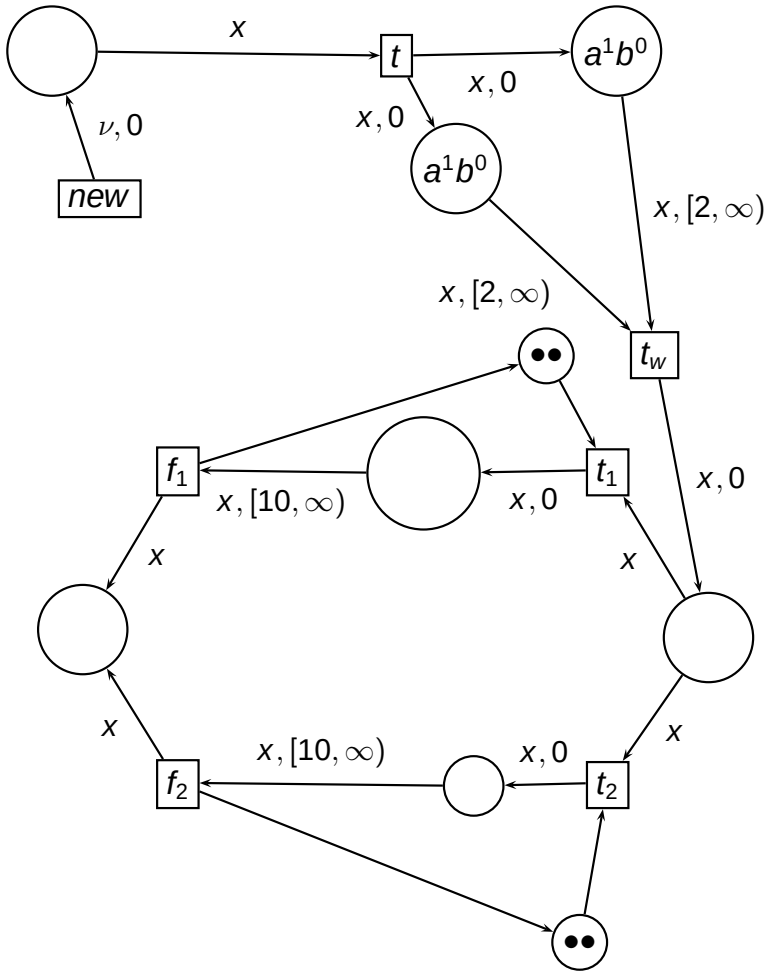
Locally-timed ν-PN



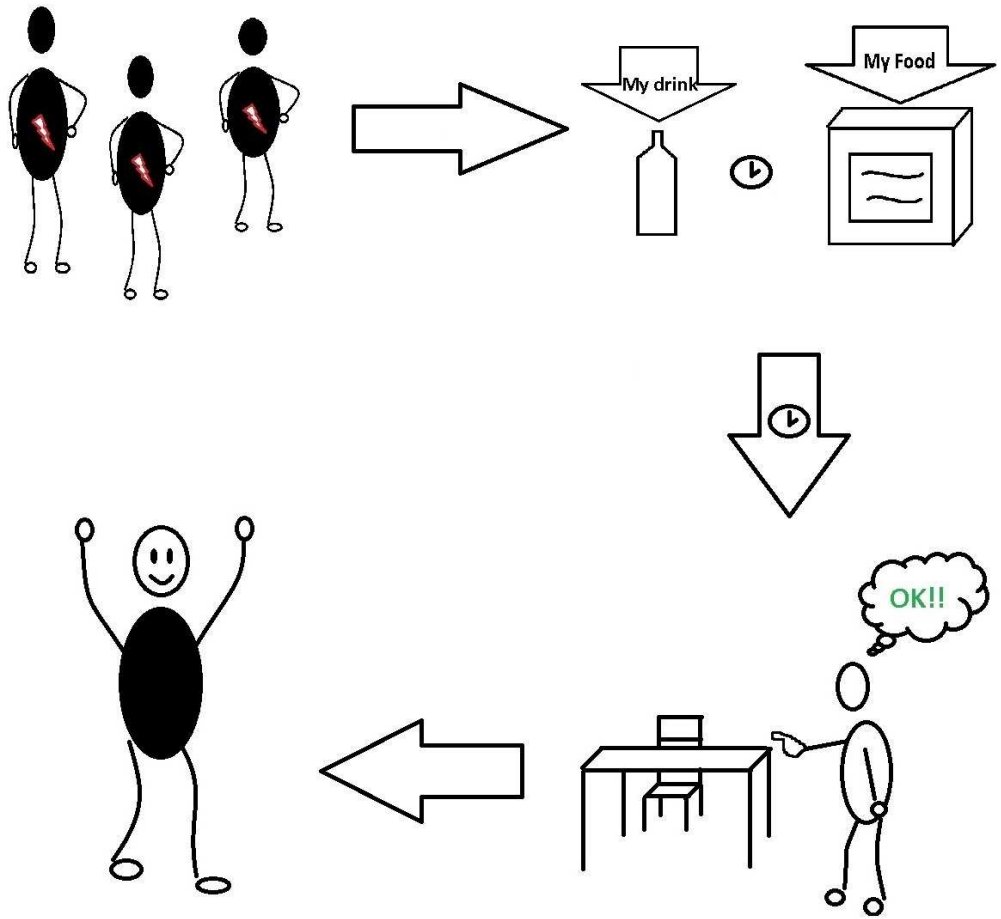


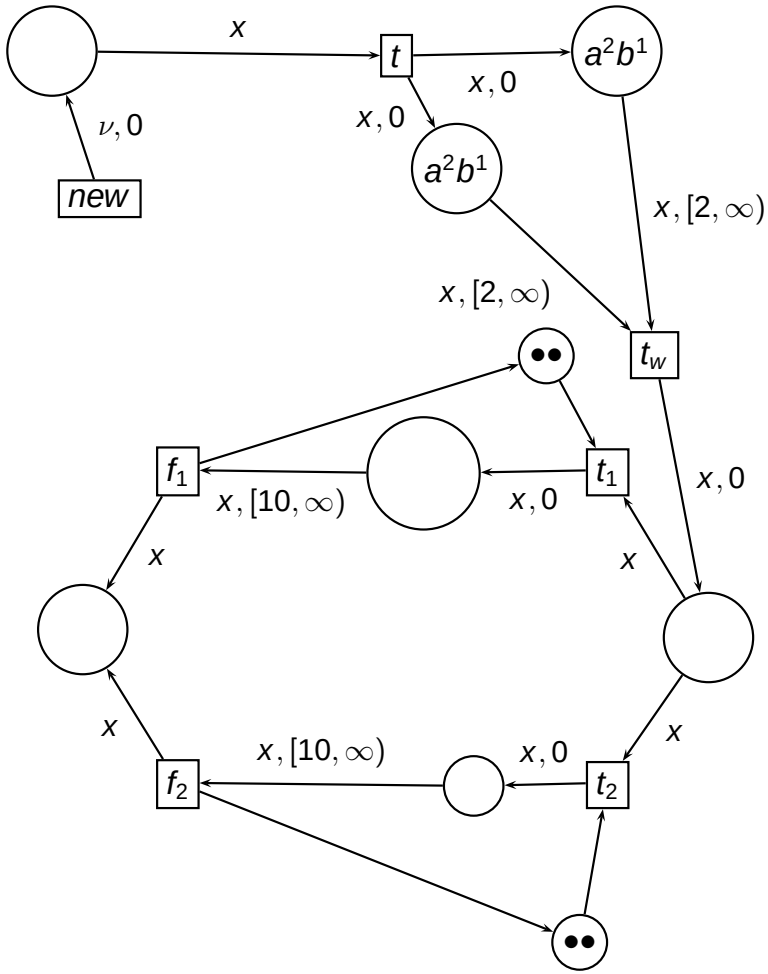
Locally-timed ν-PN



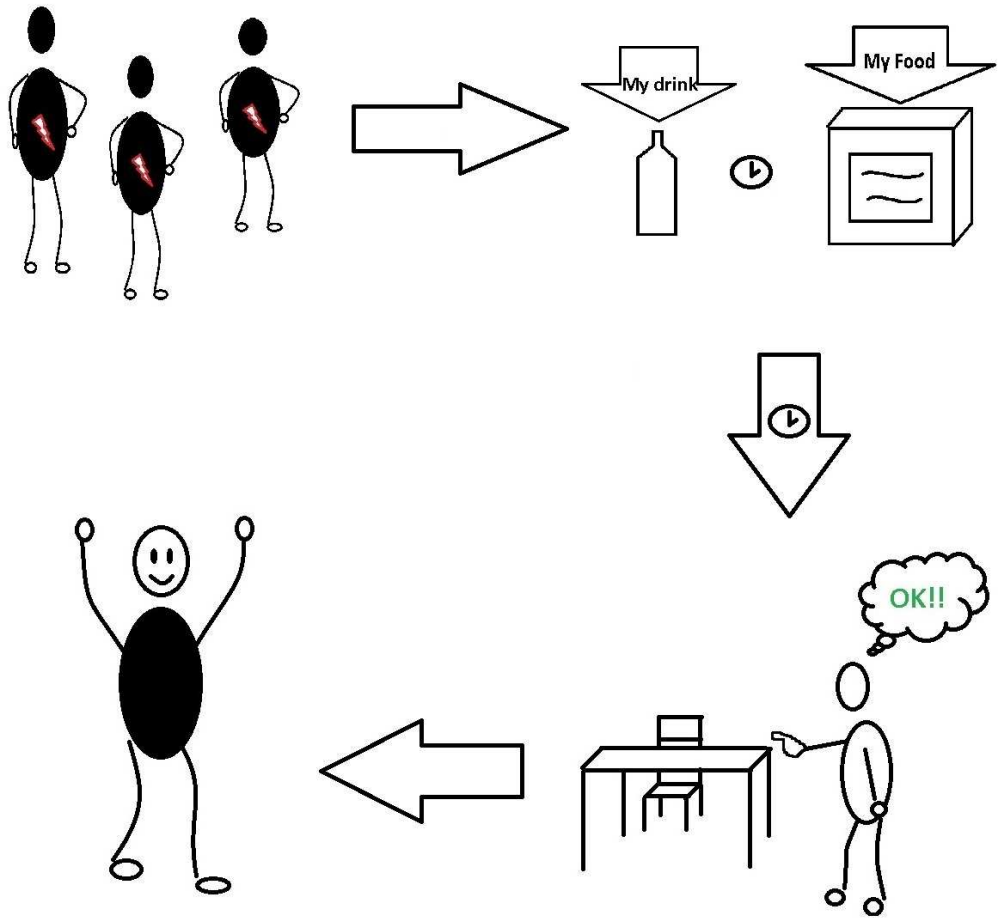


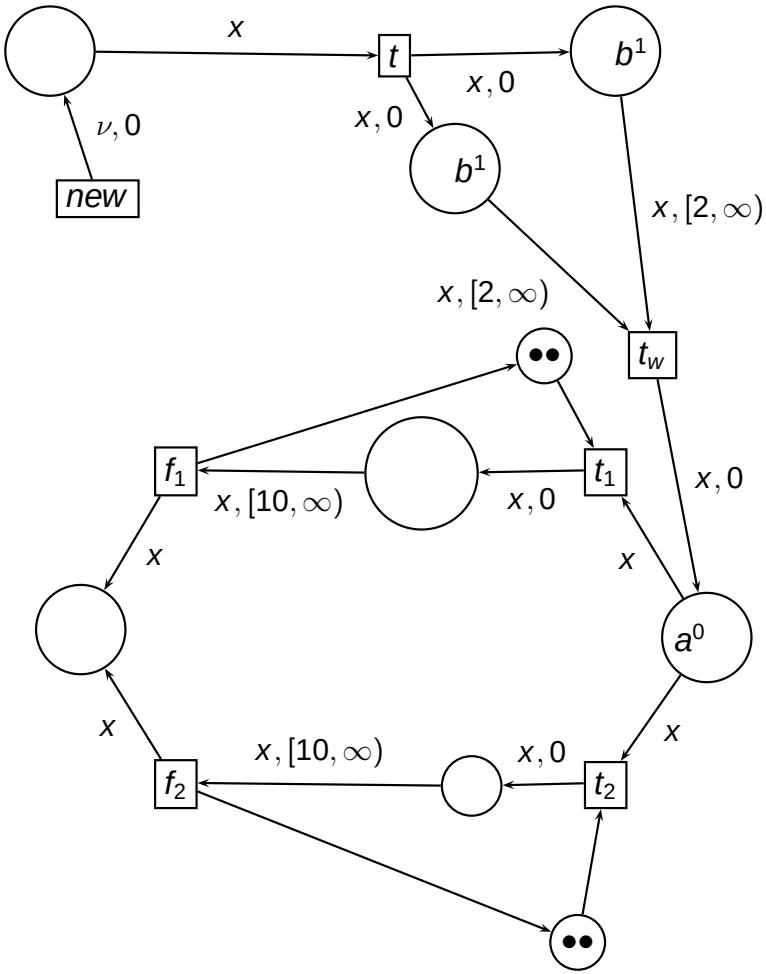
Locally-timed ν-PN



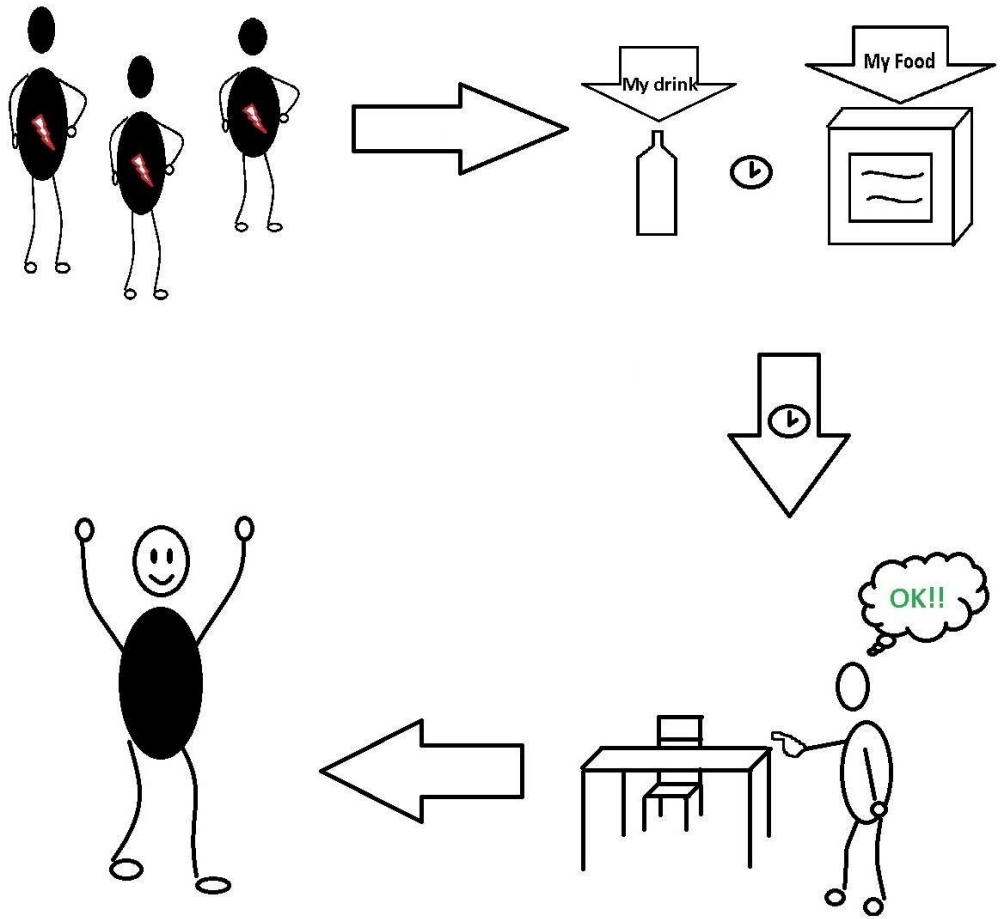


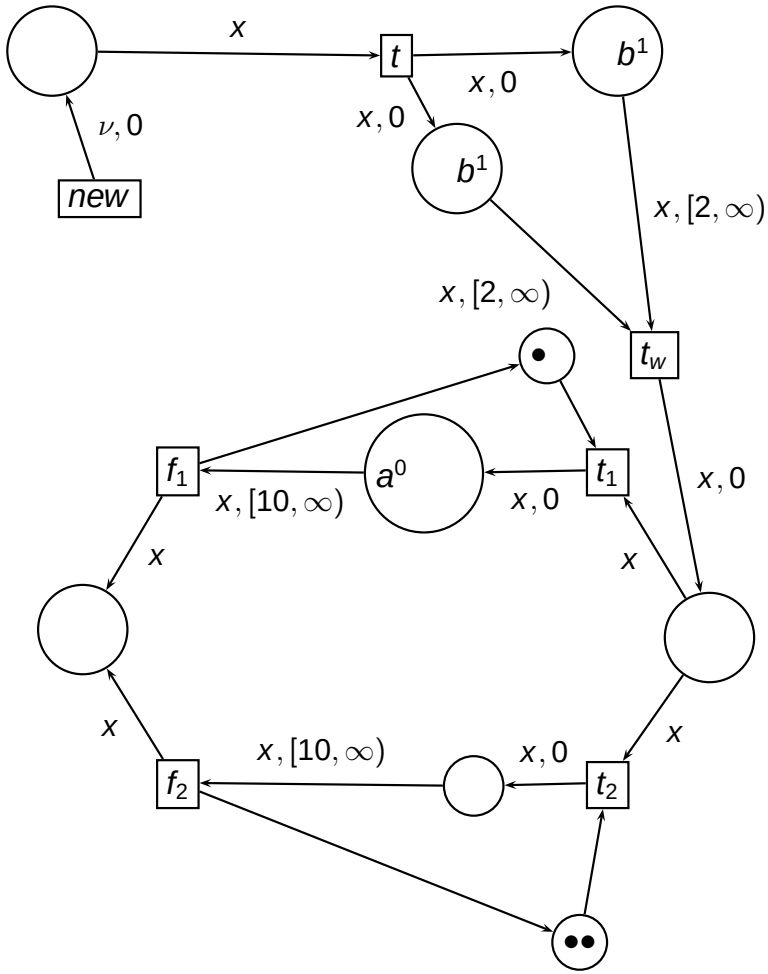
Locally-timed ν-PN



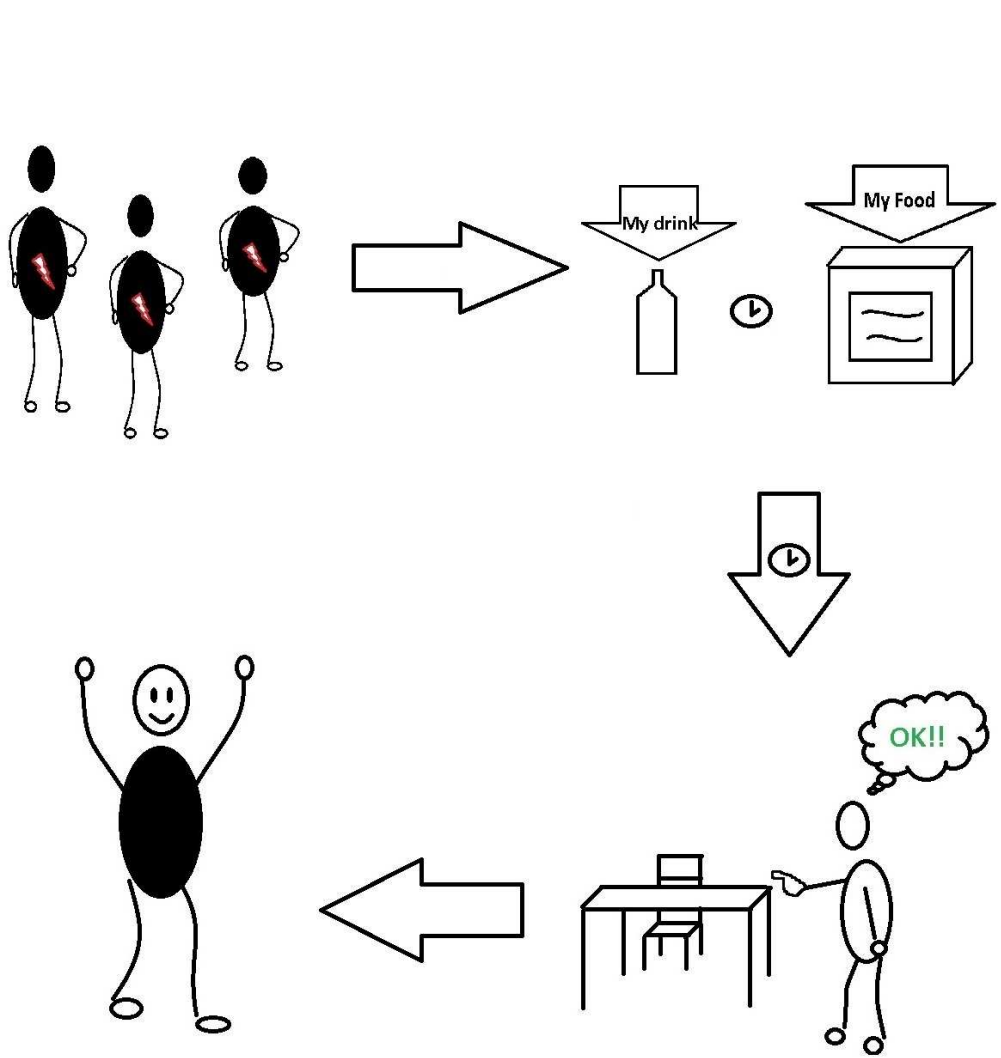


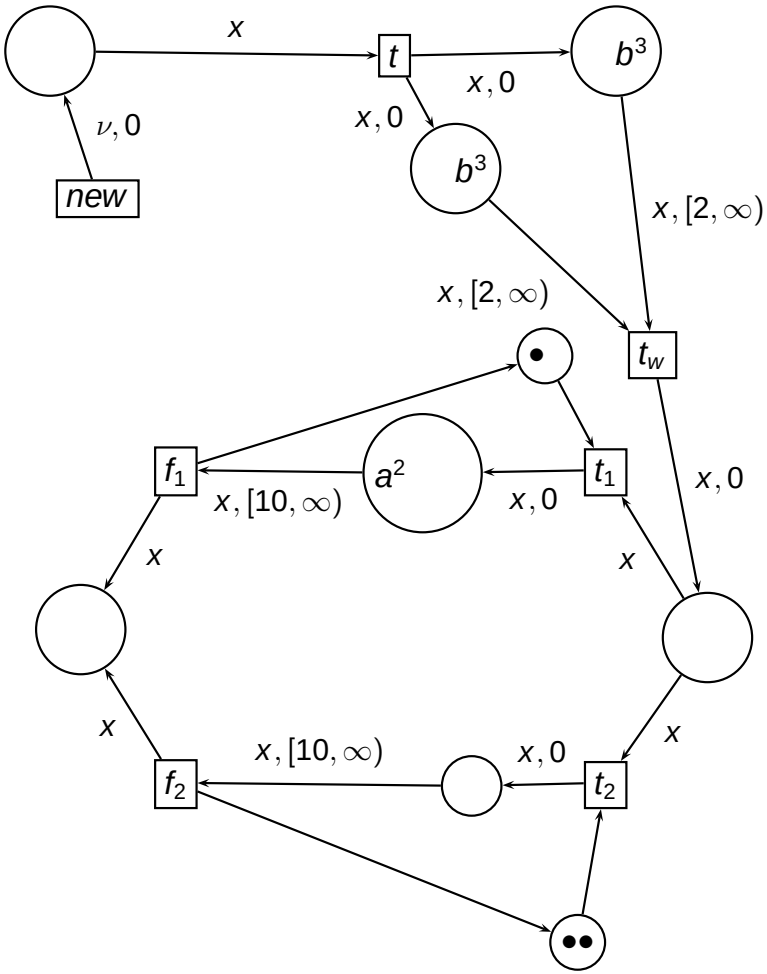
Locally-timed ν-PN



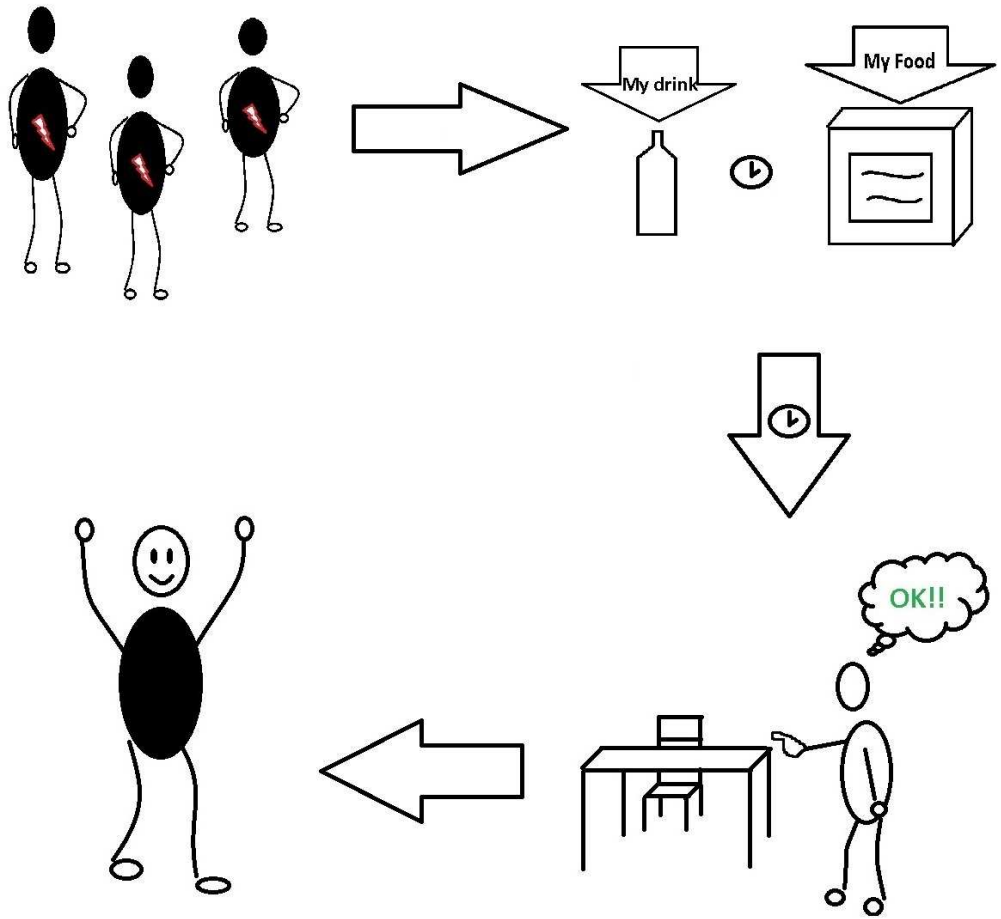


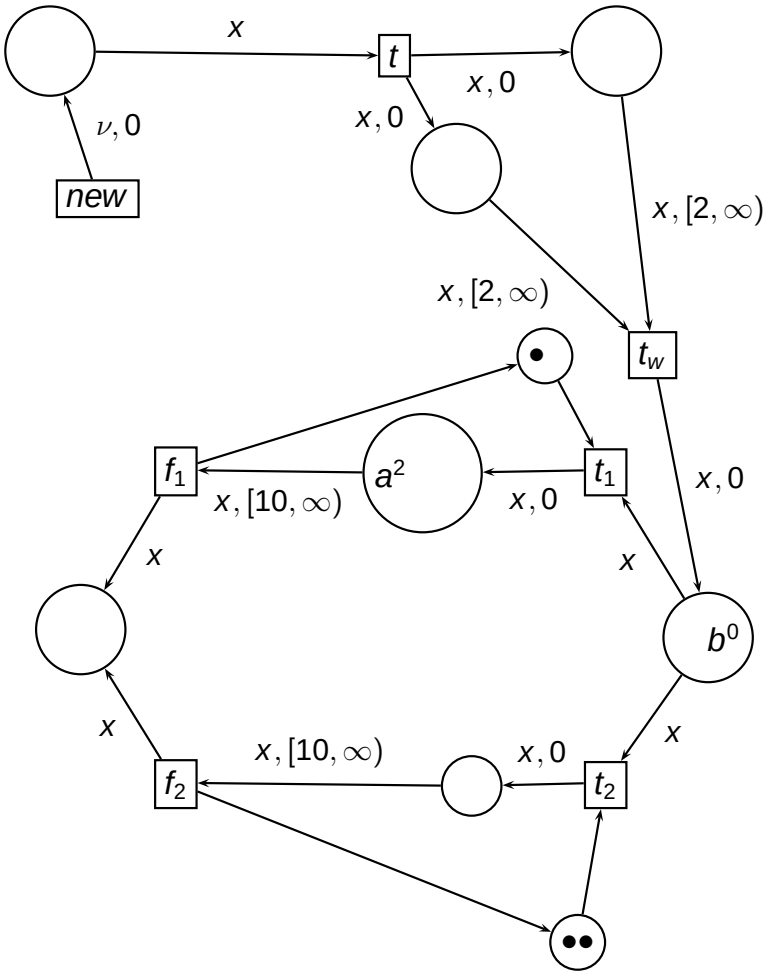
Locally-timed ν-PN



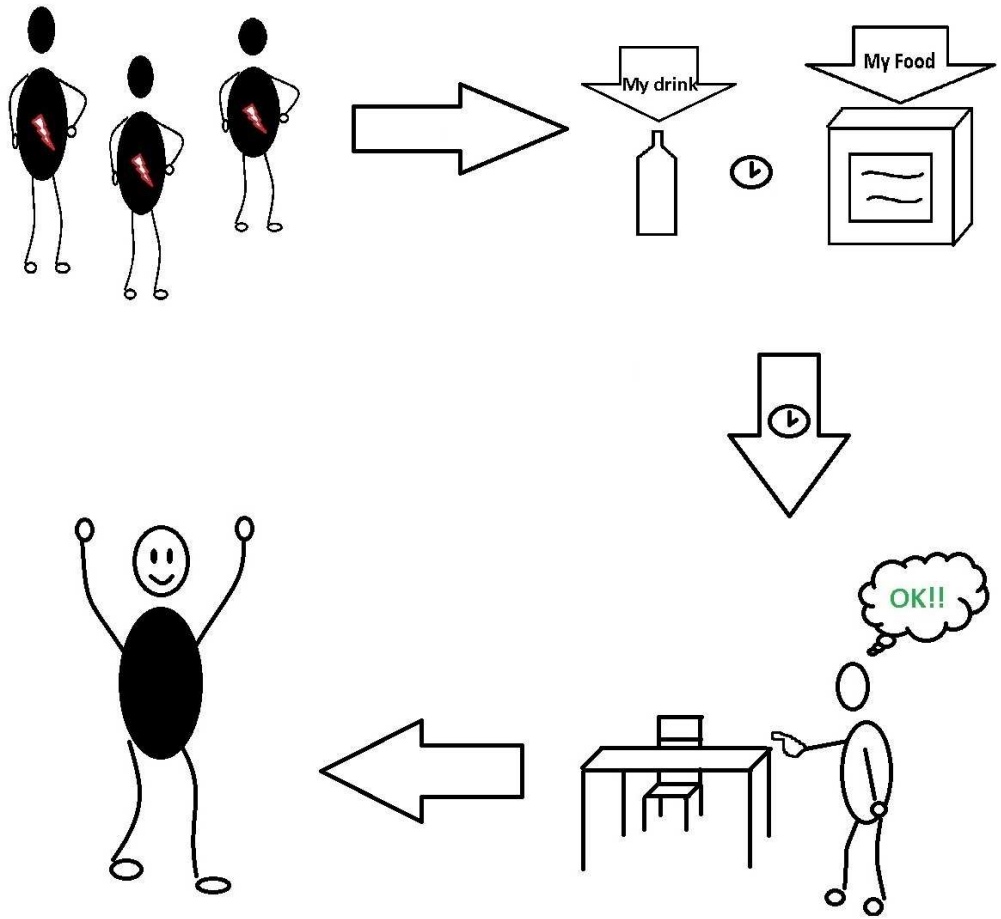


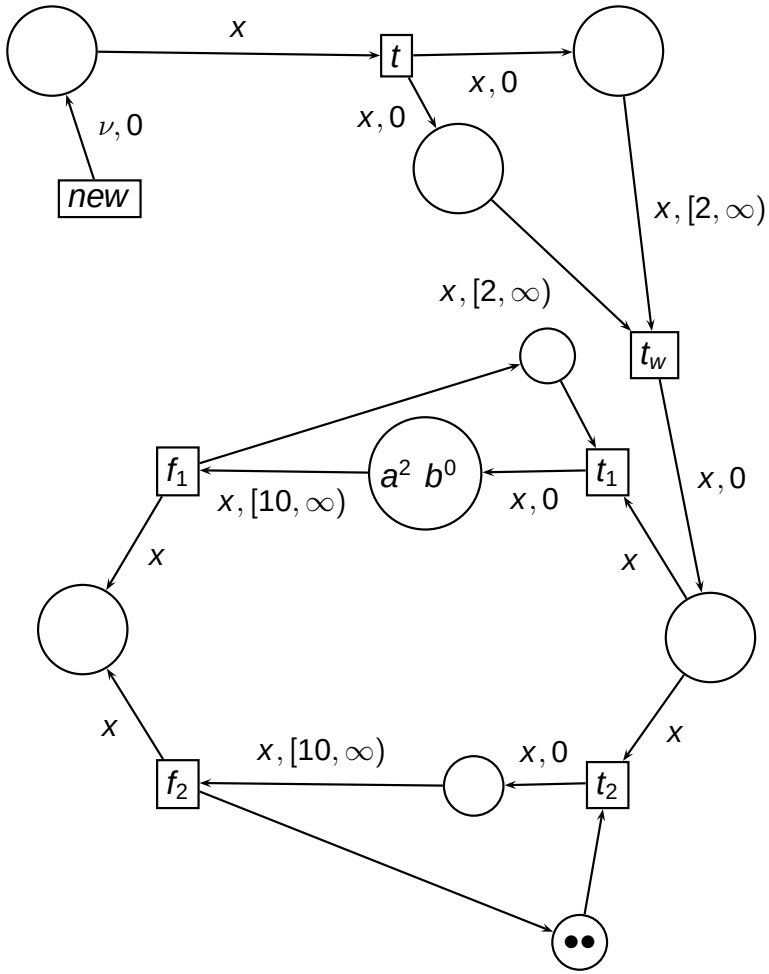
Locally-timed ν-PN



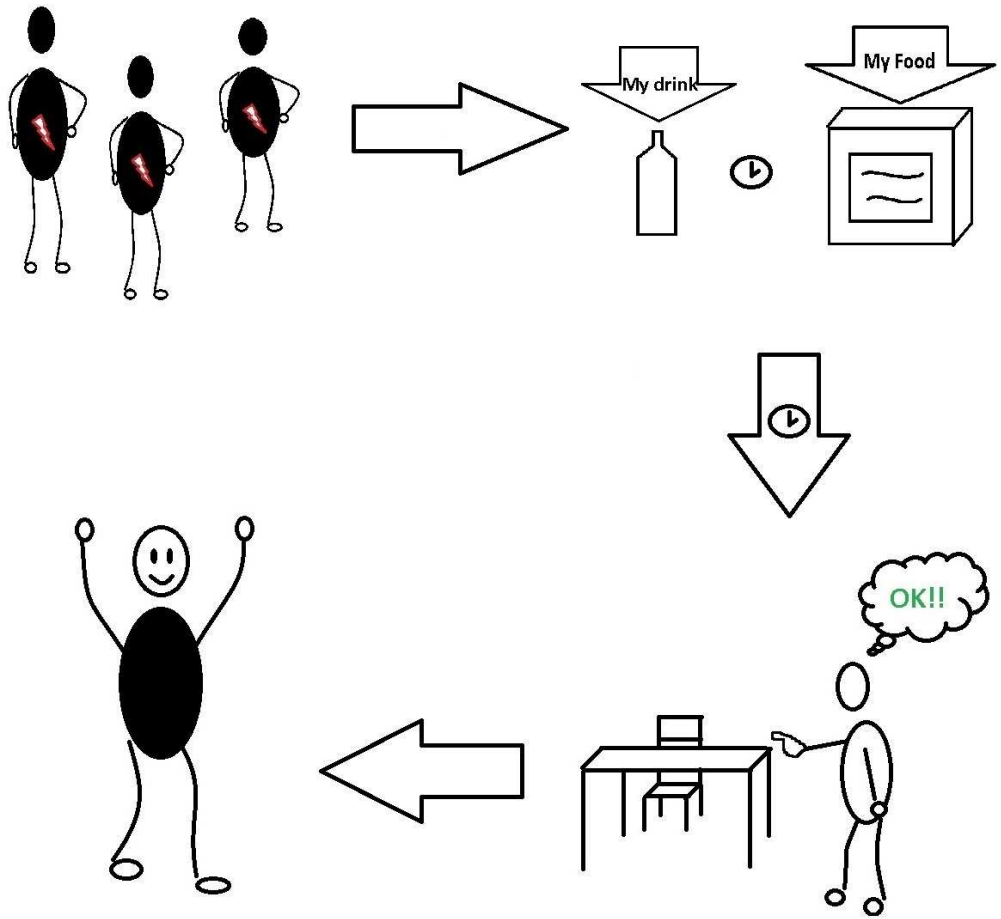


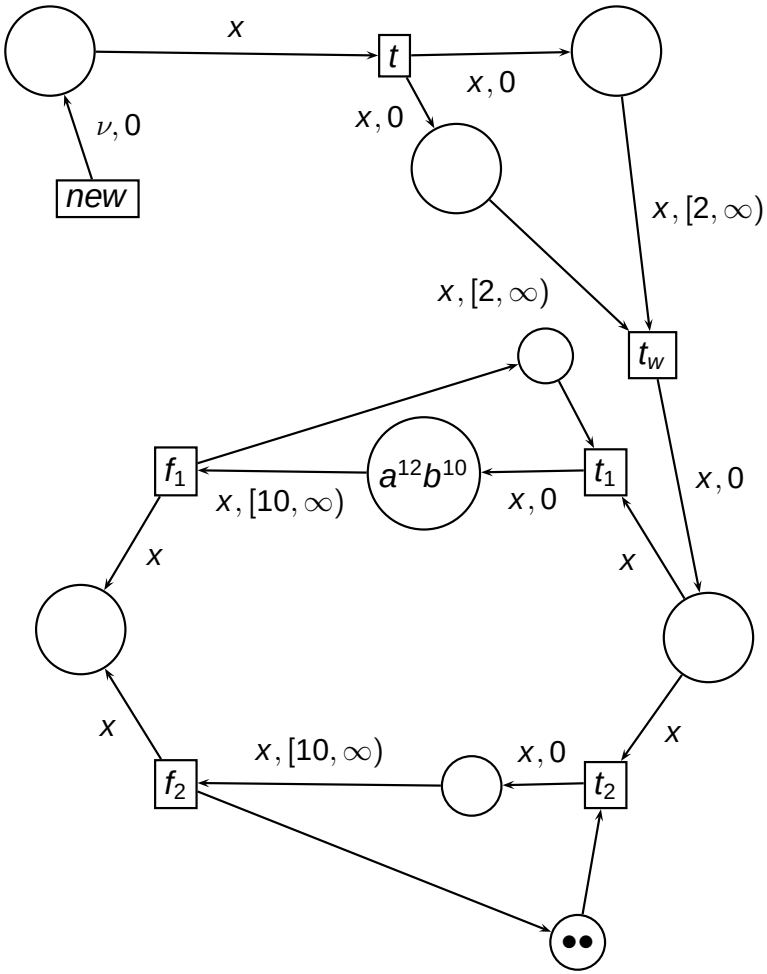
Locally-timed ν-PN



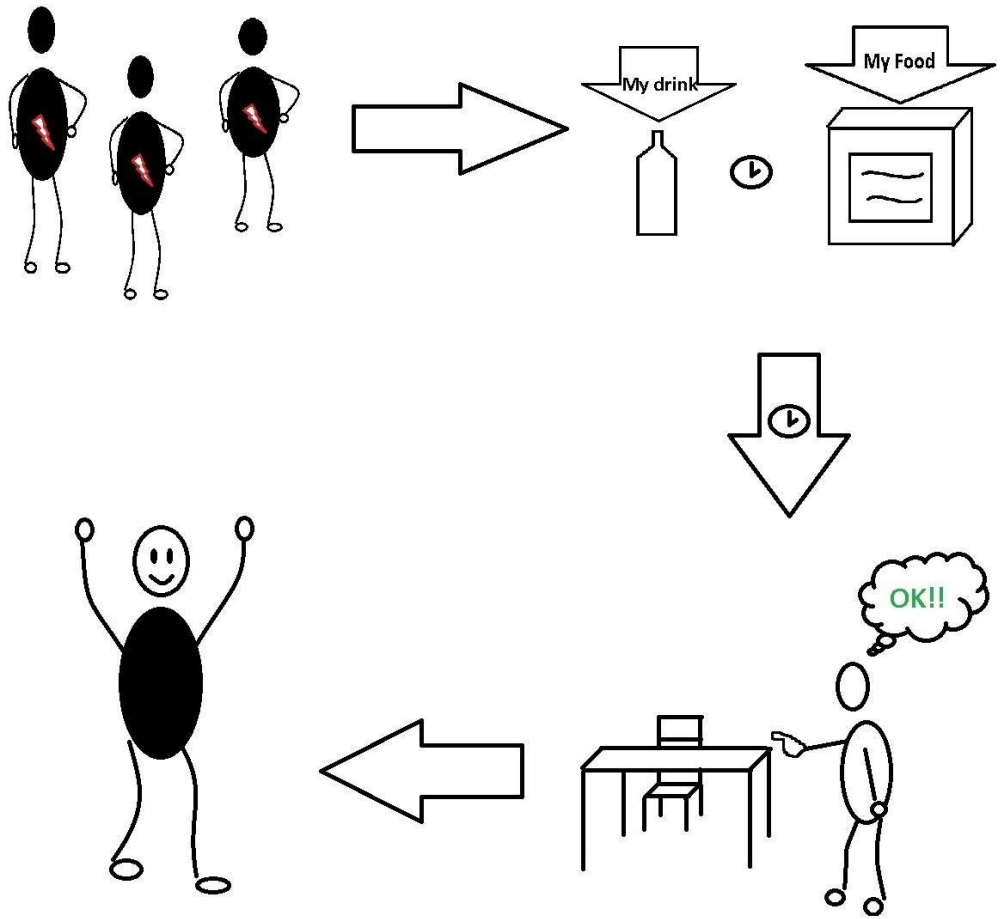


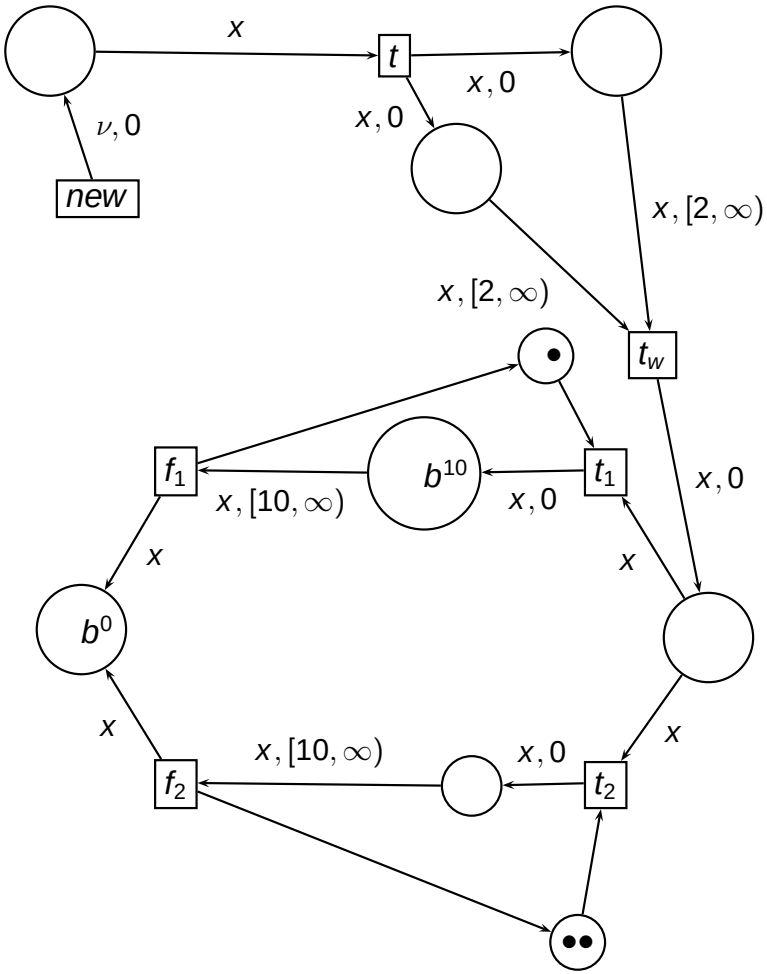
Locally-timed ν-PN



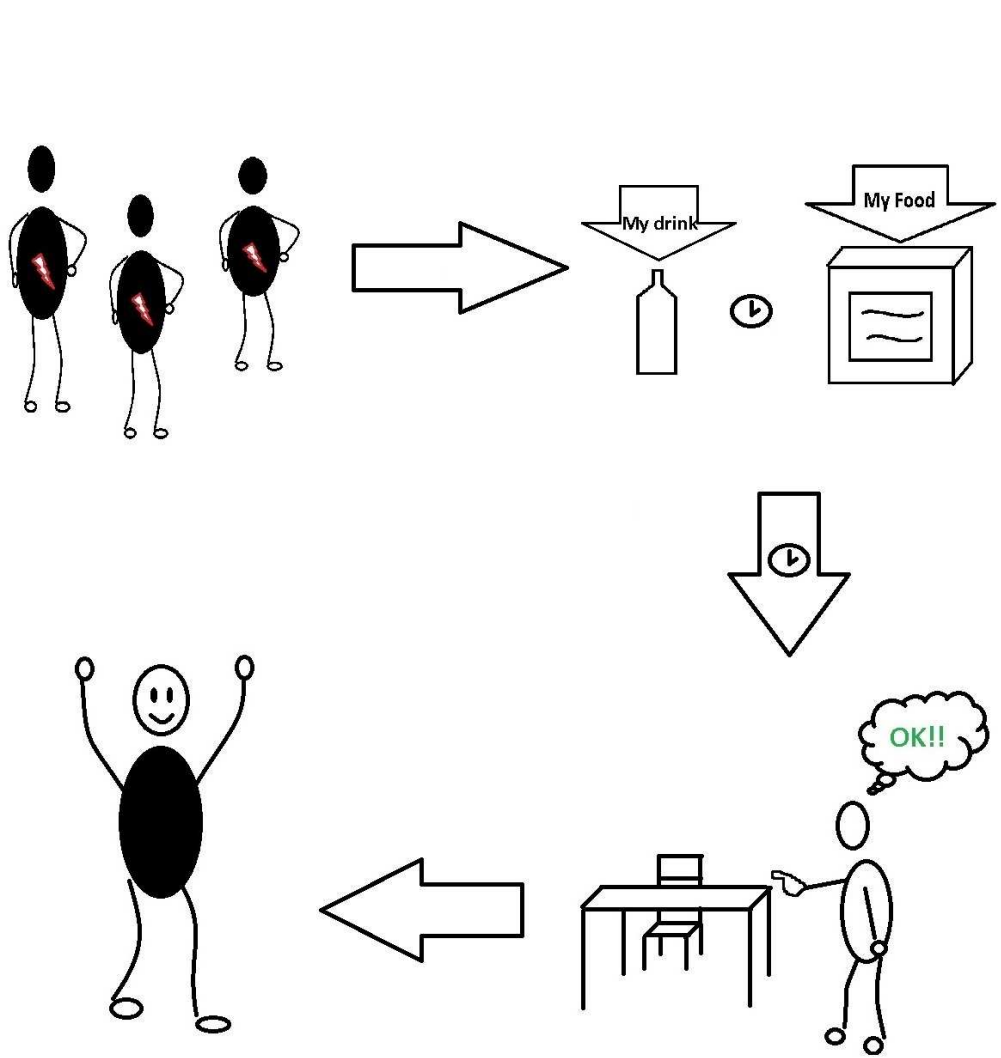


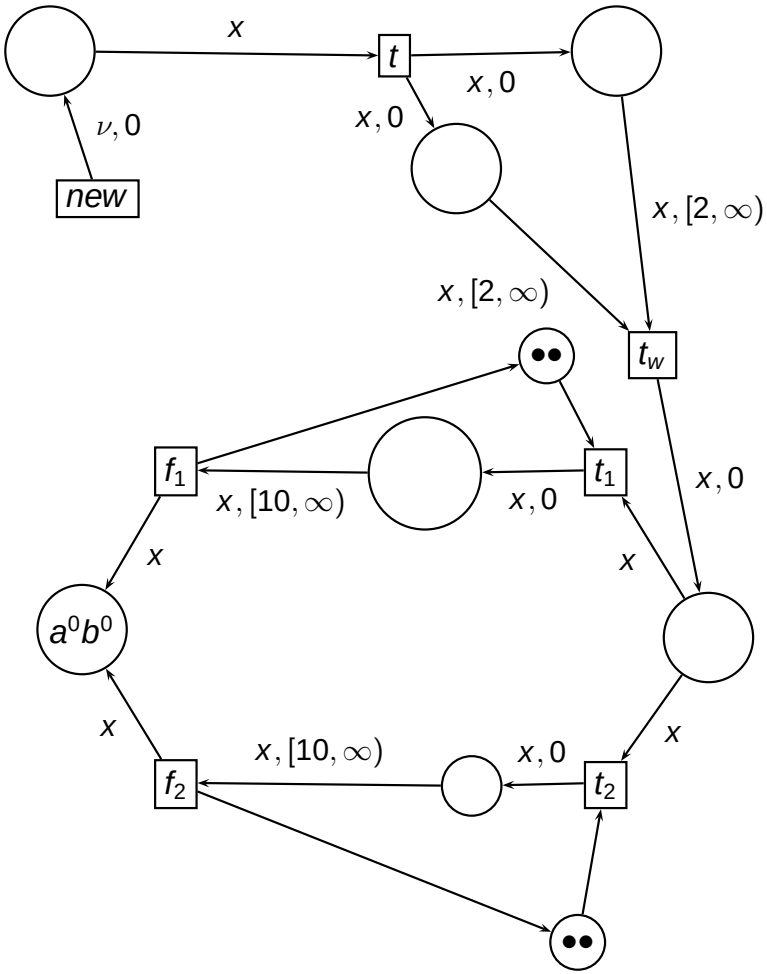
Locally-timed ν-PN





Locally-timed ν-PN



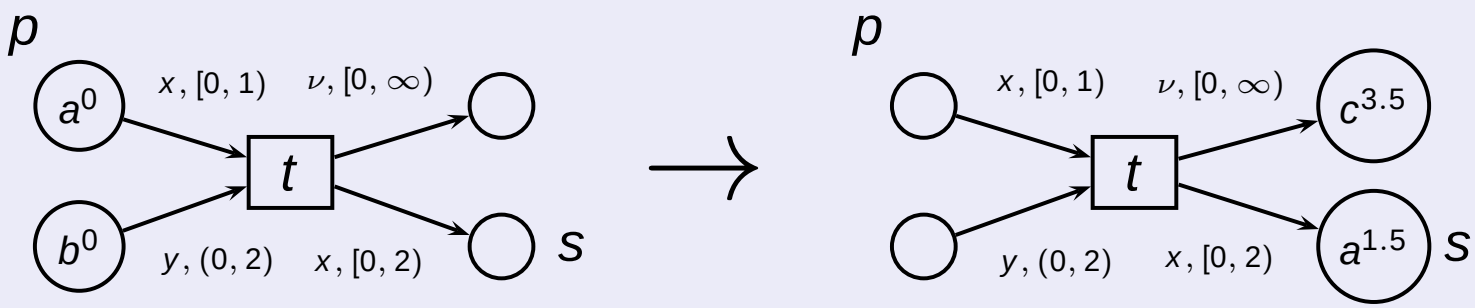


Locally-timed ν-*PN*: one clock per name

Definition

A *Locally-timed* ν-*PN* (ν-*lTdPN*) is a tuple $N = \langle P, T, F, H, \mathcal{G}, \lambda, \Sigma \rangle$, where:

- P and T are finite disjoint sets,
- for $t \in T$, $F_t, H_t : Var \to P^{\oplus}$ are the input and output functions of t,
- for $t \in T$, $\mathcal{G}_t : Var \to \mathcal{I} \times \mathcal{I}$ is the time constraints function of t.



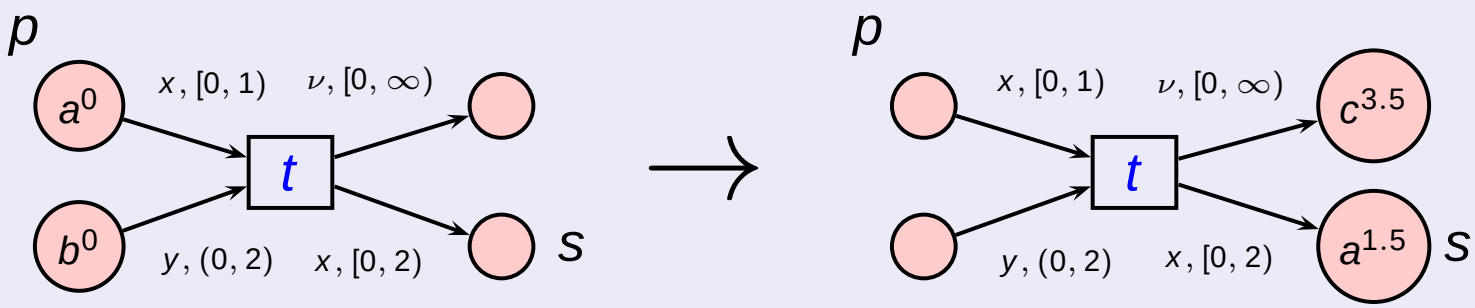
$F_t(x) = \{p\}$, $H_t(x) = \{s\}$, $\mathcal{G}_t(x) = ([0, 1), [0, 2))$.

Locally-timed ν-*PN*: one clock per name

Definition

A *Locally-timed* ν-*PN* (ν-*lTdPN*) is a tuple $N = \langle P, T, F, H, \mathcal{G}, \lambda, \Sigma \rangle$, where:

- P and T are finite disjoint sets,
- for $t \in T$, $F_t, H_t : Var \to P^{\oplus}$ are the input and output functions of t,
- for $t \in T$, $\mathcal{G}_t : Var \to \mathcal{I} \times \mathcal{I}$ is the time constraints function of t.



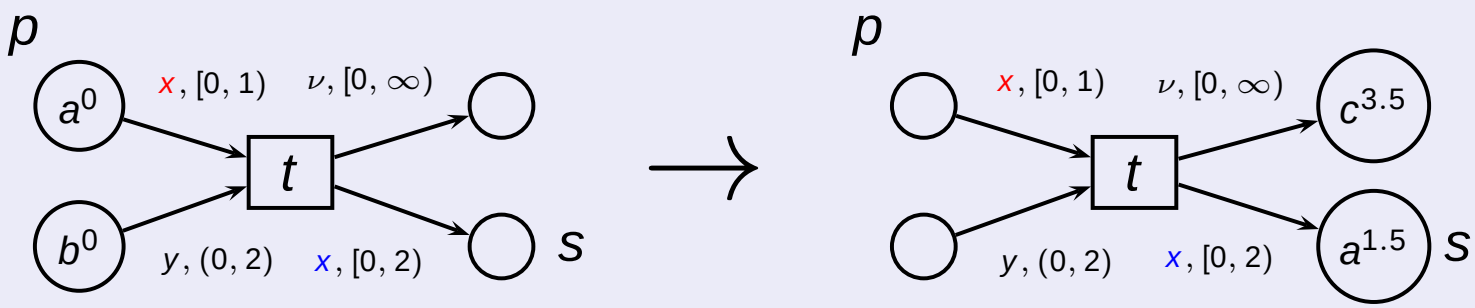
$F_t(x) = \{p\}$, $H_t(x) = \{s\}$, $\mathcal{G}_t(x) = ([0, 1), [0, 2))$.

Locally-timed ν-*PN*: one clock per name

Definition

A *Locally-timed* ν-*PN* (ν-*lTdPN*) is a tuple $N = \langle P, T, F, H, \mathcal{G}, \lambda, \Sigma \rangle$, where:

- P and T are finite disjoint sets,
- for $t \in T$, $F_t, H_t : Var \to P^{\oplus}$ are the input and output functions of t,
- for $t \in T$, $\mathcal{G}_t : Var \to \mathcal{I} \times \mathcal{I}$ is the time constraints function of t.



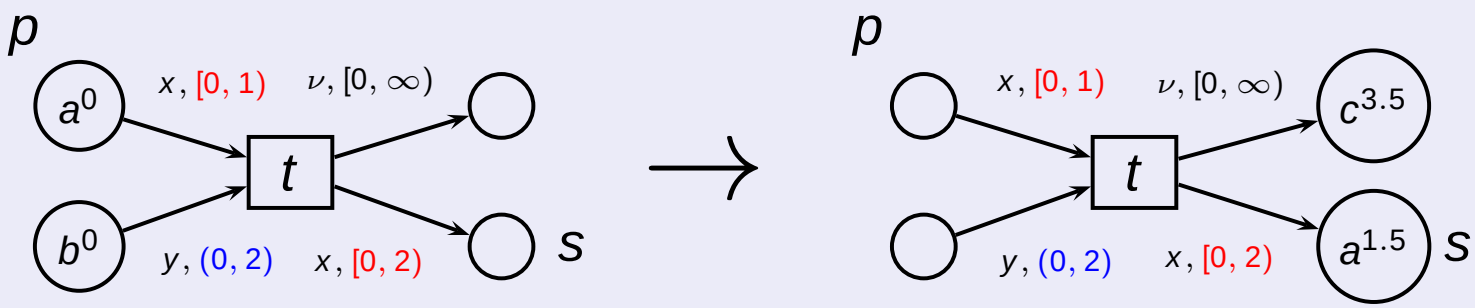
$F_t(x) = \{p\}$, $H_t(x) = \{s\}$, $\mathcal{G}_t(x) = ([0, 1), [0, 2))$.

Locally-timed ν-*PN*: one clock per name

Definition

A *Locally-timed* ν-*PN* (ν-*lTdPN*) is a tuple $N = \langle P, T, F, H, \mathcal{G}, \lambda, \Sigma \rangle$, where:

- P and T are finite disjoint sets,
- for $t \in T$, $F_t, H_t : Var \to P^{\oplus}$ are the input and output functions of t,
- for $t \in T$, $\mathcal{G}_t : Var \to \mathcal{I} \times \mathcal{I}$ is the time constraints function of t.



$F_t(x) = \{p\}$, $H_t(x) = \{s\}$, $\mathcal{G}_t(x) = ([0, 1), [0, 2))$.

Markings of a ν-lTdPN

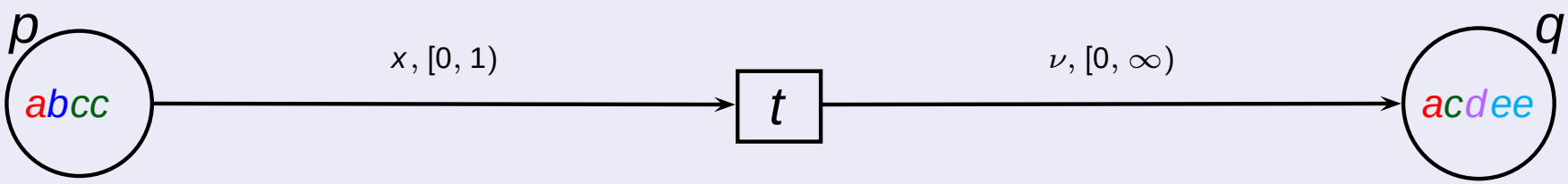
Marking

Expression of the form

$$a_1{:}(m_1, r_1), \ldots, a_n{:}(m_n, r_n)$$

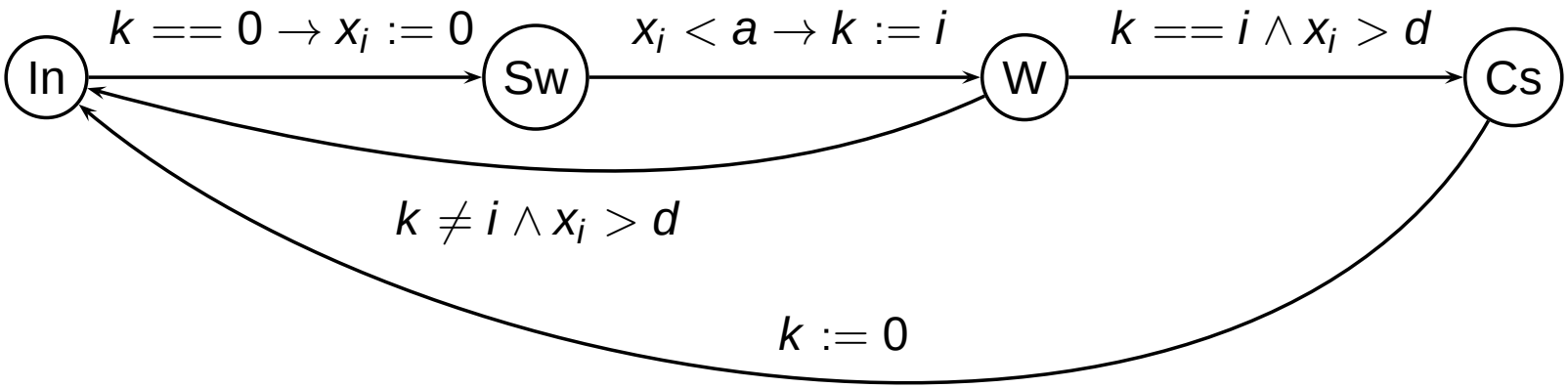
where $a_1, \ldots, a_n$ are pairwise different names, and for each $i \in n^+$, $m_i \in P^{\oplus}$ and $r_i \in \mathbb{R}_{\geq 0}$.

Example



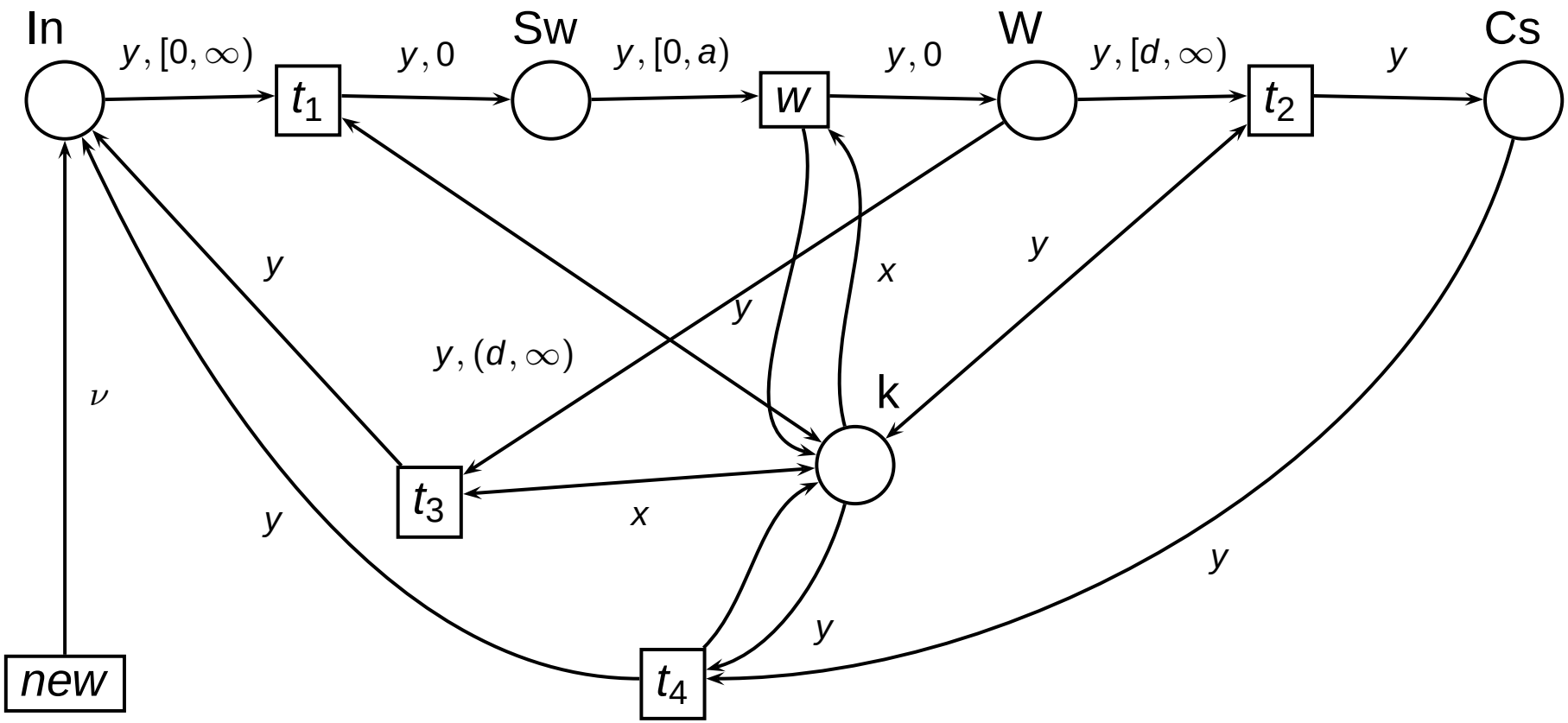
$a : (\{pq\}, 1.5)$, $b : (\{p\}, 2)$, $c : (\{ppq\}, 4.5)$, $d : (\{q\}, 2.3)$, $e : (\{qq\}, 0.3)$.

Example: Parametric Fischer's mutual exclusion protocol

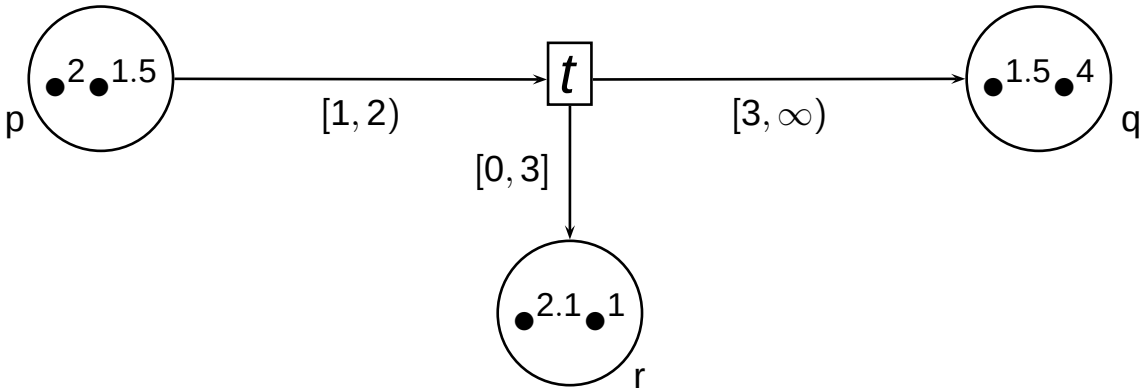



```
1  repeat
2    non critical section
3    repeat
4      await k==0;
5      k:=i;
6      delay(d);
7    until k==i;
8    critical section;
9    k:=0;
10   non critical section
11 until false;
```

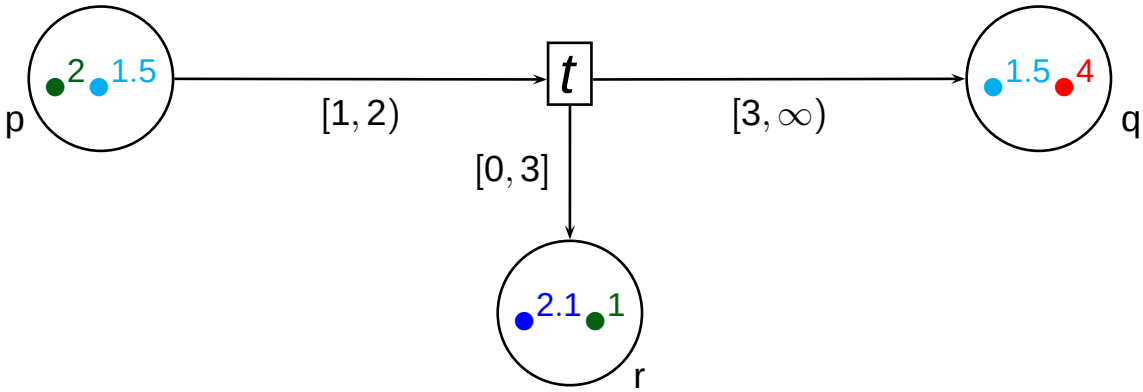
Example: Fischer's mutual exclusion protocol



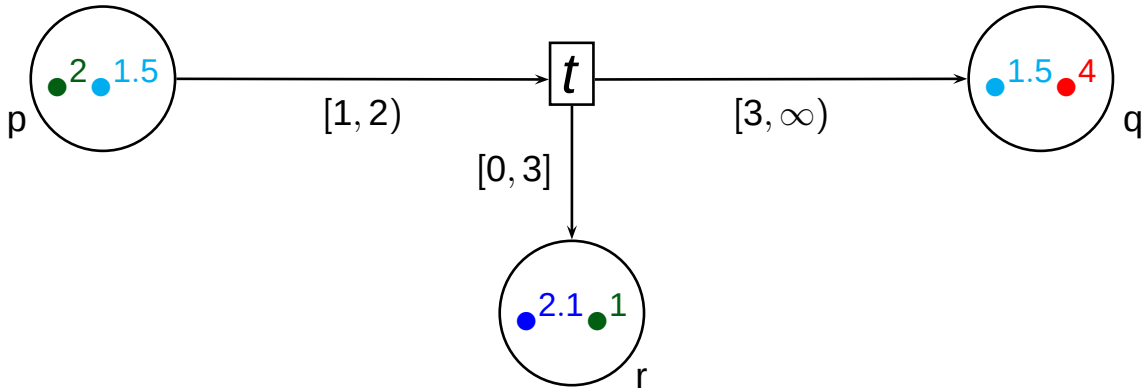
Regions for Td-PN

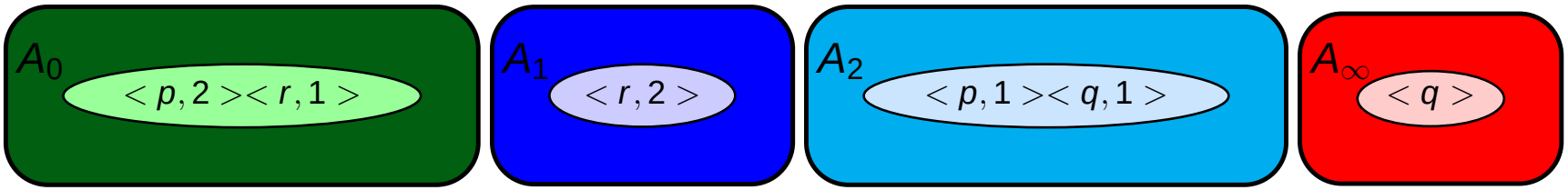


Regions for Td-PN

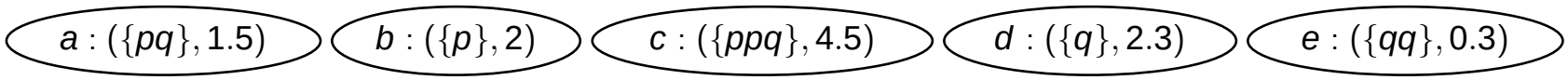


Regions for Td-PN

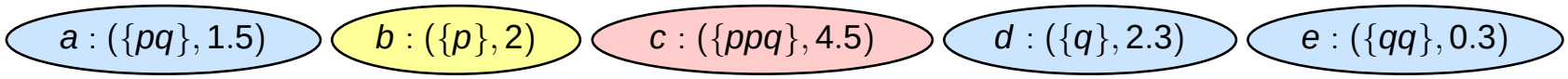




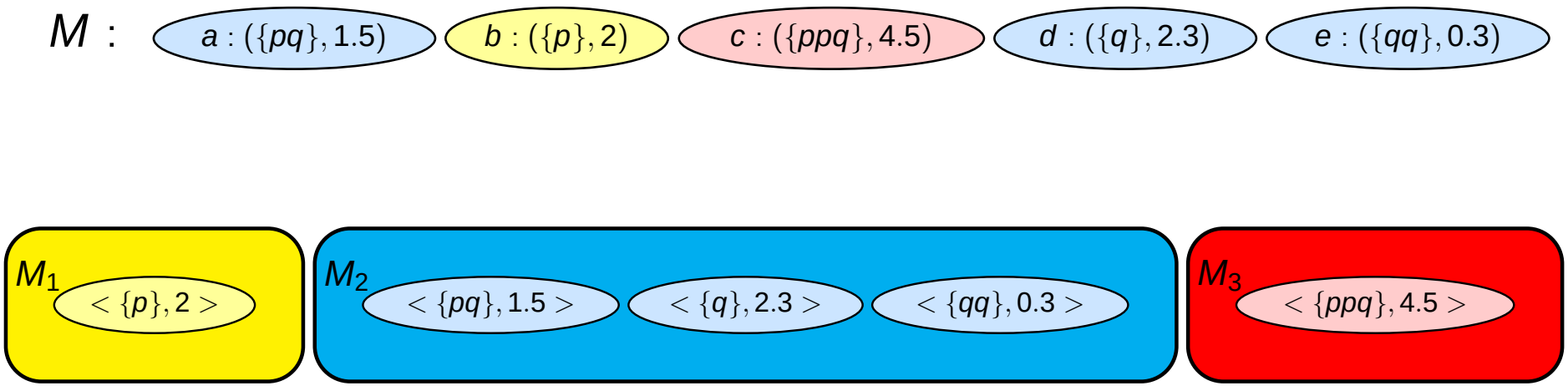
Transition system over a countable domain

M : 

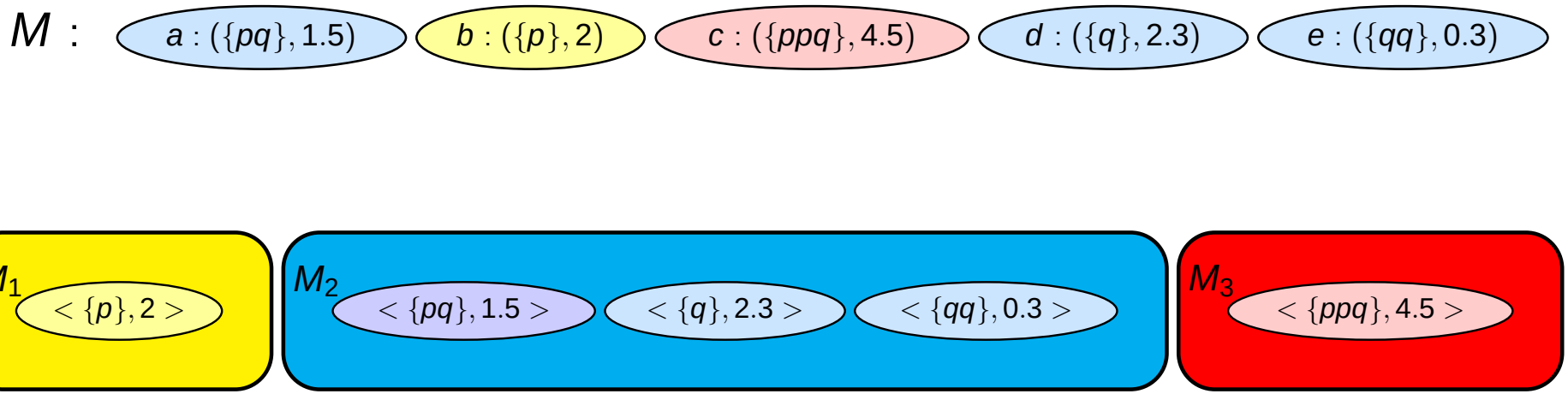
Transition system over a countable domain

M : 

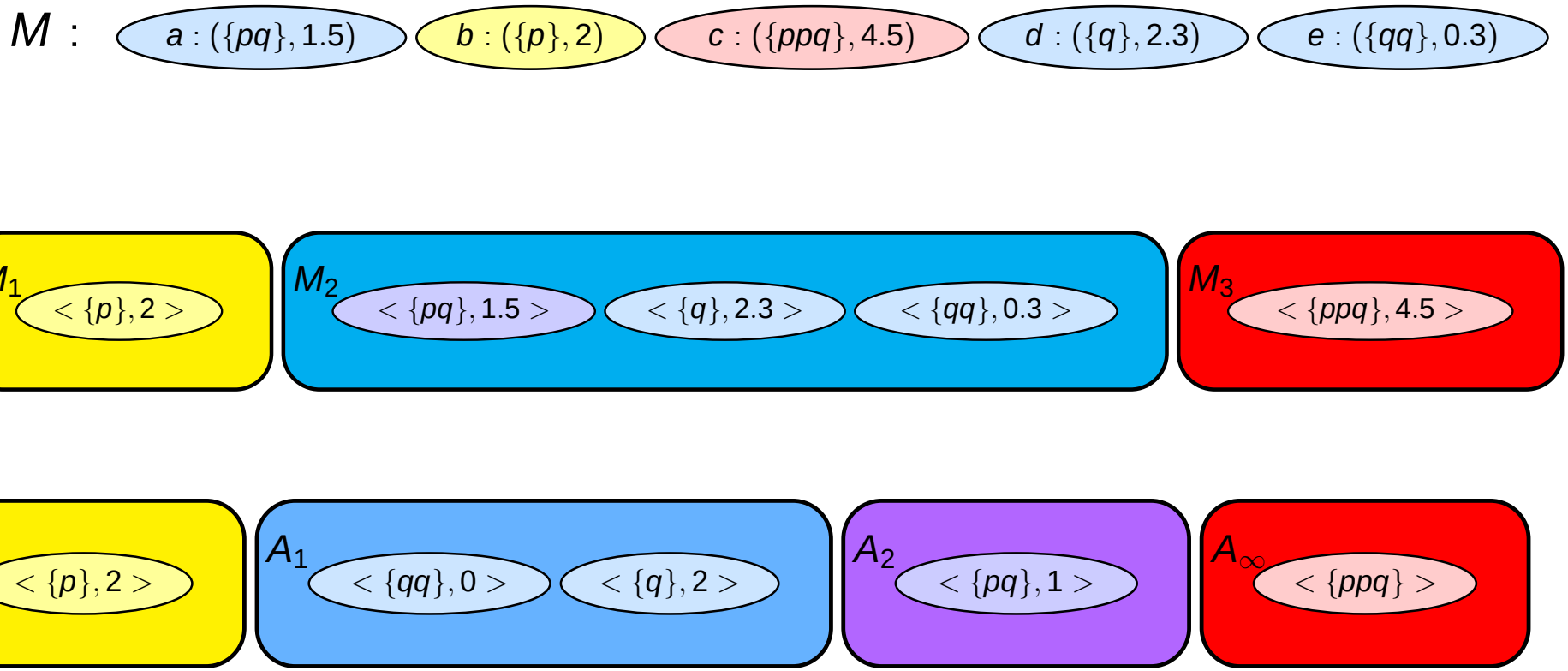
Transition system over a countable domain



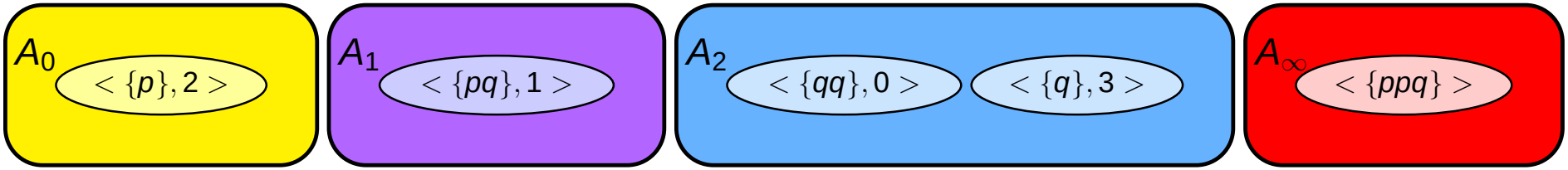
Transition system over a countable domain



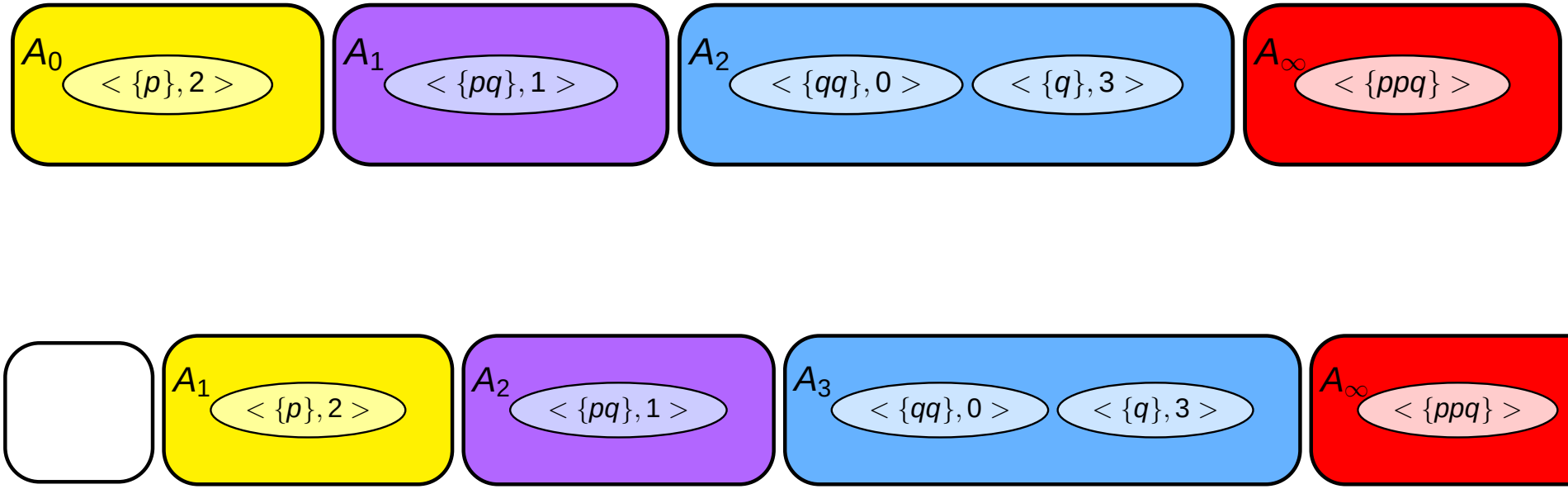
Transition system over a countable domain



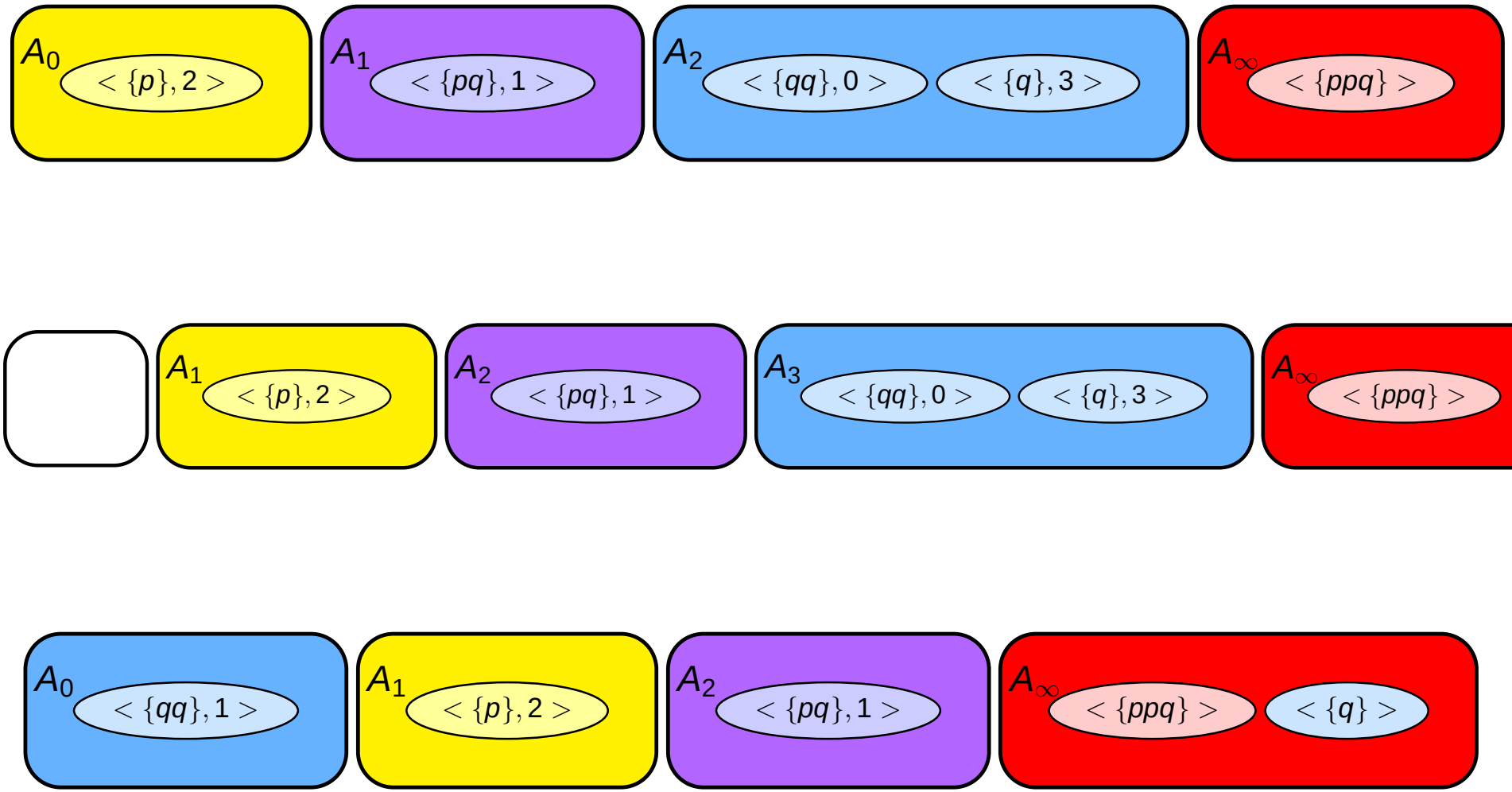
Time elapsing for regions



Time elapsing for regions



Time elapsing for regions



Well-Structured Transition Systems

Locally-timed ν-*PN* are WSTS. Therefore:

Proposition

Control-state reachability is decidable for Locally-timed ν-*PN*.

(Un)decidability results

- Control-state reachability is undecidable for ν-TdPN.
- Actually, already undecidable in the case of 2 clocks per instance
- Control-state reachability decidable for one clock per instance.

Outline

Language-based comparison

Languages of a labelled WSTS

Let $\mathcal{S} = (S, \rightarrow, s_0, s_f, \leq, \Sigma)$ be a WSTS with initial and final states s_0 and $s_f \in S$, and Σ a finite alphabet and $\rightarrow \subseteq S \times (\Sigma \cup \{\epsilon\}) \times S$. We define the coverability language: $L(\mathcal{S}) = \{w \in \Sigma^* \mid s_0 \xrightarrow{w} s \geq s_f\}$.

Languages of a class of WSTS

Let $\mathbf{M}$ be a class of WSTS. Then, $L(\mathbf{M}) = \{L(M) \mid M \in \mathbf{M}\}$

Define $\mathbf{M_1} \preceq \mathbf{M_2}$ if $L(\mathbf{M_1}) \subseteq L(\mathbf{M_2})$

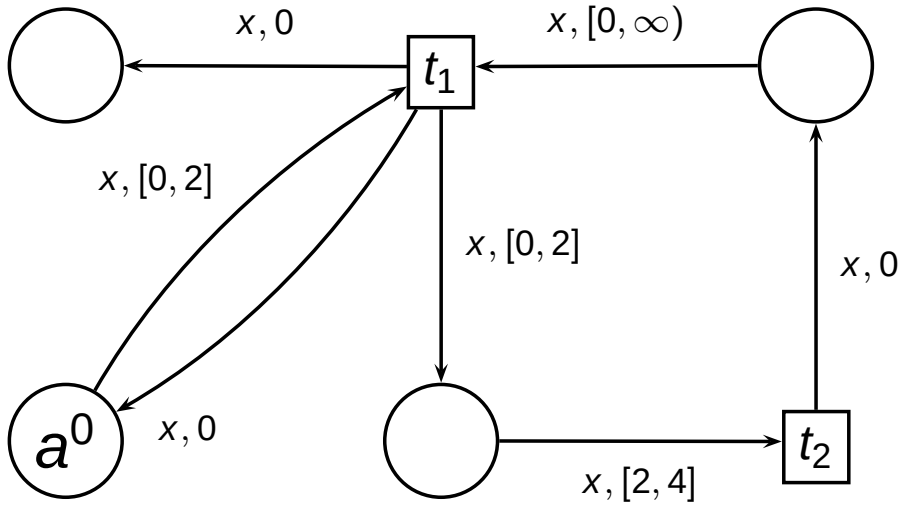
$$PN \prec AWN = LCM \prec \nu\text{-}PN \prec TdPN$$

Classes of ν-*lTdPN*

k-unboundedness

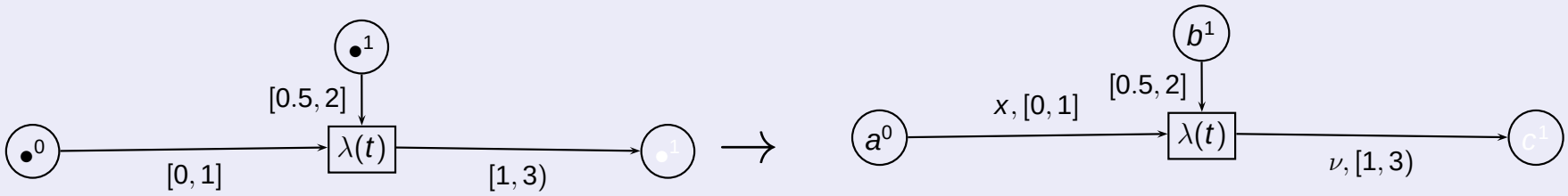
A ν-lTdPN is k-unbounded if *each instance* has at most k unbounded places.
ν-$lTdPN_k$ denotes the class of k-unbounded ν-*lTdPN*.

Example: $N \in \nu\text{-}lTdPN_1$

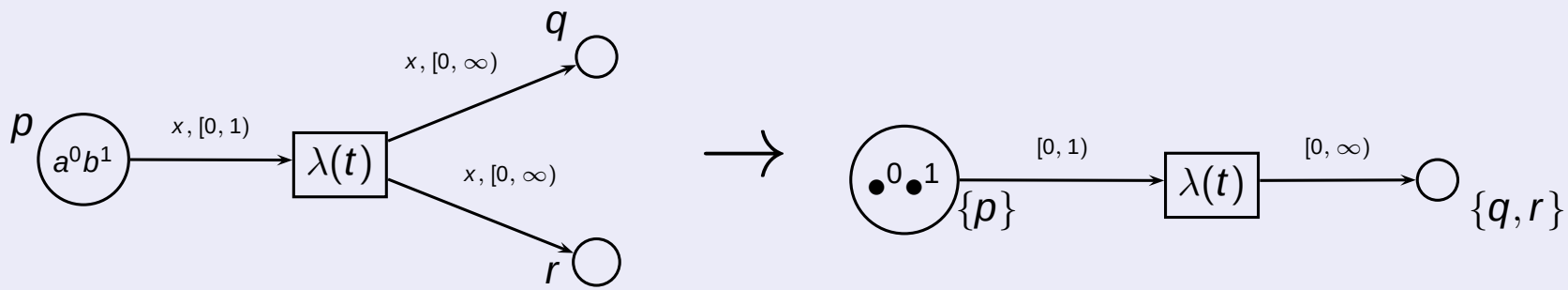


$TdPN = \nu\text{-}lTdPN_0$

$TdPN \preceq \nu\text{-}lTdPN_0$

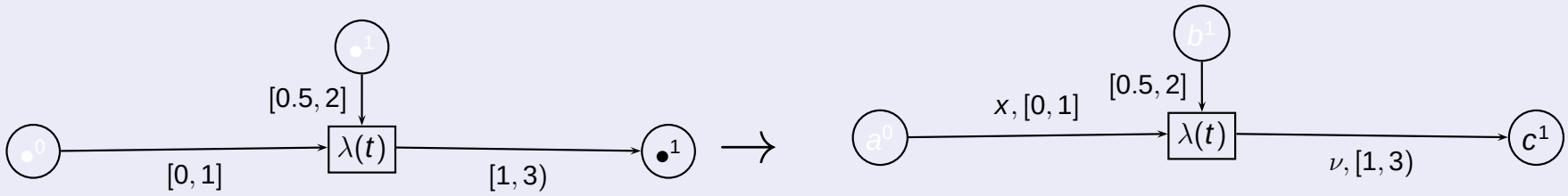


$\nu\text{-}lTdPN_0 \preceq TdPN$

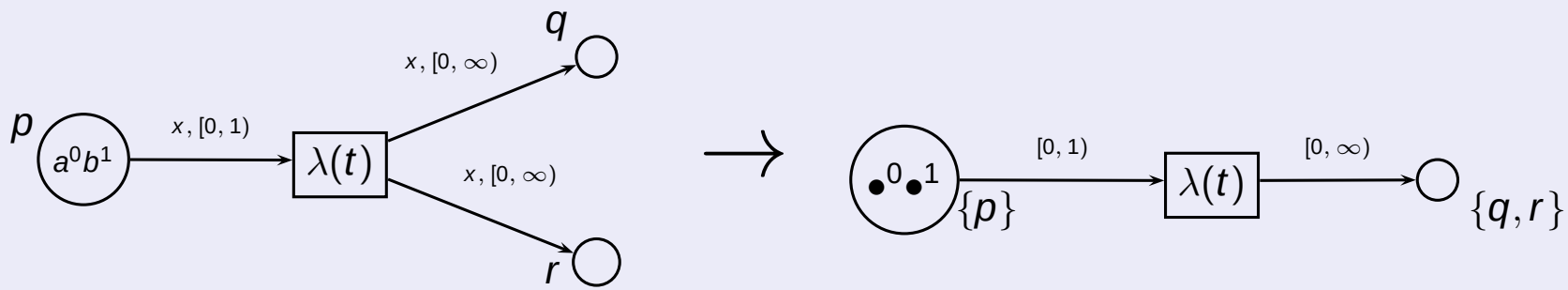


$TdPN = \nu\text{-}lTdPN_0$

$TdPN \preceq \nu\text{-}lTdPN_0$



$\nu\text{-}lTdPN_0 \preceq TdPN$



ν-lTdPN$_k \prec \nu$-lTdPN$_{k+1}$

A framework for the strict comparison of WSTS

Rule of thumb*: If $States(\mathbf{M_1}) \subset States(\mathbf{M_2})$ then $\mathbf{M_1} \not\succ \mathbf{M_2}$.

* Take with a lot of precaution, some hypotheses needed

Applying the framework based on ordinals

Q, P finite, and $k = |P|$.

- ν-lTdPN$_k \rightarrow X_k = (QxP^{\oplus})^{\oplus^*}$.
- $ot(X_k) = \omega^{\omega^{\omega^{\omega^k}} * |Q|}$.
- $k < k' \Rightarrow ot(X_k) < ot(X_{k'})$.

Expressiveness result

$$TdPN = \nu\text{-lTdPN}_0 \prec \nu\text{-lTdPN}_1 \prec \nu\text{-lTdPN}_2 \prec \ldots \prec \nu\text{-lTdPN}$$

ν-lTdPN$_k \prec \nu$-lTdPN$_{k+1}$

A framework for the strict comparison of WSTS

Rule of thumb*: If $States(\mathbf{M_1}) \subset States(\mathbf{M_2})$ then $\mathbf{M_1} \not\succ \mathbf{M_2}$.

* Take with a lot of precaution, some hypotheses needed

Applying the framework based on ordinals

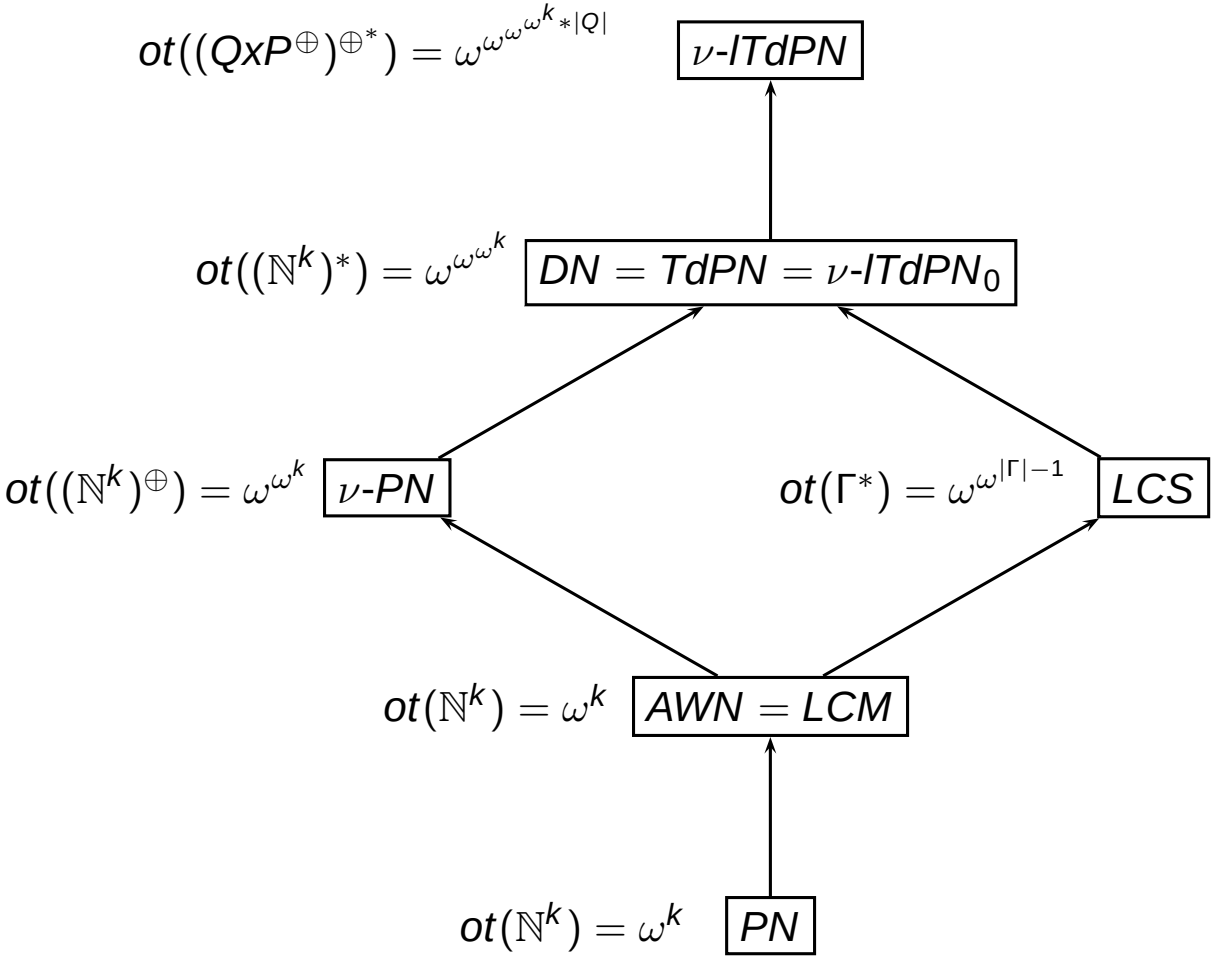
Q, P finite, and $k = |P|$.

- ν-lTdPN$_k \rightarrow X_k = (QxP^{\oplus})^{\oplus^*}$.
- $ot(X_k) = \omega^{\omega^{\omega^{\omega^k}}*|Q|}$.
- $k < k' \Rightarrow ot(X_k) < ot(X_{k'})$.

Expressiveness result

$$TdPN = \nu\text{-lTdPN}_0 \prec \nu\text{-lTdPN}_1 \prec \nu\text{-lTdPN}_2 \prec \ldots \prec \nu\text{-lTdPN}$$

Hierarchy of models



Outline

Conclusions

- Timed ν-*PN* (unbounded number of clocks per instance)$\rightarrow$ undecidable control-state reachability.
- Locally-timed ν-*PN* (one clock per instance)$\rightarrow$ decidable control-state reachability.
- $TdPN = \nu\text{-}lTdPN_0 \prec \nu\text{-}lTdPN_1 \prec \nu\text{-}lTdPN_2 \prec \ldots \prec \nu\text{-}lTdPN$.

Future Work

- Finer-grained complexity analysis.
- Other properties and restrictions, as the existence of Zeno behaviors.
- Timed Data Nets