Module T7. Discrete-Event systems

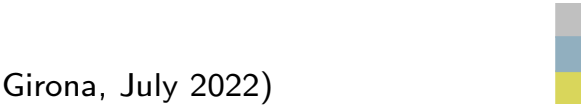
Diagnosis with Petri nets

Yannick Pencolé

7th July 2022

LAAS-CNRS, France

Preliminaries on Petri nets



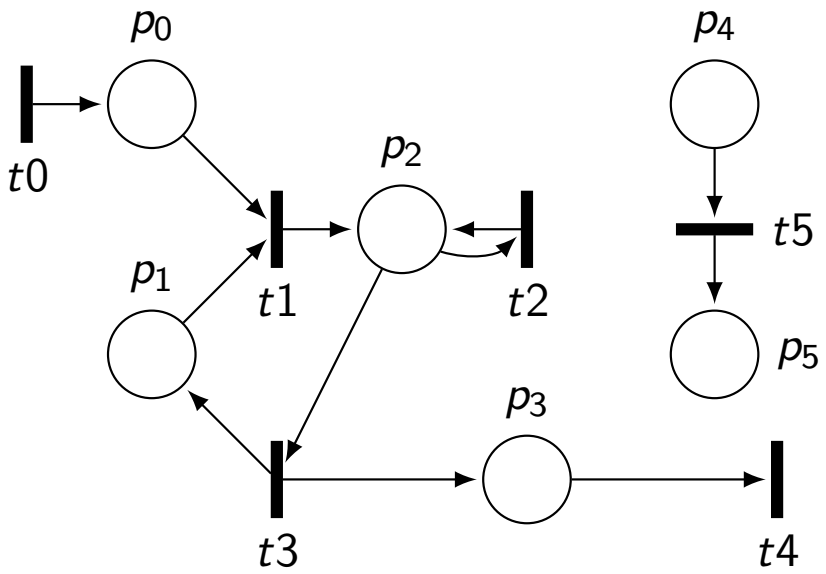
Petri nets : introduction

- Caracterisation of a class of discrete event dynamic systems
- Invented by Carl Adam Petri (1962)
- Bipartite Graph
- Two types of elements : **places** and **transitions**
- Well-suited to model distributed systems
  - Concurrency
  - Synchronisation

Elementary nets

A **net** is a tuple $N = (P, T, F)$:

1. P is the set of **places**;
2. T is the set of **transitions**; $(P \cap T = \varnothing)$
3. $F \subseteq (P \times T) \cup (T \times P)$ is a set of **arcs**.



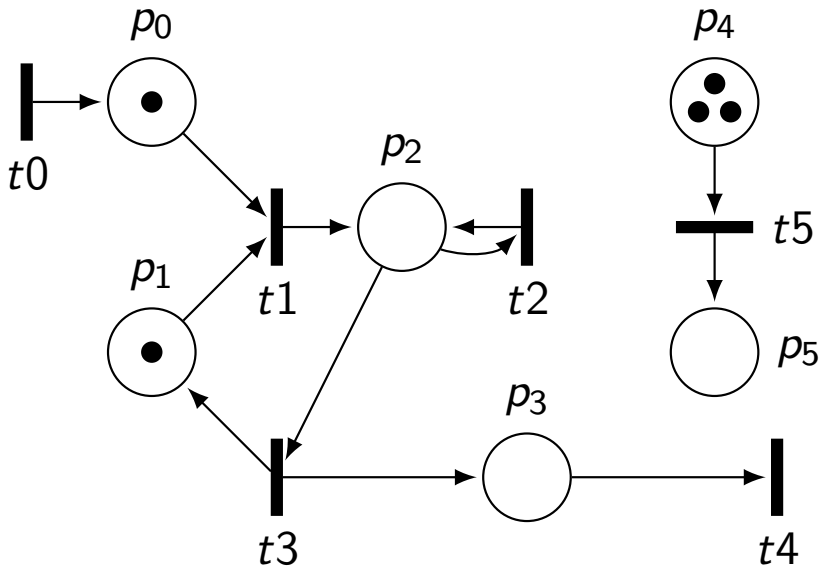
$$P = \{p_0, \ldots, p_5\}$$

$$T = \{t_0, \ldots, t_5\}$$

$$F = \{(t_0, p_0), (p_0, t_1), (p_1, t_1), \ldots\}$$

Marking of a net : tokens

Places model resources, **tokens** model their current use.
A **marking** M is a function $P \to \mathbb{N}$ that states how many tokens currently use the resources of the system.



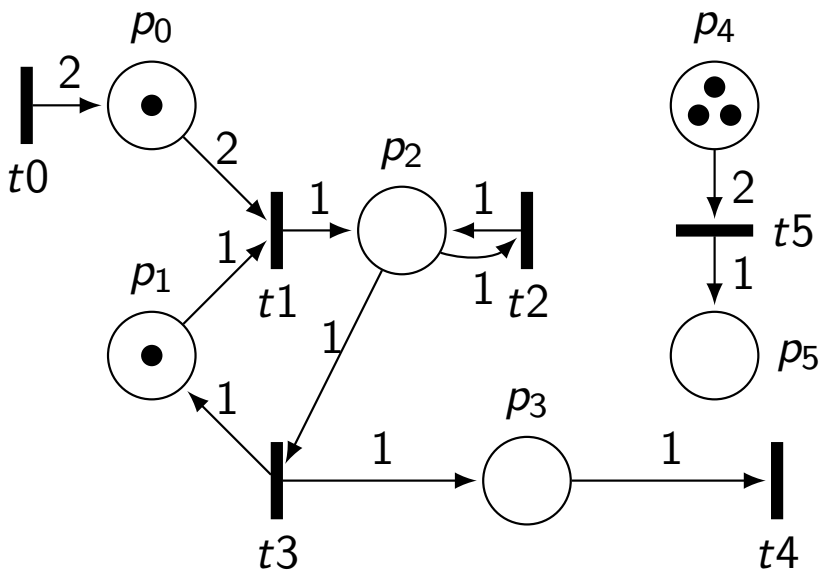
The current marking M of the net is :

- $M(p_0) = 1$
- $M(p_1) = 1$
- $M(p_2) = 0$
- $M(p_3) = 0$
- $M(p_4) = 3$
- $M(p_5) = 0$

Petri nets

A (marked) **Petri net** is a tuple (P, T, F, W, M_0) where :

1. (P, T, F) is a net,
2. M_0 is a **initial marking**,
3. W is a **weight function** : $W : F \to \mathbb{N}$.



Weights :

- $W(t_0 \to p_0) = 2$
- $W(p_0 \to t_1) = 2$
- $W(p_1 \to t_1) = 1$
- $W(t_1 \to p_2) = 1$
- ...

A (marked) net is a Petri net with all the weights set to 1.

Enabled Transitions

A Petri net evolves by firing transitions that update the current marking. A transition can be fired only if it is **enabled**.

- **Preset** of transition t :

$$pre(t) = {}^\bullet t = \{p \in P : W(p,t) > 0\}$$

- **Postset** of transition t :

$$post(t) = t^\bullet = \{p \in P : W(t,p) > 0\}$$

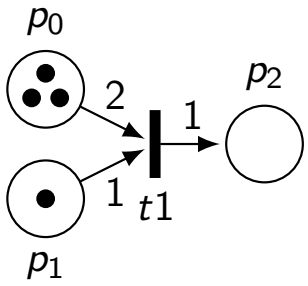
- Transition t is **enabled** in the marking M iff :

$$\forall p \in pre(t), M(p) \geq W(p,t)$$

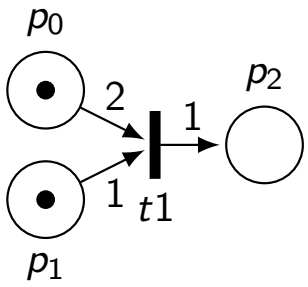
"There are enough tokens in the preset to feed the incoming arcs of the transition"

Enabled Transitions : examples

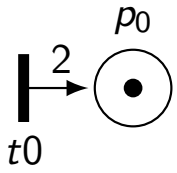
Case 1



Case 2



Case 3



- case 1 : t_1 is enabled ($M(p_0) = 3 > W(p_0, t_1) = 2$)
- case 2 : t_1 is not enabled ($M(p_0) = 1 < W(p_0, t_1) = 2$)
- case 3 : t_0 is enabled (empty preset, independent from any marking M)

Transition firing

Let t be a transition that is enabled in the current marking M. Firing t means updating the current marking M to a new marking M', only the marking of places in the pre/post of t are updated.

$$M \xrightarrow{t} M'$$

- The marking of any place p not involved in pre/post of t remains unchanged :

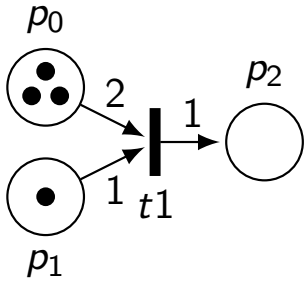
$$\forall p \in P \setminus (pre(t) \cup post(p)), M'(p) = M(p)$$

- Empty the places in the preset and feed the ones in postset :

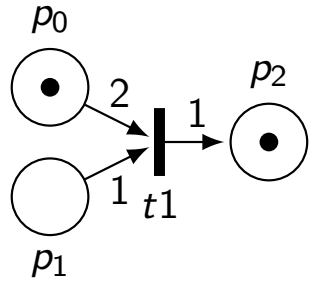
$$\forall p \in pre(t) \cup post(p), M'(p) = M(p) - W(p,t) + W(t,p)$$

Transition firing : examples

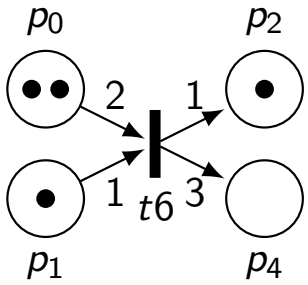
Case 1



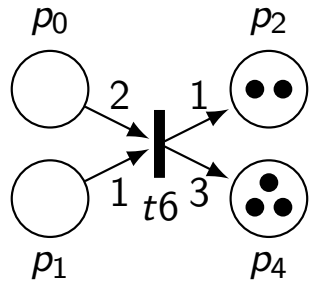
Case 1



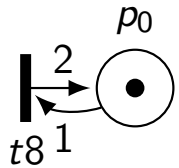
Case 2



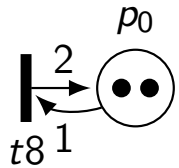
Case 2



Case 3



Case 3



Let us see the model of a system as a Petri net

About the system : a set of conveyors moving boxes

- Two levels, two conveyors per level
- A lift between the two levels.
- Boxes from level 1 are dispatched on the conveyors of level 0

Marking graph

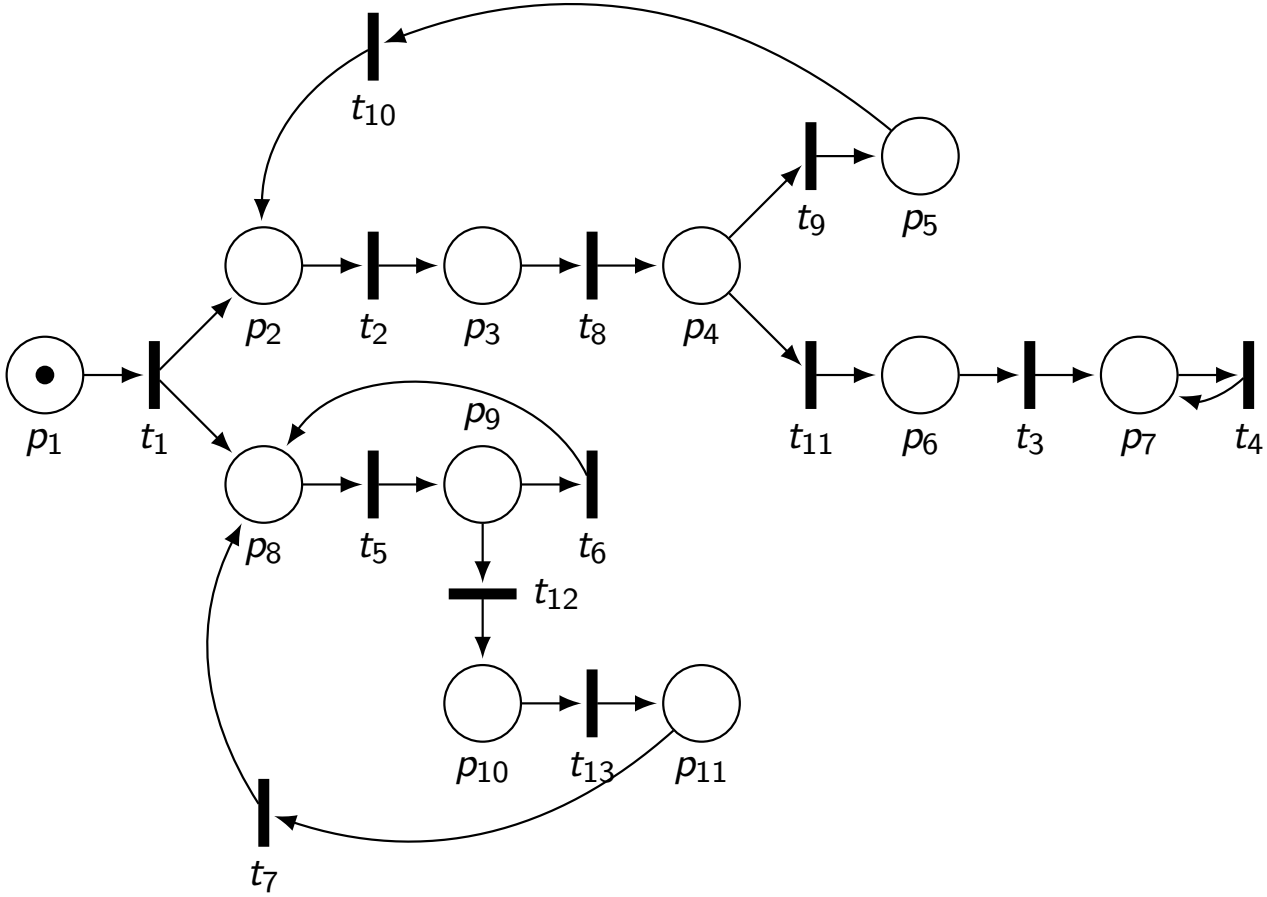
A marking M is **reachable** from an initial marking M_0 of a Petri net P if there exists a sequence of transition fires :

$$M_0 \xrightarrow{t_0} M_1 \xrightarrow{t_1} \dots \xrightarrow{t_n} M_n = M$$

The set of reachable markings $M \in R(P, M_0)$ can be explored through the **marking graph** :
Marking graph $G = (\mathcal{Q}, \mathcal{T}, \mathcal{E}, q_0)$

- $\mathcal{Q} = R(P, M_0)$ set of reachable markings
- $\mathcal{T}$ is the set of transitions $M \xrightarrow{t} M'$ where $M \in R(P, M_0)$ and t is fired in M to lead in marking M'.
- $\mathcal{E}$ set of transitions of the Petri net N.
- $q_0 = M_0$ the initial marking.

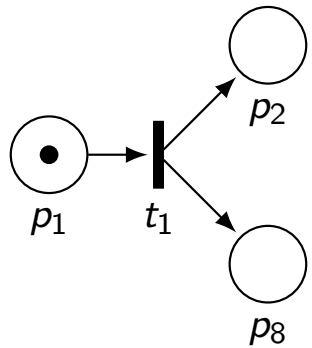
Petri net $\rightarrow$ Marking Graph example



In this example, all arcs have a weight = 1.

Petri net $\rightarrow$ Marking Graph example

Petri net extract : 1 enabled transition

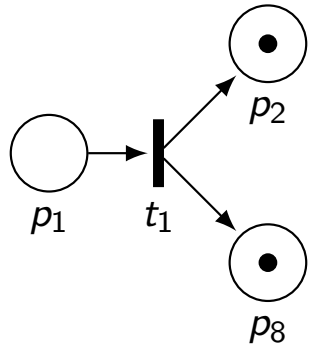


Computation of the Marking graph : step 1

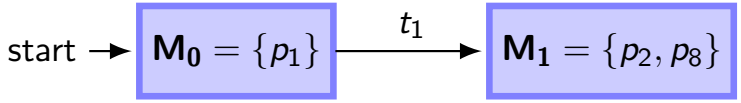
start $\rightarrow$ $\mathbf{M_0} = (1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)^T = \{p_1\}$

Petri net $\rightarrow$ Marking Graph example

Petri net extract : fire of t_1

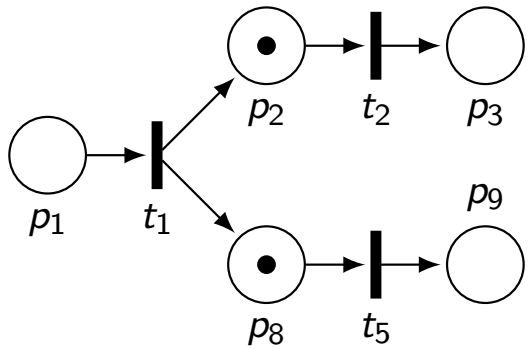


Computation of the Marking graph : step 2

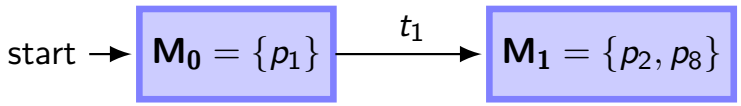


Petri net $\rightarrow$ Marking Graph example

Petri net extract : now t_2 and t_5 are enabled

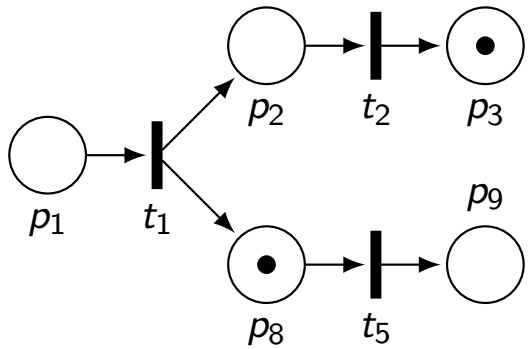


Computation of the Marking graph : step 2

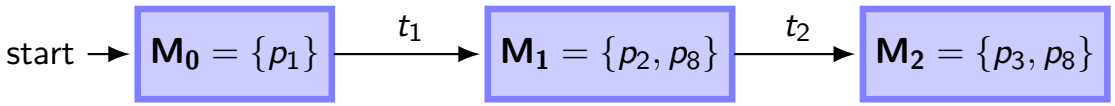


Petri net $\rightarrow$ Marking Graph example

Petri net extract : fire t_2, t_5 still enabled

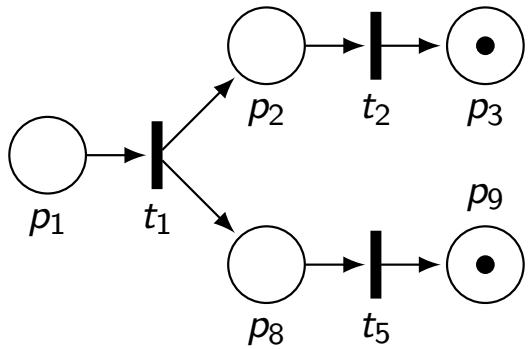


Computation of the Marking graph : step 3

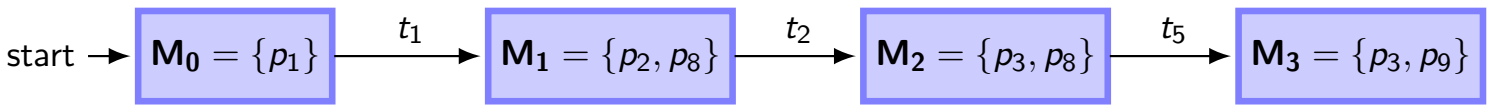


Petri net $\rightarrow$ Marking Graph example

Petri net extract : t_2 fired, fire t_5

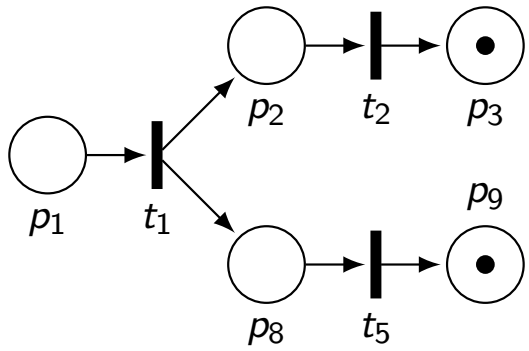


Computation of the Marking graph : step 4

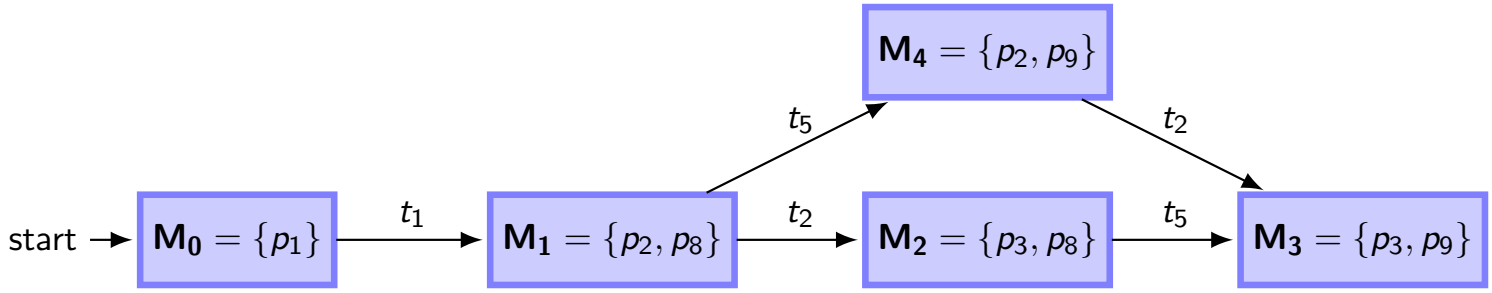


Petri net $\rightarrow$ Marking Graph example

Petri net extract : fire t_5 then fire t_2 from M_1



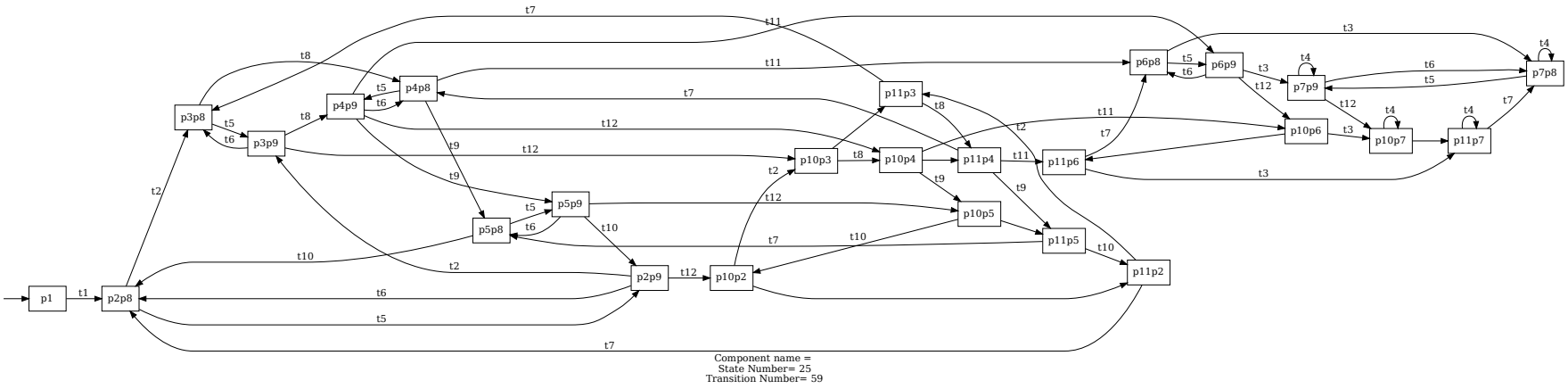
Computation of the Marking graph : steps 5 and 6



Transitions t_2 and t_5 are concurrent : interleaving

Petri net $\rightarrow$ Marking Graph : finally

In this example, the marking graph has **25 states** and **59 transitions**.



Remarks :

1. The number of states (reachable markings) in G is exponential to the number of places in the bounded worst case
2. The number of states in G might be unbounded
3. Classes of **Bounded**/**Unbounded** PN
4. Classes of **k-Bounded** Petri nets. (**safe = 1-bounded**)

Diagnosis/Diagnosability of bounded PNs

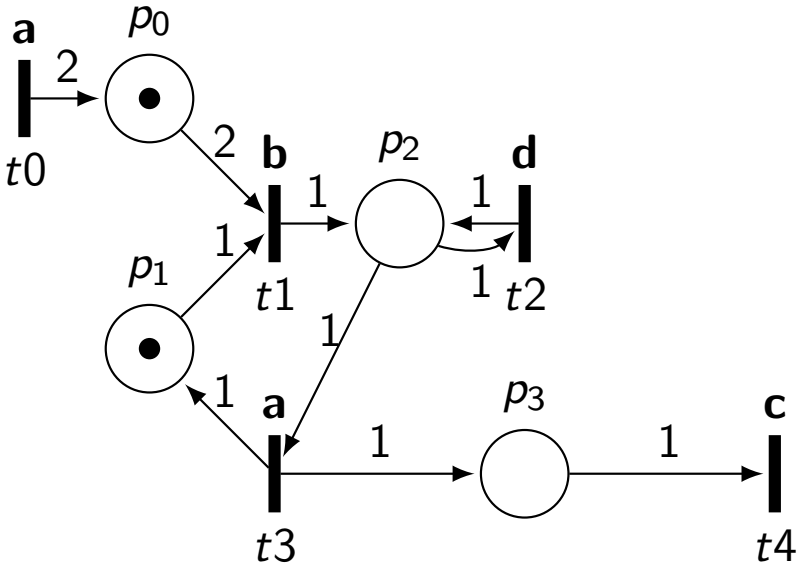
- Finite marking graph G is a finite automaton
- Faults $\rightarrow$ faulty transitions : a subset of Petri Net transitions
- Observables $\rightarrow$ observable transitions : a subset of Petri Net transitions
- It follows that : **any fault diagnosis/diagnosability method on automaton can be used on the marking graph of any bounded Petri Net.** (see lecture from this morning)
- Belief states, Sampath's diagnoser, Twin Plants...

Extensions to Labeled Petri Nets

A **Labeled Petri net** is a Petri net with a **labelling function** ℓ of the transitions.

$$\ell : T \to \mathcal{E}$$

A transition is associated with an event label. An event label may be associated with several transitions.

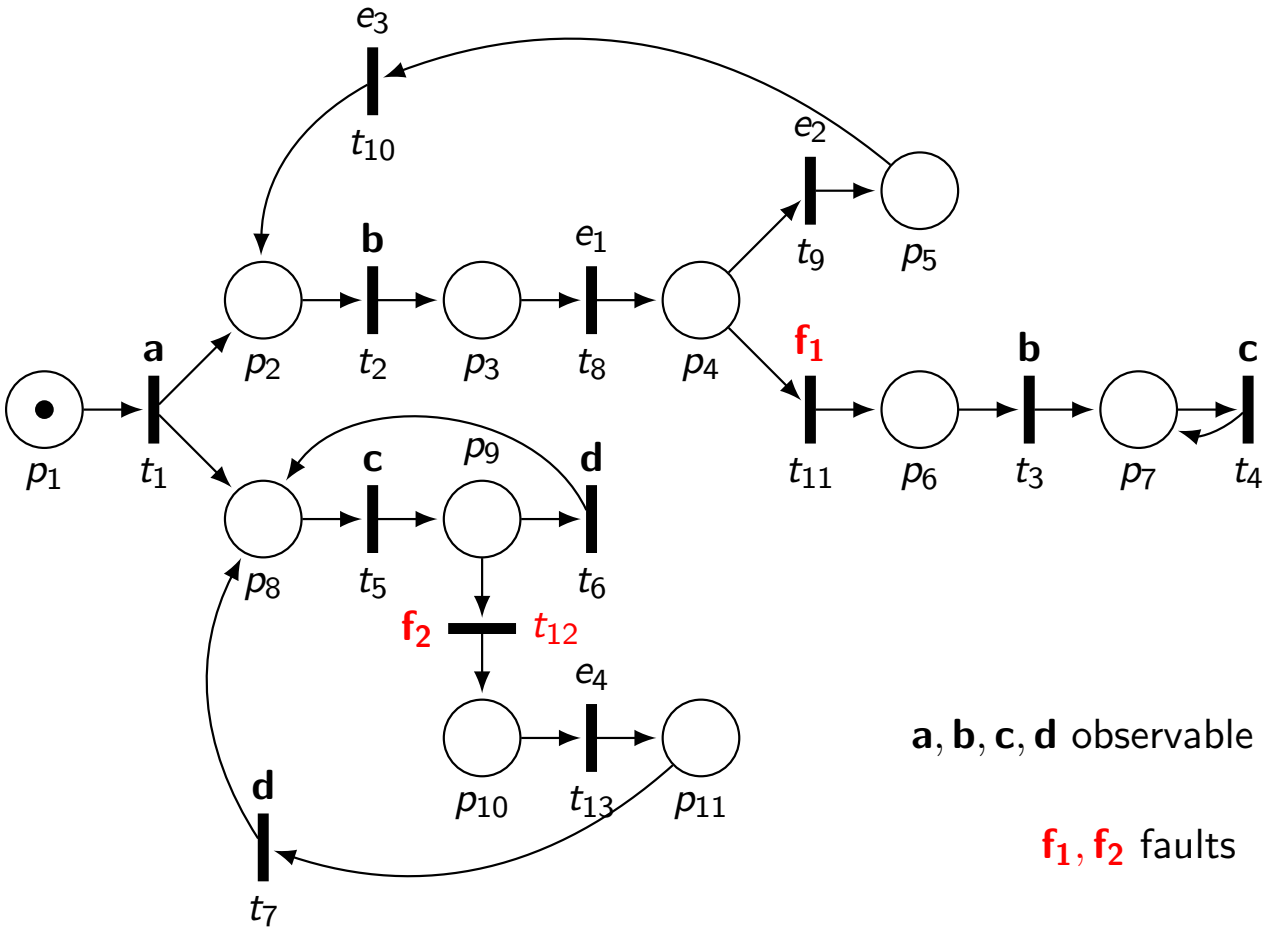


Labels

- $\ell(t_0) = \mathbf{a} = \ell(t_3)$
- $\ell(t_1) = \mathbf{b}$, $\ell(t_2) = \mathbf{b}$, $\ell(t_4) = \mathbf{c}$
- Generated langage of a LPN : $\mathcal{L} \subseteq \mathcal{E}^*$
- ex : $\sigma = \ell(t_0)\ell(t_1)\ell(t_0)\ell(t_2) = \mathbf{abad} \in \mathcal{L}$

Fault diagnosis in Petri nets : a model-checking approach

Diagnosis LPN model : an example

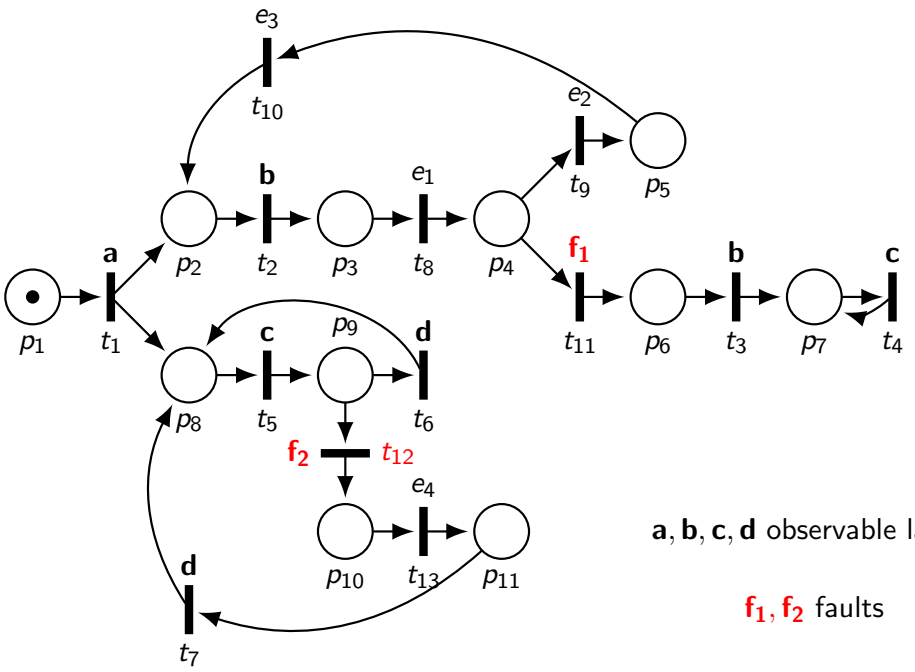


a, **b**, **c**, **d** observable labels

$\mathbf{f_1}$, $\mathbf{f_2}$ faults

A diagnosis problem in LPN

- Consider a diagnosis LPN model Θ.
- Consider one fault **f** labeling a set of faulty unobservable transitions T_f.
- Consider a sequence of observations σ produced by Θ.
- Has f definitely occurred or not ?.



a, **b**, **c**, **d** observable labels

$\mathbf{f_1}$, $\mathbf{f_2}$ faults

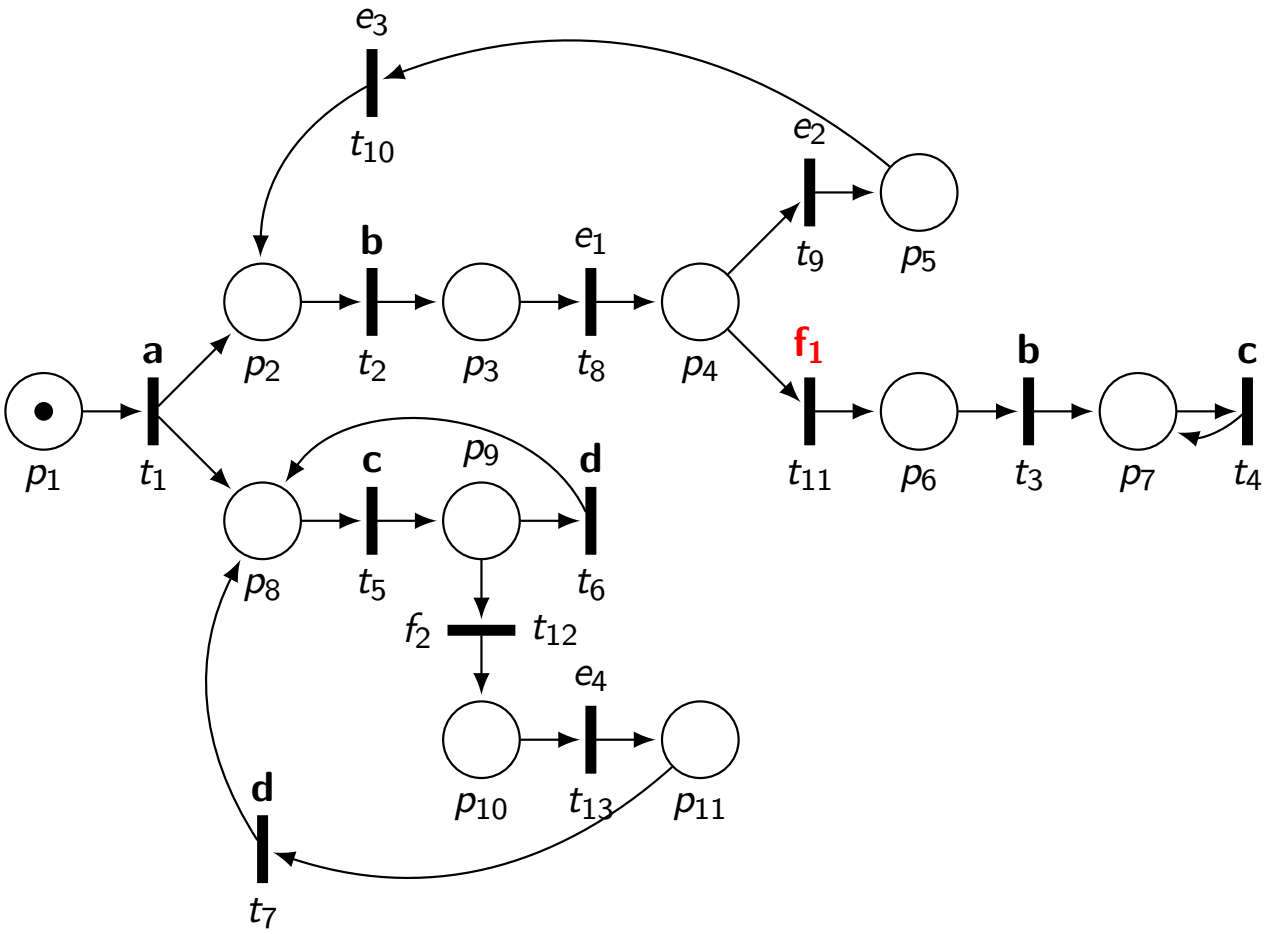
Example :

- Fault $\mathbf{f_1}$ on transitions $T_f = \{t_{11}\}$
- Observations : $\sigma =$ **abbc**
- Has $\mathbf{f_1}$ occurred if **abbc** is observed ?
- What about if $\sigma =$ **abbcc** ?

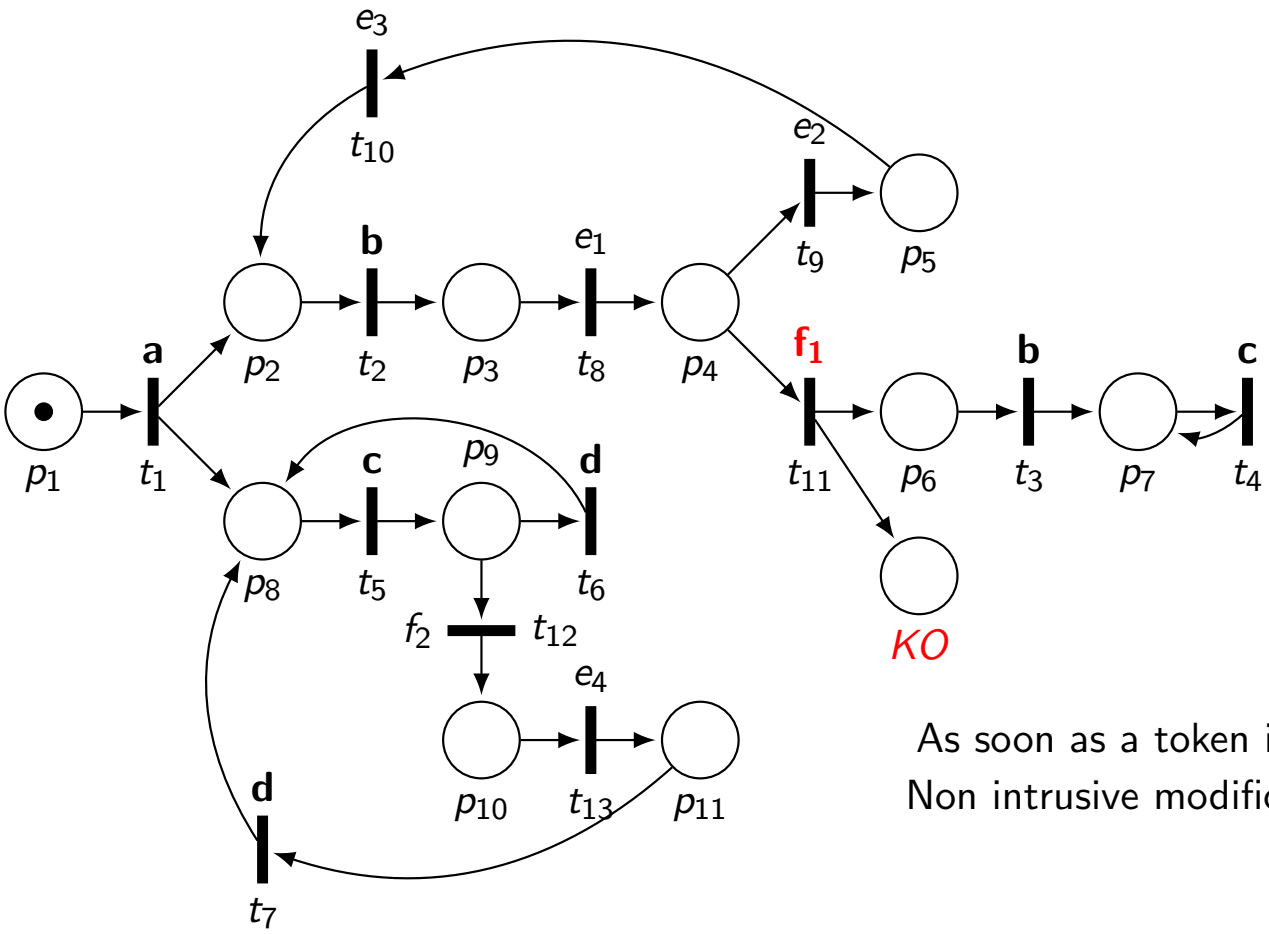
A model-checking method

- Model-checking problem :
  1. Given a transition system and a property Φ
  2. Checking whether property Φ holds
     - If yes, the model-checker will say yes.
     - If no, the model-checker will provide a behaviour of the system that does not respect Φ. (counter-example)
- Translate a diagnosis LPN problem into a model-checking problem
- Let the model-checker do the computational job.

Synthesis of the model checking problem (1)



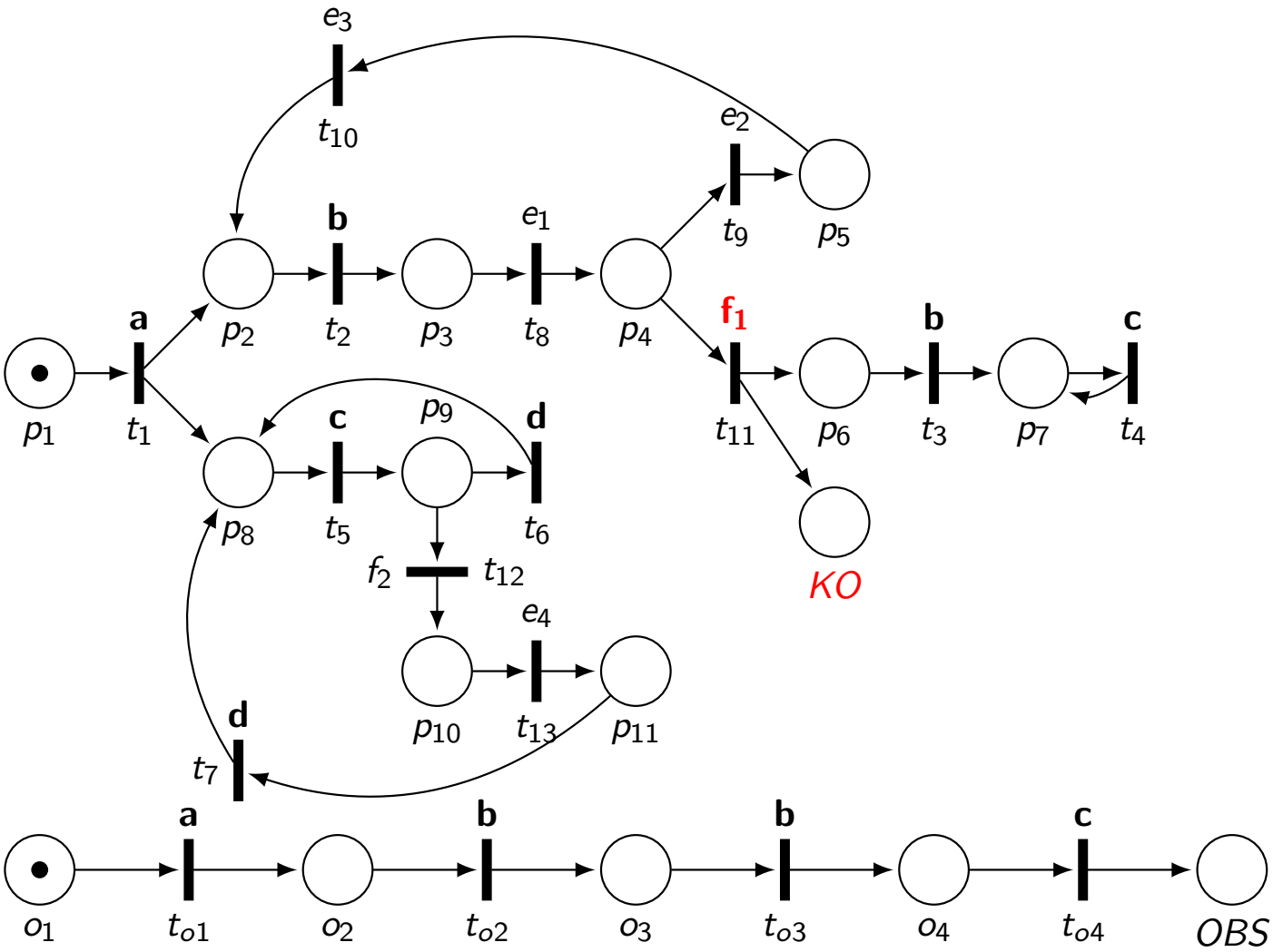
Synthesis of the model checking problem : place KO (1)



As soon as a token is in KO, fault f_1 has occurred

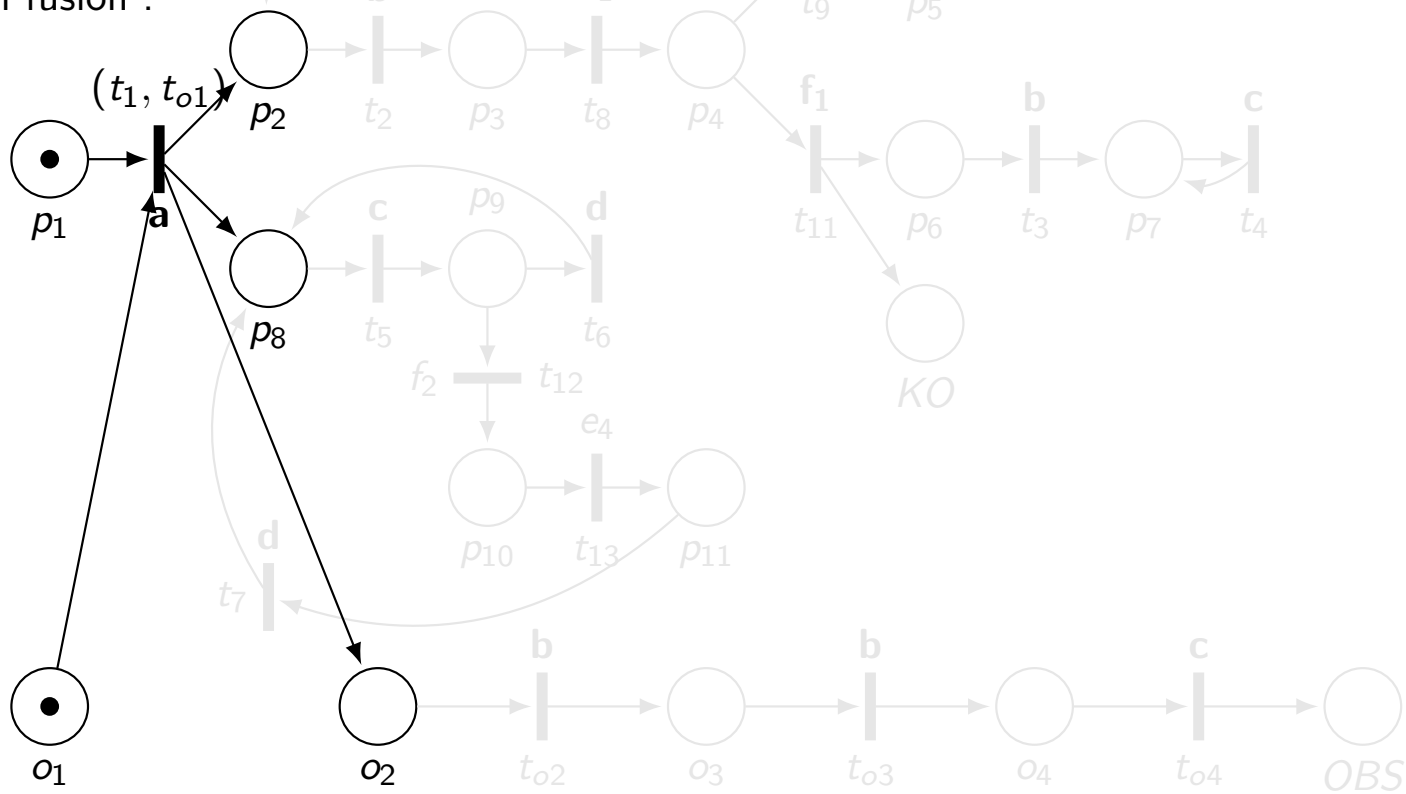
Non intrusive modification

Synthesis of the model checking problem : adding observations (2)



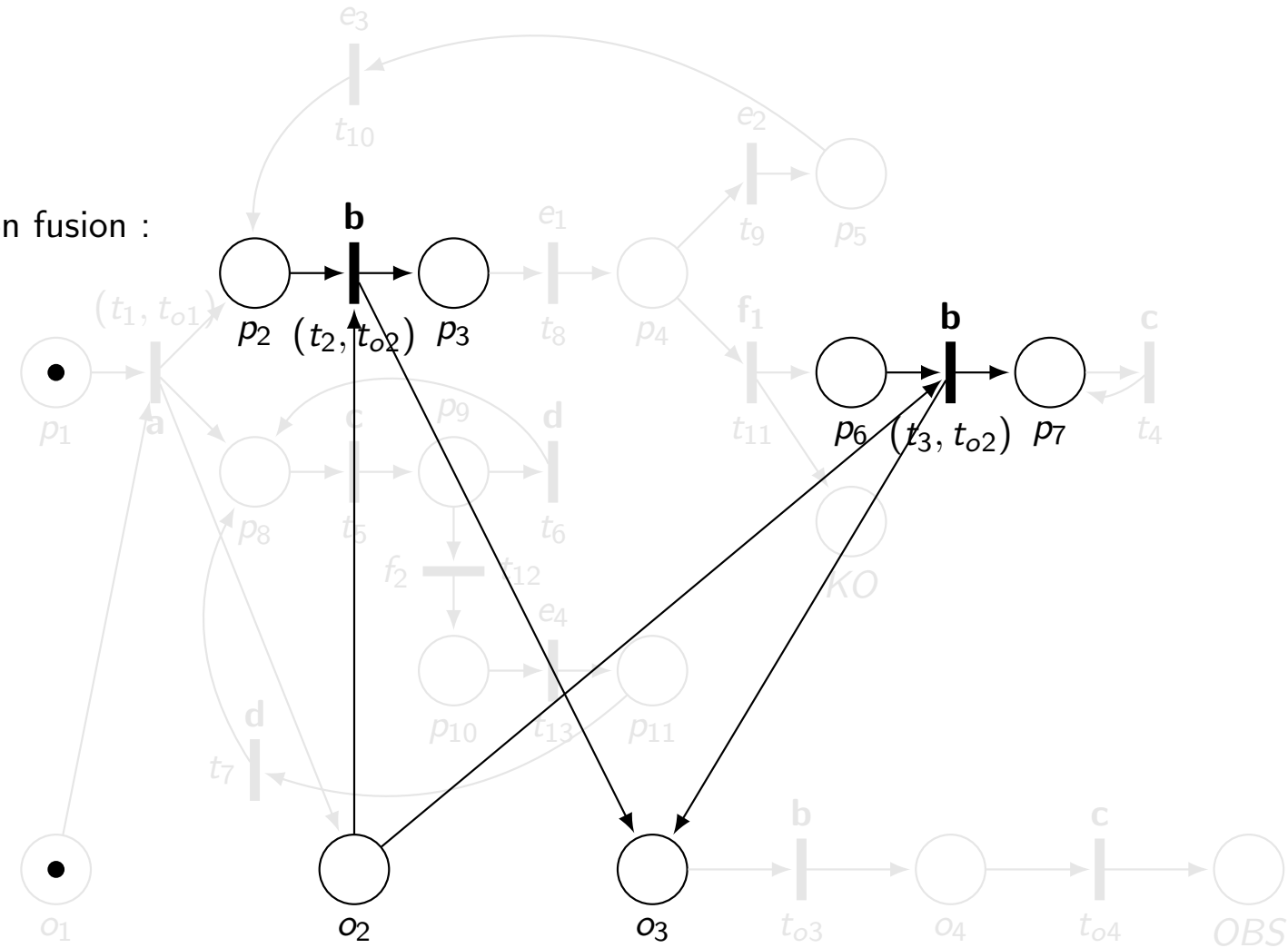
Synthesis of the model checking problem : transition fusion for event a (3)

Transition fusion :



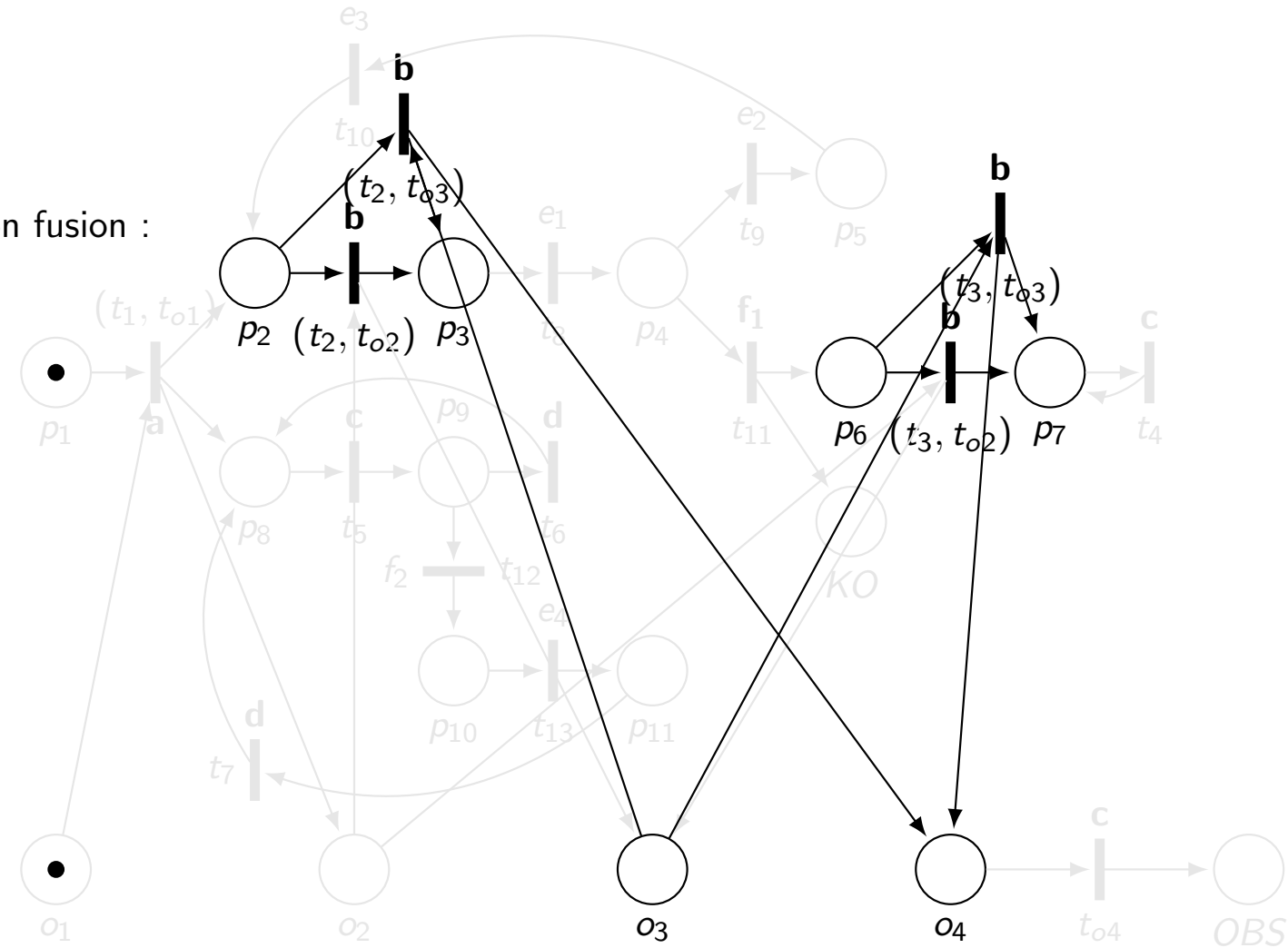
Synthesis of the model checking problem : transition fusion for 1st event b (4)

Transition fusion :



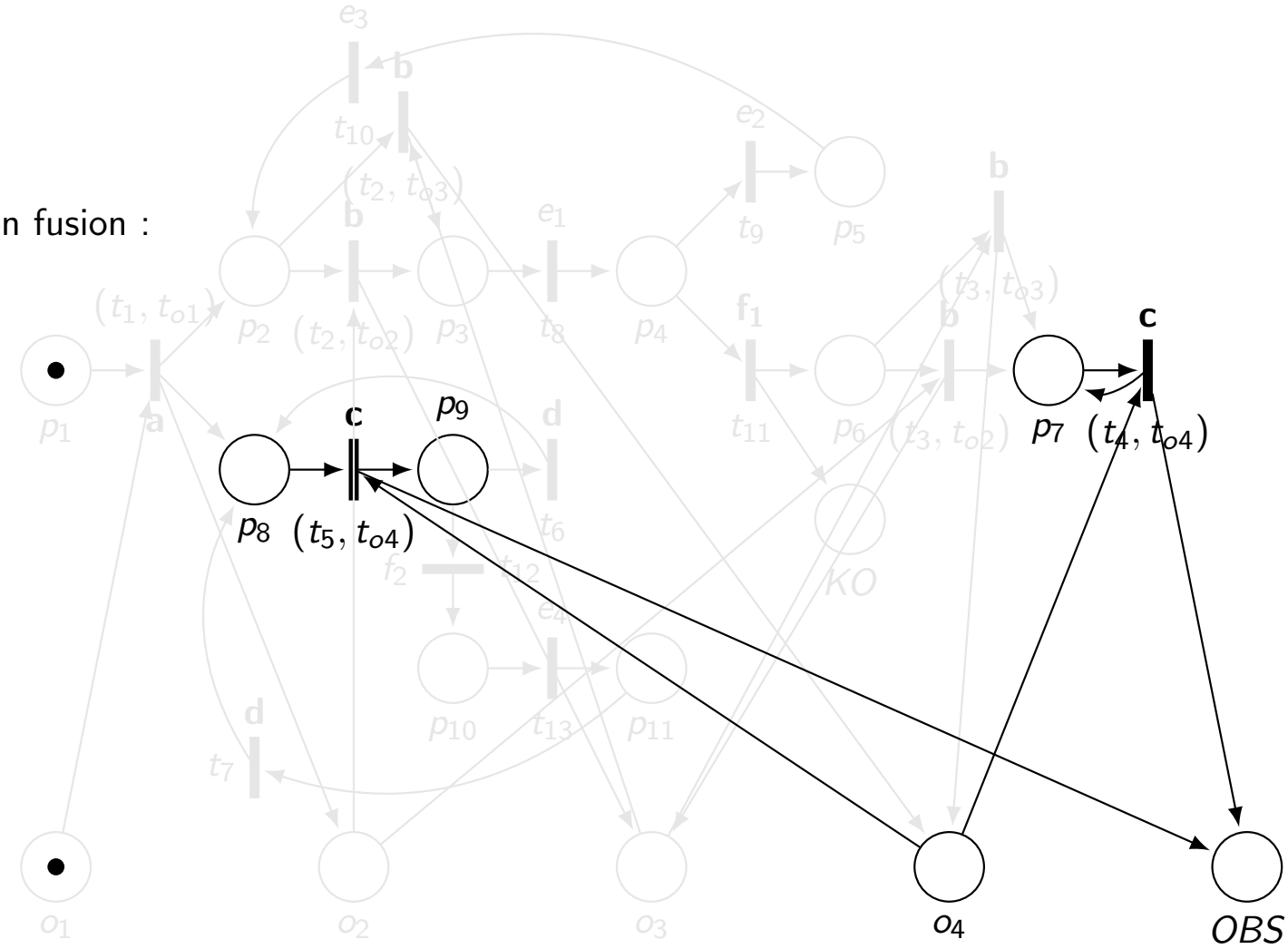
Synthesis of the model checking problem : transition fusion for 2nd event b (5)

Transition fusion :

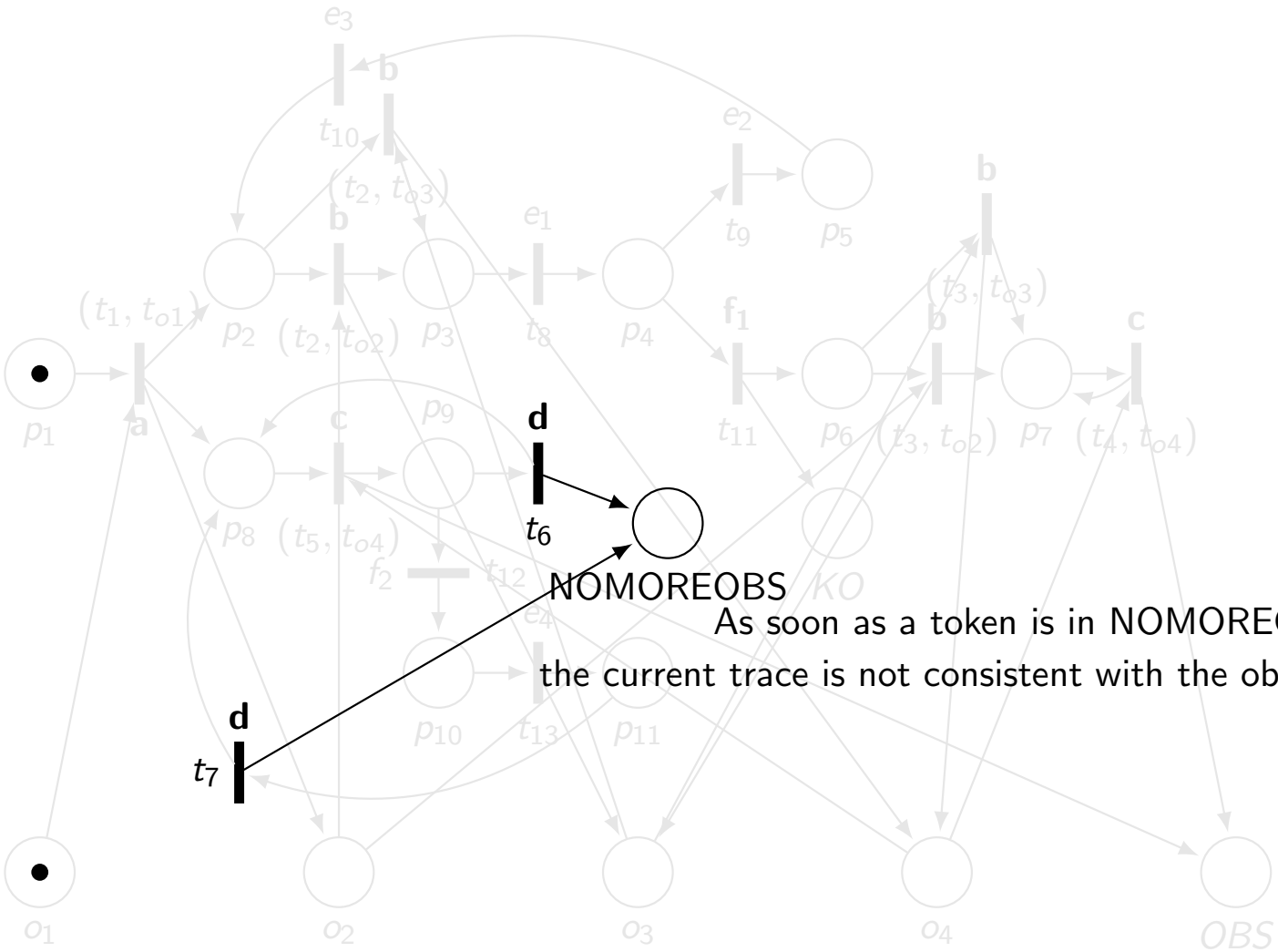


Synthesis of the model checking problem : transition fusion for event c (6)

Transition fusion :



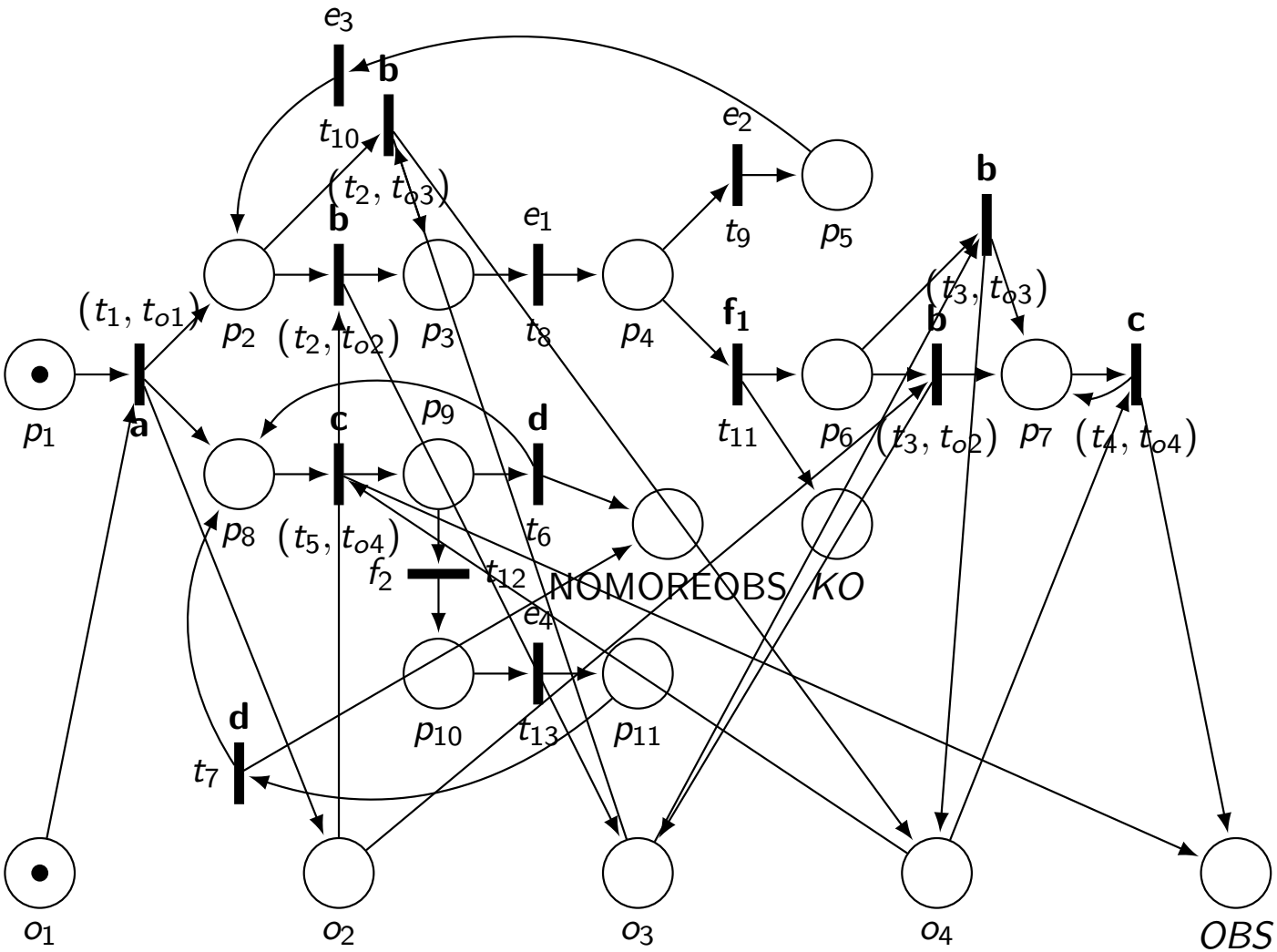
Synthesis of the model checking problem : place NOMOREOBS (7)



As soon as a token is in NOMOREOBS,

the current trace is not consistent with the observations **abbc**

Model checking problem : final model to check



Summary

By construction of the previous model :

- KO : as soon as there is a token, the trace is faulty :

$$KO \geq 1$$

- OBS : as soon as there is a token, the trace produced the observation **abbc** :

$$OBS = 1$$

- NOMOREOBS : as long as there is no token, the trace does not produce more observation than **abbc**

$$NOMOREOBS = 0$$

Properties to check on the model

- Property FAULTY : **is every trace consistent with abbc a faulty trace ?** Formally in LTL (linear temporal logic) :

$$\Box[(OBS = 1 \wedge NOMOREOBS = 0) \rightarrow (KO \geq 1)]$$

If the model-checker answers YES, we are DONE : f_1 has definitely occured. If NOT, we check a second property :

- Property HEALTHY : **is every trace consistent with abbc a healthy trace ?**

$$\Box[(OBS = 1 \wedge NOMOREOBS = 0) \rightarrow (KO = 0)]$$

If the model-checker answers YES, we are DONE : f_1 has definitely NOT occured. If NOT, we are DONE : there is an ambiguity.

Solutions

For this example : $\sigma = \mathbf{abbc}$

- FAULTY is false
- HEALTHY is false
- the occurence of f_1 **is ambiguous**

For another example : $\sigma = \mathbf{abbcc}$

- FAULTY is true
- the occurence of f_1 **is certain**

Summary of the method

- Translation of a diagnosis problem into a couple of model checking problems
- The translation is not complex (quadratic manipulations on LPN, transition fusions)
- No complex ad'hoc search algorithms
- No computation of belief states, just about the occurrence of a fault
- The complexity is in the model-checking phase :
  - Very efficient tools (ex : model-checker TINA (LAAS))
  - Perform symbolic encodings, partial-order reductions, symmetry

Diagnosability : a model-checking problem

- Diagnosability checking is a model-checking problem
- Used for diagnosability of automaton and LPN
- For LPN : pretty similar to the previous method
  1. Consider a LPN Θ and add the KO place.
  2. Duplicate it : Θ' and KO'
  3. Transition fusions of the observable transition Θ and Θ' : **twin-plant**
  4. Property to check : is there a **critical pair** ? Looks like this :

$$\square[(KO \geq 1) \rightarrow \Diamond(KO' \geq 1 \vee deadlock)]$$

Answer YES : Diagnosable

Answer NO : the given counter-example is a critical pair

Diagnoser based on a basis reachable graph (BRG)

Motivation

- Always the same issue : **combinatorial state explosion**
- How to design a diagnoser that does not require the computation of the marking graph
- Definition of an abstraction : **Basis Reachable Graph**
- BRG : Finite-state machine that store **minimal necessary explanations**
- Assumption : no unobservable transition cycles in the underlying LPN.
- Based on the notion of **firing vectors**.

Incidence matrix

Transition fire, remainder :

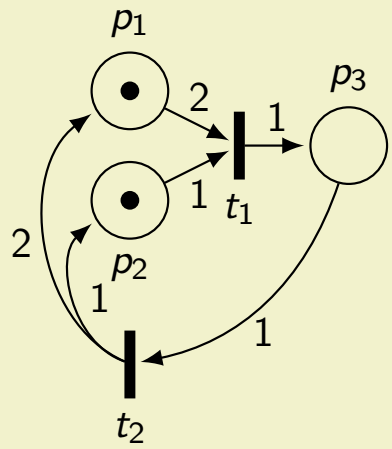
$$\forall p \in pre(t) \cup post(p), M'(p) = M(p) - W(p,t) + W(t,p)$$

and $M'(p) = M(p)$ otherwise (i.e. when $W(p,t) = W(t,p) = 0$)

Factorization as an **incidence matrix C** :

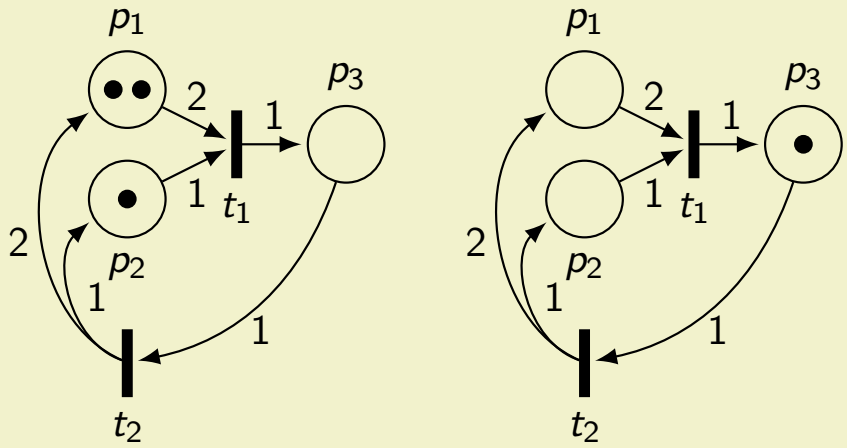
$$\mathbf{C} = \begin{pmatrix} W(t_1,p_1) - W(p_1,t_1) & \cdots & W(t_n,p_1) - W(p_1,t_n) \\ W(t_1,p_2) - W(p_2,t_1) & \cdots & W(t_n,p_2) - W(p_1,t_n) \\ \vdots & & \vdots \\ W(t_1,p_m) - W(p_m,t_1) & \cdots & W(t_n,p_m) - W(p_m,t_n) \end{pmatrix}$$

Incidence matrix : example



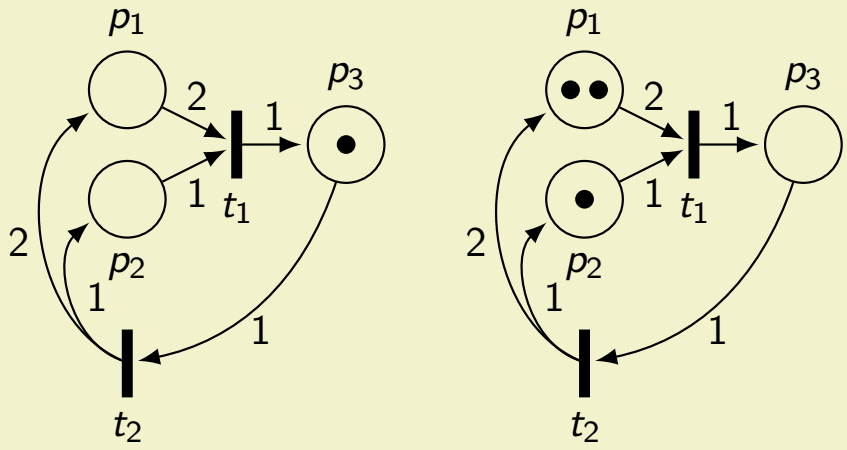
$$\mathbf{C} = \begin{pmatrix} 0-2 & 2-0 \\ 0-1 & 1-0 \\ 1-0 & 0-1 \end{pmatrix} = \begin{pmatrix} -2 & 2 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}$$

Incidence matrix : reachable marking



$$\mathbf{M_1} = \mathbf{M_0} + \mathbf{C} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -2 & 2 \\ -1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -2 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Incidence matrix : reachable marking



$$\mathbf{M_2} = \mathbf{M_1} + \mathbf{C}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\1\end{pmatrix} + \begin{pmatrix}-2 & 2\\-1 & 1\\1 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\1\end{pmatrix} + \begin{pmatrix}2\\1\\-1\end{pmatrix} = \begin{pmatrix}2\\1\\0\end{pmatrix}$$

$\mathbf{M_2} = \mathbf{M_0}$.

Incidence matrix : firing vectors (Parikh)

Remark :

$$\mathbf{M_2} = \mathbf{M_1} + \mathbf{C}\begin{pmatrix}0\\1\end{pmatrix} = \mathbf{M_0} + \mathbf{C}\begin{pmatrix}1\\0\end{pmatrix} + \mathbf{C}\begin{pmatrix}0\\1\end{pmatrix} = \mathbf{M_0} + \mathbf{C}\begin{pmatrix}1\\1\end{pmatrix}$$

In this example, generally :

$$\mathbf{M_0} = \mathbf{M_0} + \mathbf{C}\begin{pmatrix}k\\k\end{pmatrix} \qquad \mathbf{M_1} = \mathbf{M_0} + \mathbf{C}\begin{pmatrix}k+1\\k\end{pmatrix}$$

Vectors $\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}k\\k\end{pmatrix}, \begin{pmatrix}k+1\\k\end{pmatrix}$ are **firing vectors** (kind of Parikh vectors) : number of transition fires of every t_i.

Possibly reachable markings

Let $\mathbf{M}$ be a marking, if $\mathbf{M}$ is reachable from $\mathbf{M_0}$ then there exists a firing vector $\mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, $x_i \geq 0$ such that :

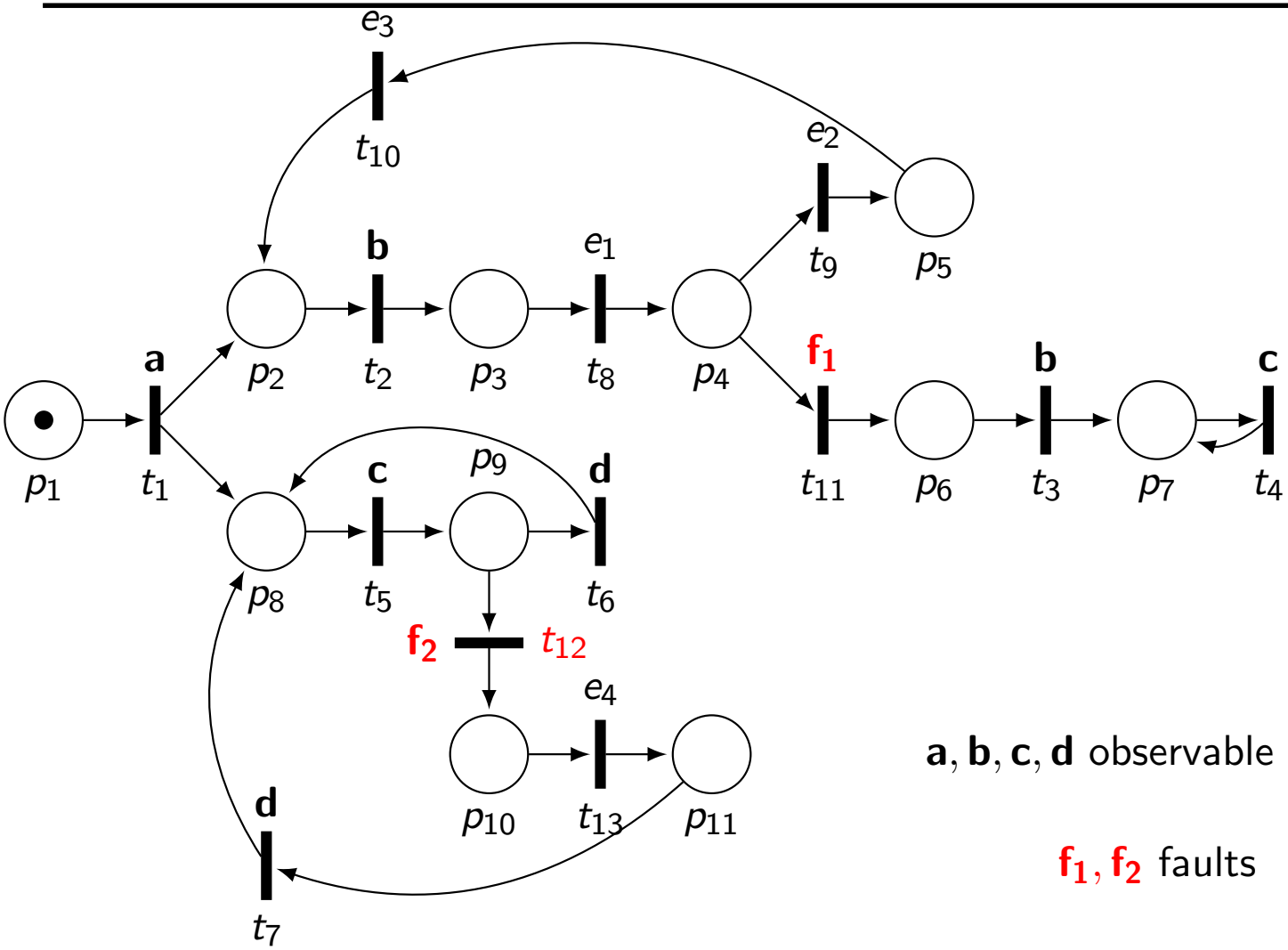
$$\mathbf{M} = \mathbf{M_0} + \mathbf{CX}$$

Necessary but not sufficient condition !

If a marking $\mathbf{M}$ is such that $\mathbf{M} = \mathbf{M_0} + \mathbf{CX}$, then $\mathbf{M}$ is a **possibly reachable marking** but not necessarily reachable.

Now, here is the trick : **if the Petri net is not cyclic, it becomes a sufficient condition.**

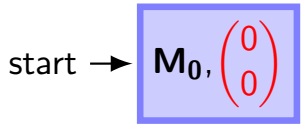
Diagnosis problem : same as before



a, **b**, **c**, **d** observable labels

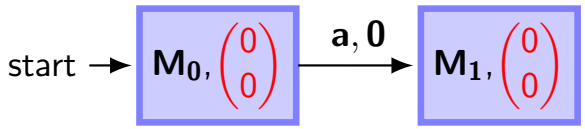
$\mathbf{f_1}, \mathbf{f_2}$ faults

Computation of the Basis Reachable Graph : step (1)



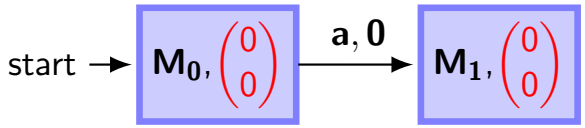
- Initial marking $\mathbf{M_0} = \{p_1\}$
- Firing vector $\mathbf{X_0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ about the **feasible faulty trajectories** from $\mathbf{M_0}$:
  1. Compute any marking $\mathbf{M}$ and firing vector $\mathbf{X}$ from $\mathbf{M_0}$ such that
     $$\mathbf{M} = \mathbf{M_0} + \mathbf{C_u}\mathbf{X}, \mathbf{X}(f_1) > 0, \mathbf{X}(f_2) > 0,$$
     $\mathbf{C_u}$ incidence matrix restricted to unobservable transitions of the model. $\mathbf{M}$ is reachable due to acyclicity
  2. If for a computed $\mathbf{X}$, the number of occurences of $\mathbf{f_i}$ is greater than 0 then $\mathbf{X_0}(f_i) = 1$ otherwise $\mathbf{X_0}(f_i) = 0$
  3. In this case, $\mathbf{X_0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Computation of the Basis Reachable Graph : step (2)



- $\mathbf{a}, \mathbf{0} = \mathbf{a}, \mathbf{E}^0_{min} = [0, 0, 0, 0, 0, 0]^T$
- **a** first observable that can fired after $\mathbf{M_0}$
- $\mathbf{M_1}$ **basis reachable marking** from $\mathbf{M_0}$
- $\mathbf{E}^0_{min}$ is the **minimal explanation** (firing vector)

Computation of the Basis Reachable Graph : basis reachable marking



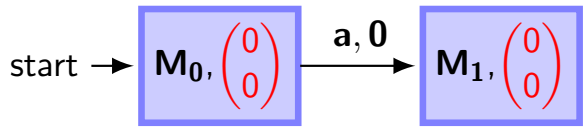
- $\mathbf{M_1}$ basis reachable marking from $\mathbf{M_0}$

$$\mathbf{M_0} \xrightarrow{\tau_1} \dots \xrightarrow{\tau_n} \mathbf{M_1}$$

with

1. $\tau_i, i < n$ are unobservable ($\ell(\tau_i) \in \mathcal{E}_u$)
2. $\ell(\tau_n) = \mathbf{a}$
3. $\xrightarrow{\tau_1} \dots \xrightarrow{\tau_{n-1}}$ is **minimal**
4. **No subsequence of** $\xrightarrow{\tau_1} \dots \xrightarrow{\tau_{n-1}}$ leads to the observation of **a** and reaches $\mathbf{M_1}$
5. In other words, $\xrightarrow{\tau_1} \dots \xrightarrow{\tau_{n-1}}$ is **necessary** to reach $\mathbf{M_1}$ by transition τ_n

Computation of the Basis Reachable Graph : minimal explanation



- $\mathbf{M_1}$ basis reachable marking from $\mathbf{M_0}$

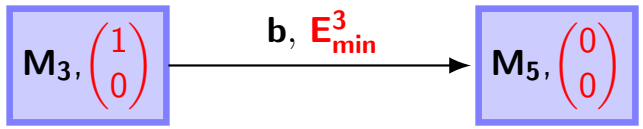
$$\mathbf{M_0} \xrightarrow{\tau_1} \ldots \mathbf{M_1}' \xrightarrow{\tau_n} \mathbf{M_1}$$

  - $\xrightarrow{\tau_1} \ldots \xrightarrow{\tau_{n-1}}$ is **minimal**
  - $\mathbf{E}^0_{min}$ is the firing vector of $\xrightarrow{\tau_1} \ldots \xrightarrow{\tau_{n-1}}$ called the **minimal explanation** :

$$\mathbf{M_1}' = \mathbf{M_0} + \mathbf{C_u}\mathbf{E}^0_{min}$$

  - $\mathbf{E}^0_{min}$ minimal number of occurrences per unobservable transitions between $\mathbf{M_0}$ and $\mathbf{M_1}$.

Computation of the Basis Reachable Graph : minimal explanation

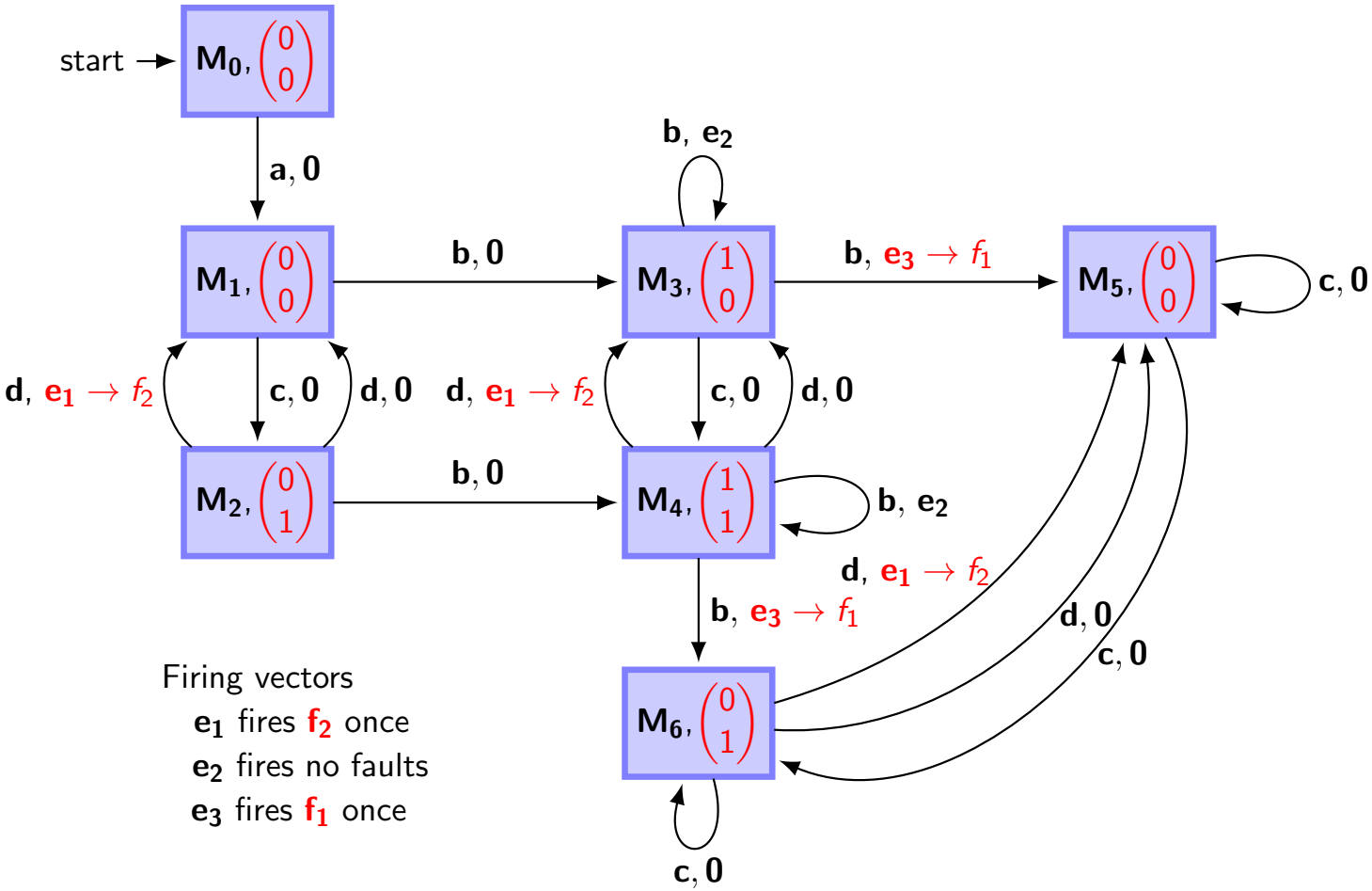


- $\mathbf{M_5}$ basis reachable marking from $\mathbf{M_3}$
- $\mathbf{M_3} = \{p_3 p_8\}$
- $\mathbf{M_5} = \{p_7 p_8\}$
- $\mathbf{E}^3_{min}$ is the firing vector :

$$[1_{t_8}\, 0_{t_9}\, 0_{t_{10}}\, 1_{t_{11}}\, 0_{t_{12}}\, 0_{t_{13}}]^T$$

- Faulty transition : t_{11} $(\mathbf{f_1})$ necessarily occurs as it is in the minimal explanation between $\mathbf{M_3}$ and $\mathbf{M_5}$

Basis Reachable Graph



Use of the BRG as a diagnoser (1)

Case 1 : we observe nothing ε.

- Initial marking in $\mathbf{M_0}$:

$$(1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)^T = "p_1"$$

- Based on BRG : we stay in state $\mathbf{M_0}$, $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- The firing vector $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ asserts that there is no run from the silent closure after $\mathbf{M_0}$ where f_1 or f_2 has occurred.

Diagnosis : **both f_1 and f_2 have not occurred.**

Use of the BRG as a diagnoser (2)

Case 2 : we observe **ab**.

- From initial marking $\mathbf{M_0}$, we reach

$$\mathbf{M_3} = (0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0)^T = "p_3 p_8"$$

through transitions $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix}$

- From $\xrightarrow{a,0}$ and $\xrightarrow{b,0}$: no fault has necessarily occurred
- The firing vector $\begin{pmatrix}1\\0\end{pmatrix}$ asserts that there is a run from the silent closure after $\mathbf{M_3}$ where f_1 has occurred.

Diagnosis : f_1 **may have occurred but not** f_2**.**

Use of the BRG as a diagnoser (3)

Case 3 : we observe **abbc**.

- From BRG : 3 possible sequences
  1. $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{b,e_2} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_4}, \begin{pmatrix}1\\1\end{pmatrix}$
  2. $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{b,e_3 \rightarrow f_1} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix}$
  3. $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{b,e_3 \rightarrow f_1} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_6}, \begin{pmatrix}0\\1\end{pmatrix}$

Diagnosis : **Both f_1 and f_2 may have occurred.**

Use of the BRG as a diagnoser (4)

Case 4 : we observe **abbcc**.

- From BRG : 3 possible sequences
  1. $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{b,e_3\to f_1} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix}$
  2. $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{b,e_3\to f_1} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_6}, \begin{pmatrix}0\\1\end{pmatrix}$
  3. $\mathbf{M_0}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{a,0} \mathbf{M_1}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{b,0} \mathbf{M_3}, \begin{pmatrix}1\\0\end{pmatrix} \xrightarrow{b,e_3\to f_1} \mathbf{M_5}, \begin{pmatrix}0\\0\end{pmatrix} \xrightarrow{c,0} \mathbf{M_6}, \begin{pmatrix}0\\1\end{pmatrix} \xrightarrow{c,0} \mathbf{M_6}, \begin{pmatrix}0\\1\end{pmatrix}$

Diagnosis : f_1 **has certainly occurred and** f_2 **may have occurred.**

Summary of the BRG approach

- Computation of a BRG
  - smaller than a Marking Graph : (7 $<$ 25 states, 18 $<$ 59 transitions)
  - a way to solve the combinatorial explosion problem

- Abstraction of the problem
  - Use of partial-order reduction technique (abstraction of concurrent and unecessary unobservable transitions)
  - Solving equations $\mathbf{M} = \mathbf{M_0} + \mathbf{CX}$: efficient solvers

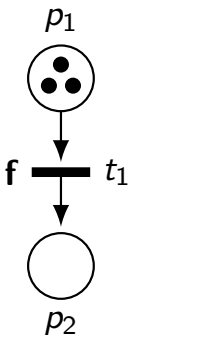
References

1. **Diagnosability analysis of patterns on bounded labeled prioritized Petri nets**. Gougam, Pencolé, Subias, JDEDS 2016
2. **Diagnosis of supervision patterns on bounded labeled Petri nets by Model Checking**. Pencolé, Subias DX 2018
3. **How to use Model Checking for diagnosing fault patterns in Petri nets**. Bakalara, Pencolé, Subias, Wodes 2020
4. **Diagnosis of discrete event systems using labeled Petri nets. An application to manufacturing systems** Cabasino, Giua Seatzu Control Engineering Practice 2011
5. **Petri Nets : Properties, Analysis and Applications.** Murata, IEEE 1989
6. **Diagnosis of DES With Petri Net Models**, Lefebvre, Delherm. IEEE TASE 2007
7. **Diagnosis of asynchronous discrete-event systems : a net unfolding approach**, Benveniste, Fabre, Haar, Jard, TAC 2003.
8. **Fault Detection of Discrete Event Systems Using Petri Nets and Integer Linear Programming**, Dotoli, Fanti, Mangini, IFAC 2008

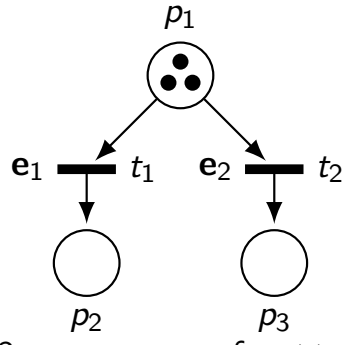
Diagnosis of discrete event systems : some extensions

Extensions to patterns

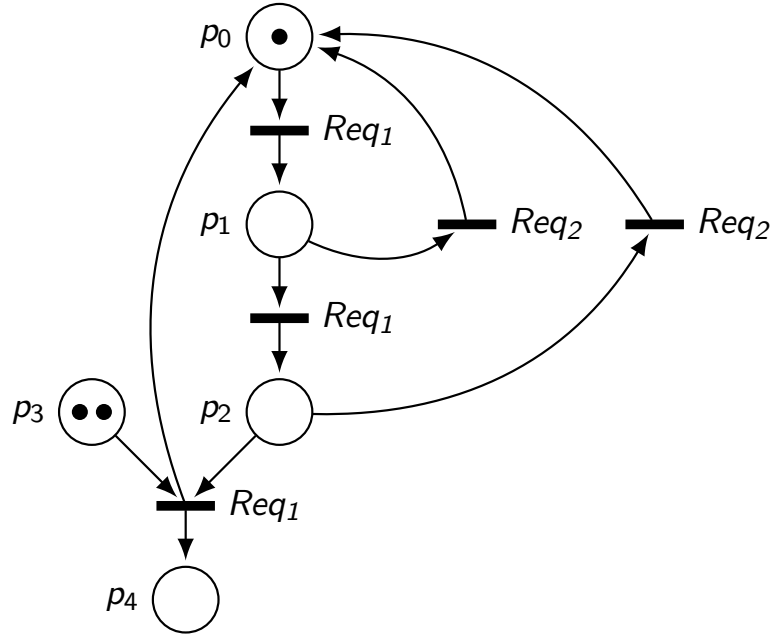
- In this lecture, a fault is the occurrence of a single event.
- Can be extended to more complex events : **patterns**



3 occurrences of **f**



3 occurrences of $\mathbf{e_1} \vee \mathbf{e_2}$



3 occurrences of $\mathbf{Req_1}$ without $\mathbf{Req_2}$, twice unbalanced behavior
bottom-left conveyors is more often used.

Diagnosis of timed Discrete-Event Systems

- In this lecture, time is represented as a **sequence of events** **abbccd**
- There exist a lot of very recent works on diagnosis about timed Discrete-Event Systems
  - **timed sequence of events** **1a3b3b4c3c2d** is not **2a4b3b6c3c2d** : same sequence but not the same dates
  - Time is discriminant (delays...)
  - Some work on time automata (Alur), time Petri Nets, time event graphs (max,+) algebra
  - Diagnosability of TDES
  - Δ-diagnosability
  - Ad'hoc algorithms, Model-Checking techniques, SMT-solvers (SAT + arithmetic theory for time constraints)

Diagnosis of stochastic DES systems

- Stochastic DES : markov processes
- Diagnosis : look for the most likely trajectories consistent with the observations
- Stochastic diagnosers
- Definitions of a set of diagnosability problems on probabilistics DES. (limits to probability 0)
- Stochastic automaton, Stochastic Petri Nets
- Use of firing probability laws.