Réseaux de Petri temporels à inhibitions/permissions

Application à la modélisation et vérification de systèmes de tâches temps réel

Florent Peres
Directeurs: Bernard Berthomieu, François Vernadat
Tuteur industriel: Patrick Farail

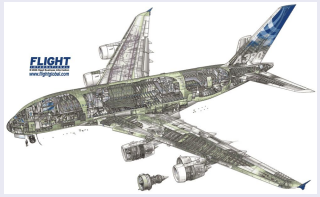
CIFRE Airbus/LAAS-CNRS

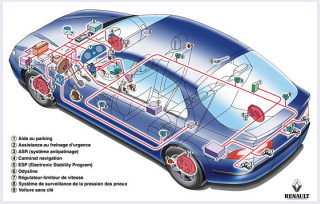
26 Janvier 2010

Plan

Contexte d'étude

Systèmes comprenant du logiciel critique pour la sécurité

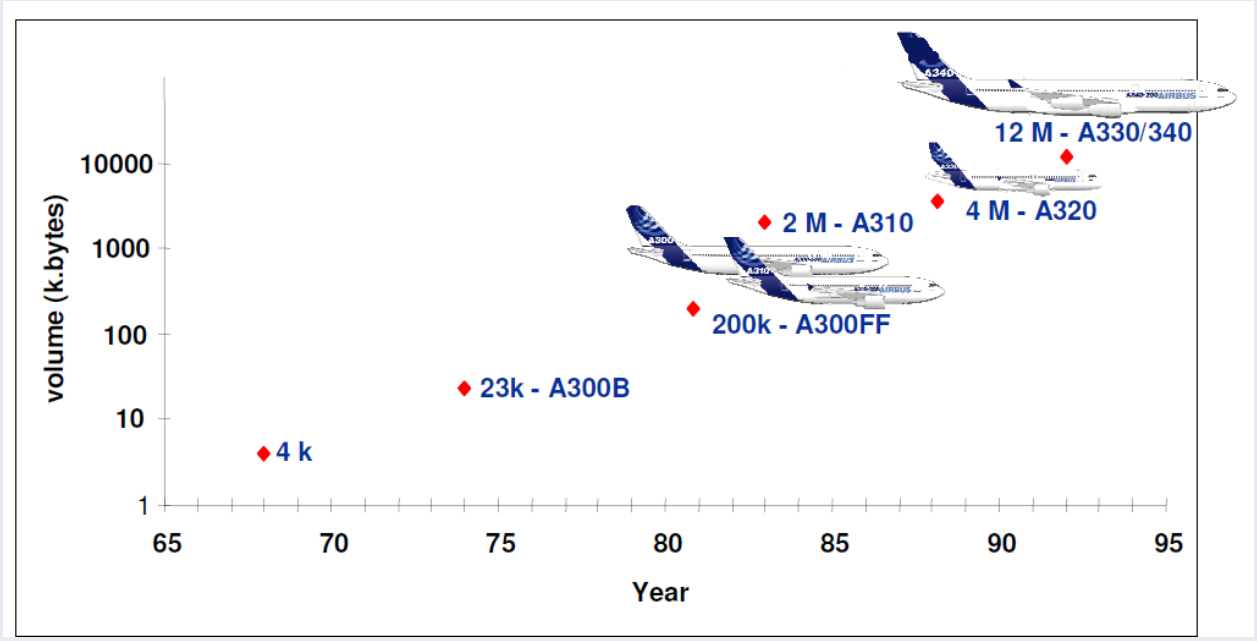


,

,

,

, ...

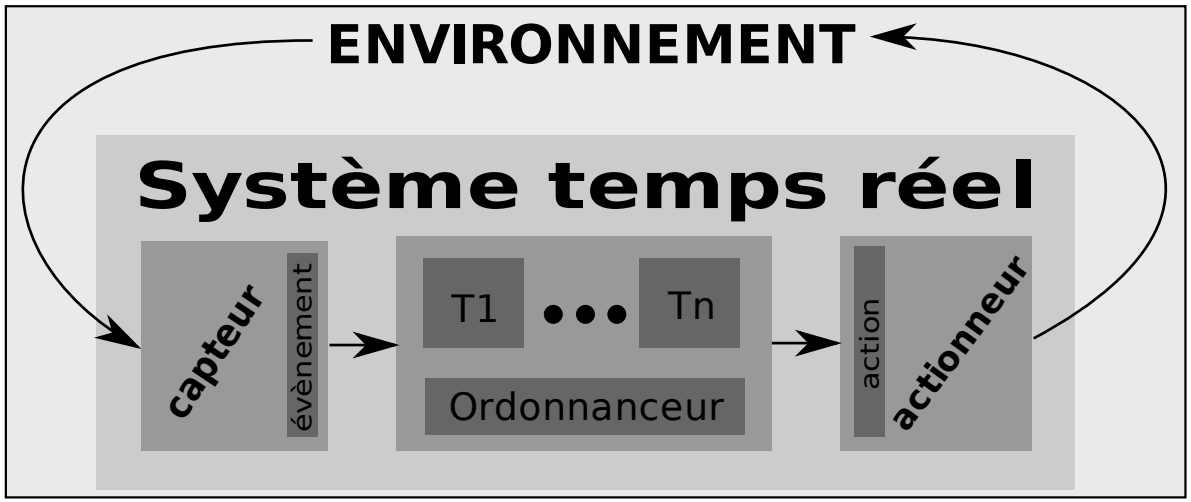
Augmentation du volume logiciel dans l'avionique



Contexte d'étude

Systèmes temps réel critiques:

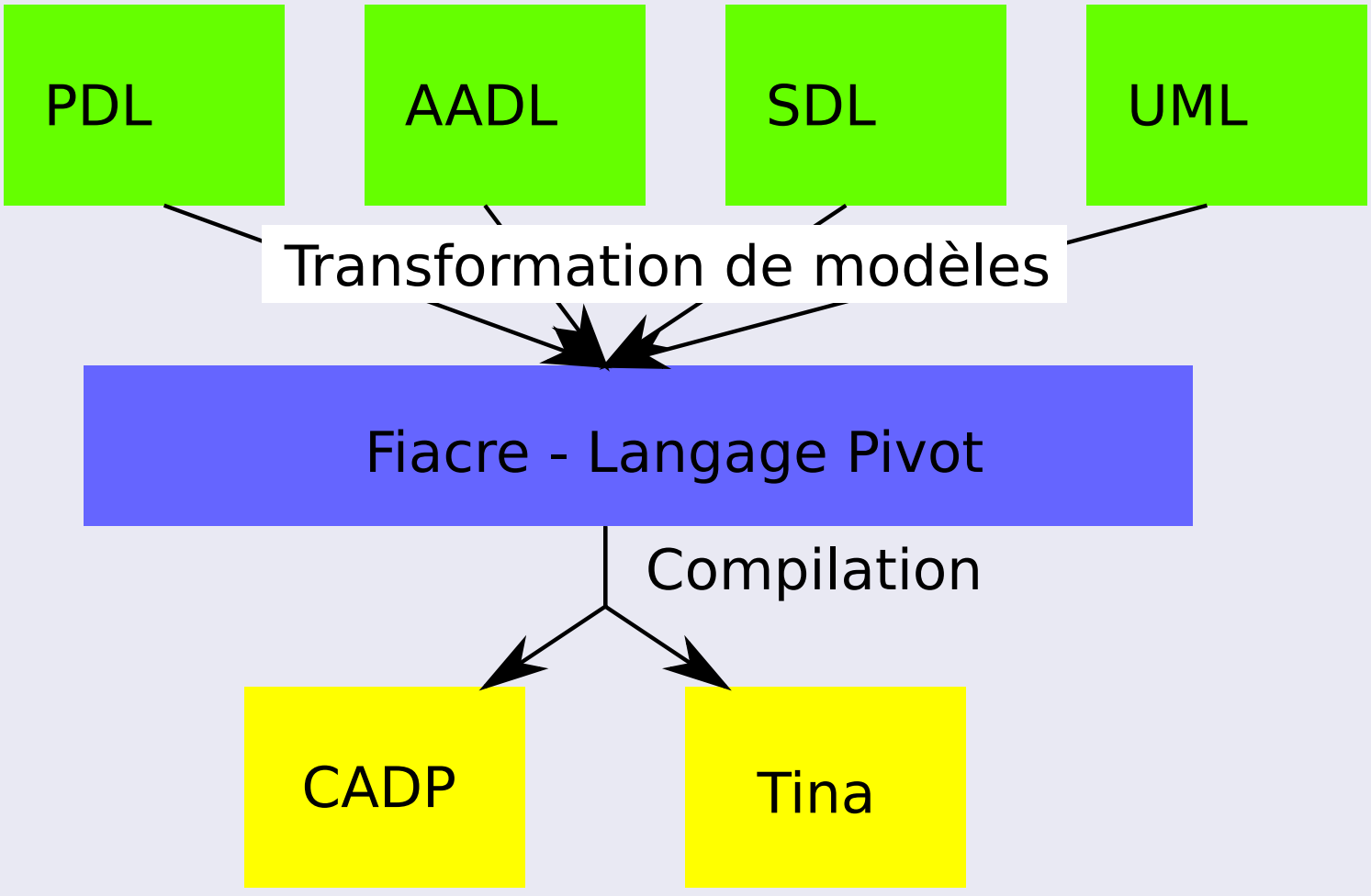
- interagissent avec un environnement non coopératif
  - peu d'hypothèses sur l'environnement
  - la réaction des tâches est temporellement contrainte



- critiques pour la sécurité des personnes
  - confiance ⇒ vérification (tests, preuves, ...)
  - non respect des contraintes temporelles = BUG
  - la simplicité est préférée

Contexte d'étude

Toolkit in **O**pen Source for **C**ritical **A**pplications & **S**ystems **D**evelopment



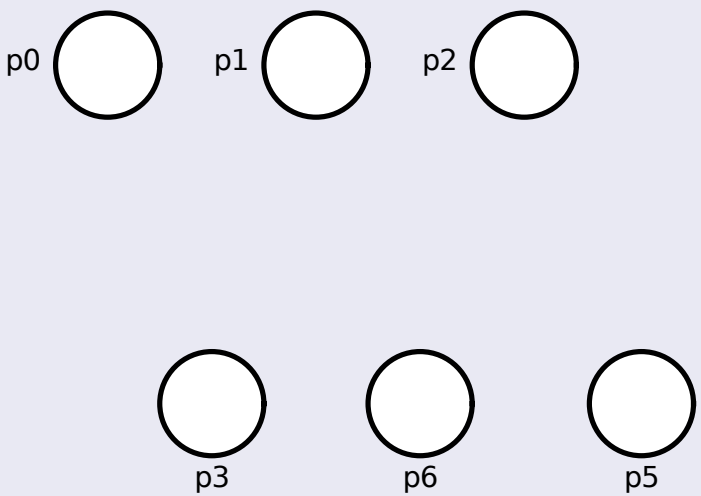
Plan

Plan

Réseaux de Petri temporels (TPN)

Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P,$ $\rangle$

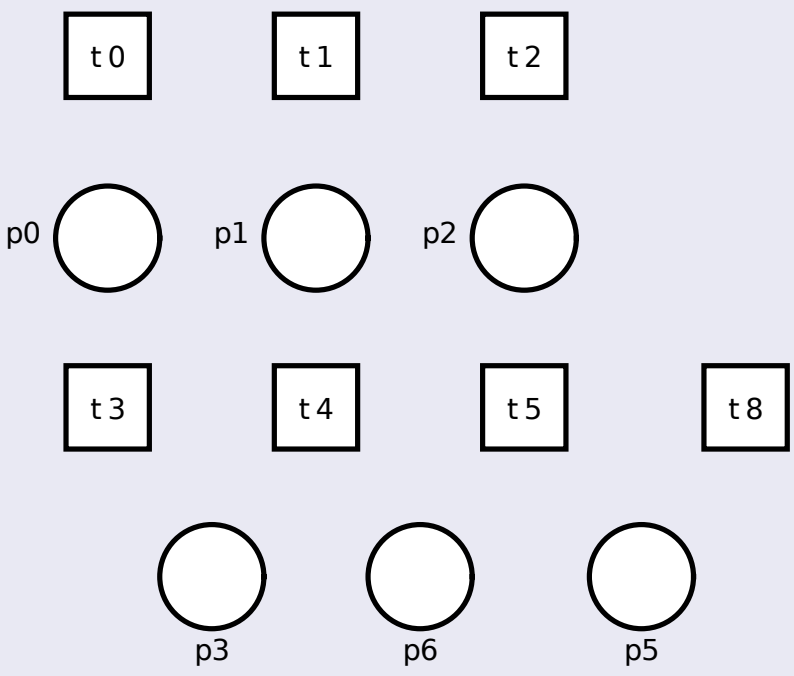


- P : ensemble de places

Réseaux de Petri temporels (TPN)

Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P, T,$ $\rangle$

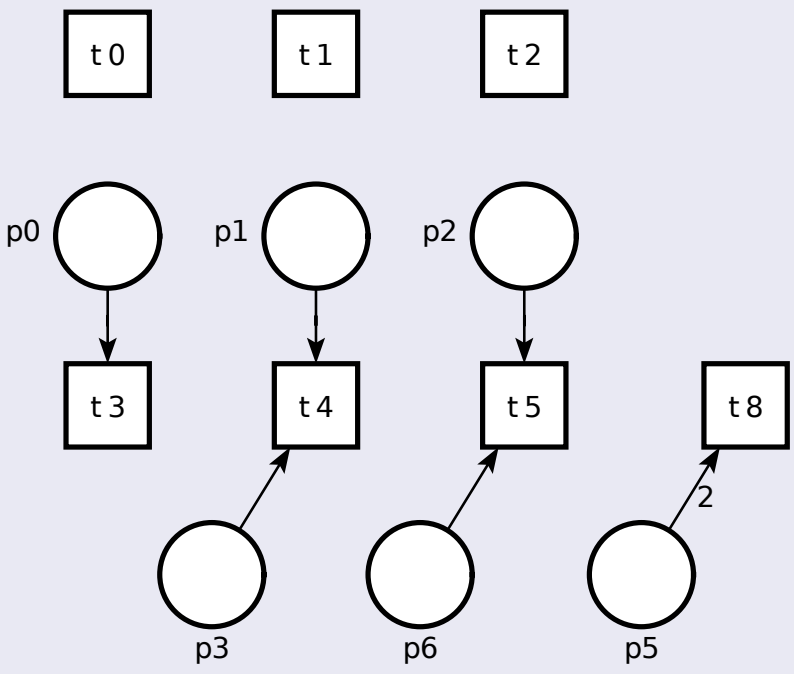


- P, T: ensemble de places et de transitions

Réseaux de Petri temporels (TPN)

Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P, T, Pre, \quad \rangle$

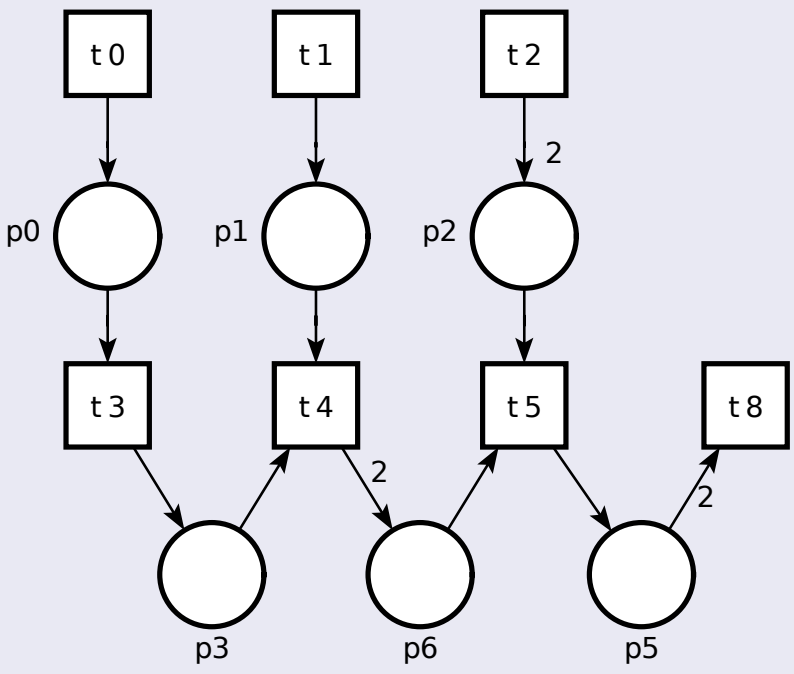


- P, T: ensemble de places et de transitions
- Pre : pré -conditions

Réseaux de Petri temporels (TPN)

Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P, T, Pre, Post, \quad \rangle$

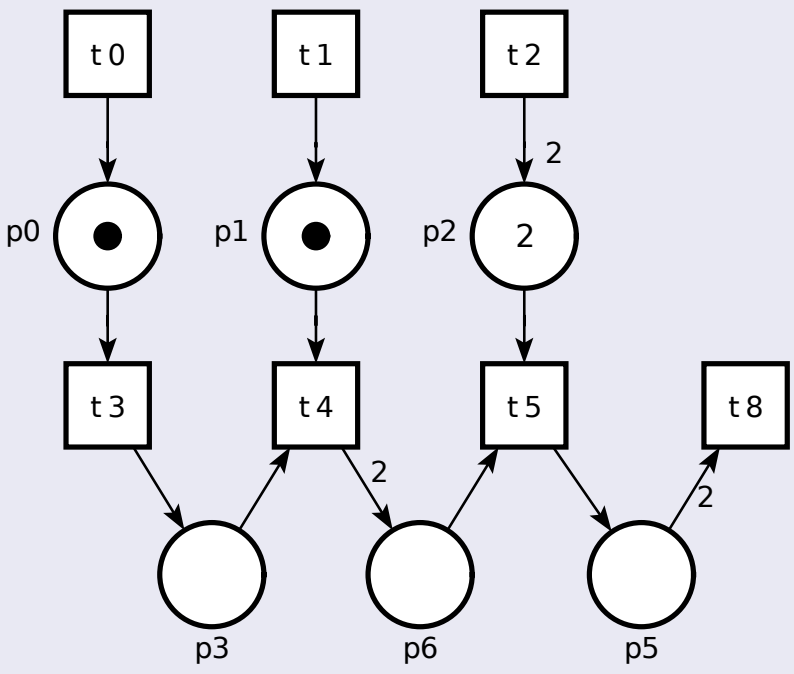


- P, T: ensemble de places et de transitions
- Pre, Post: pré/post-conditions

Réseaux de Petri temporels (TPN)

Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P, T, Pre, Post, \mathcal{M}_0, \quad \rangle$

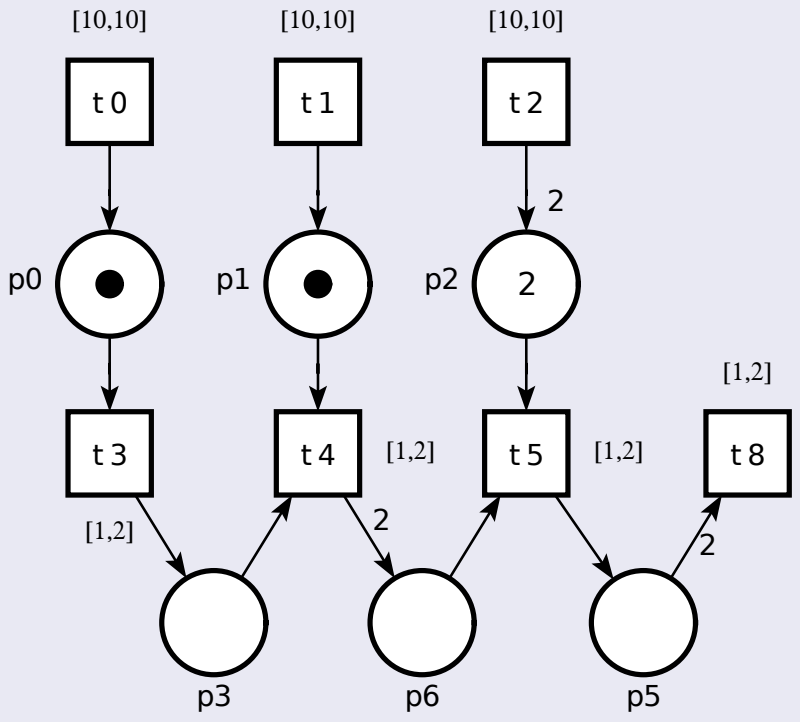


- P, T: ensemble de places et de transitions
- Pre, Post: pré/post-conditions
- $\mathcal{M}_0$: marquage initial

Réseaux de Petri temporels (TPN)

Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P, T, Pre, Post, \mathcal{M}_0, I_S, \quad \rangle$

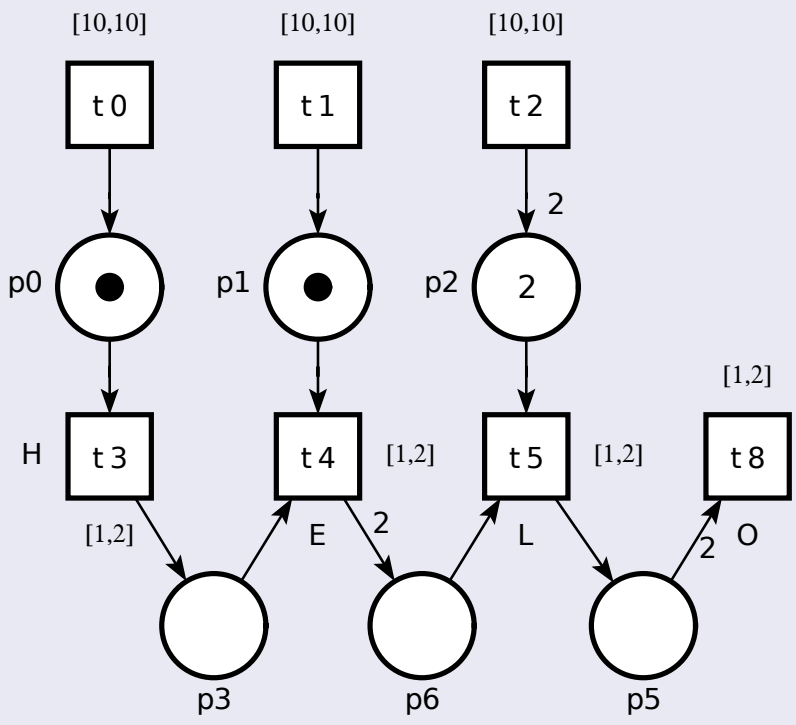


- P, T: ensemble de places et de transitions
- Pre, Post: pré/post-conditions
- $\mathcal{M}_0$: marquage initial
- I_S: fonction intervalle de tir statique

Réseaux de Petri temporels (TPN)

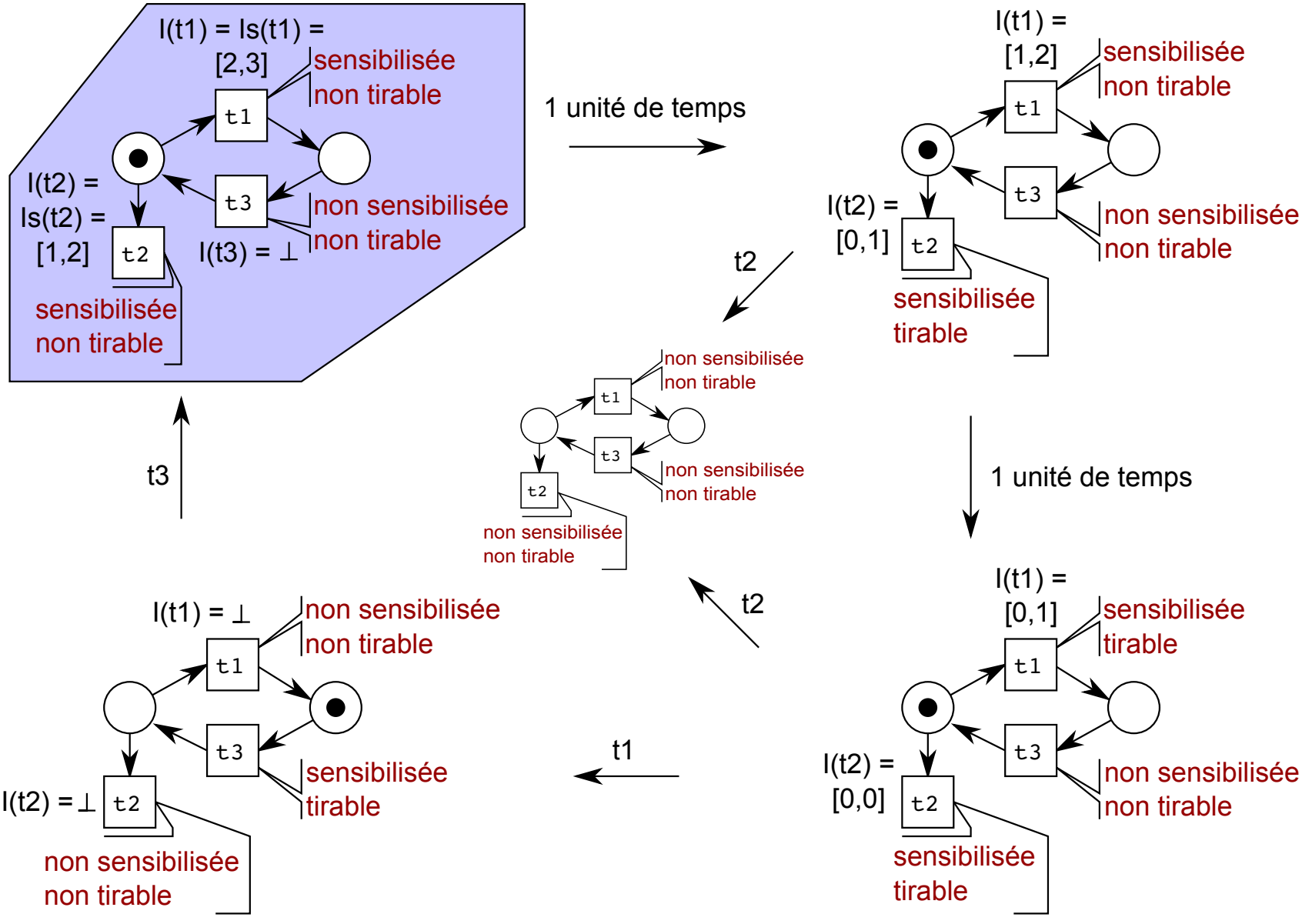
Un réseau de Petri temporel (Merlin, 1974)

est un n-uplet: $\langle P, T, Pre, Post, \mathcal{M}_0, I_S, \Sigma, \mathcal{L} \rangle$

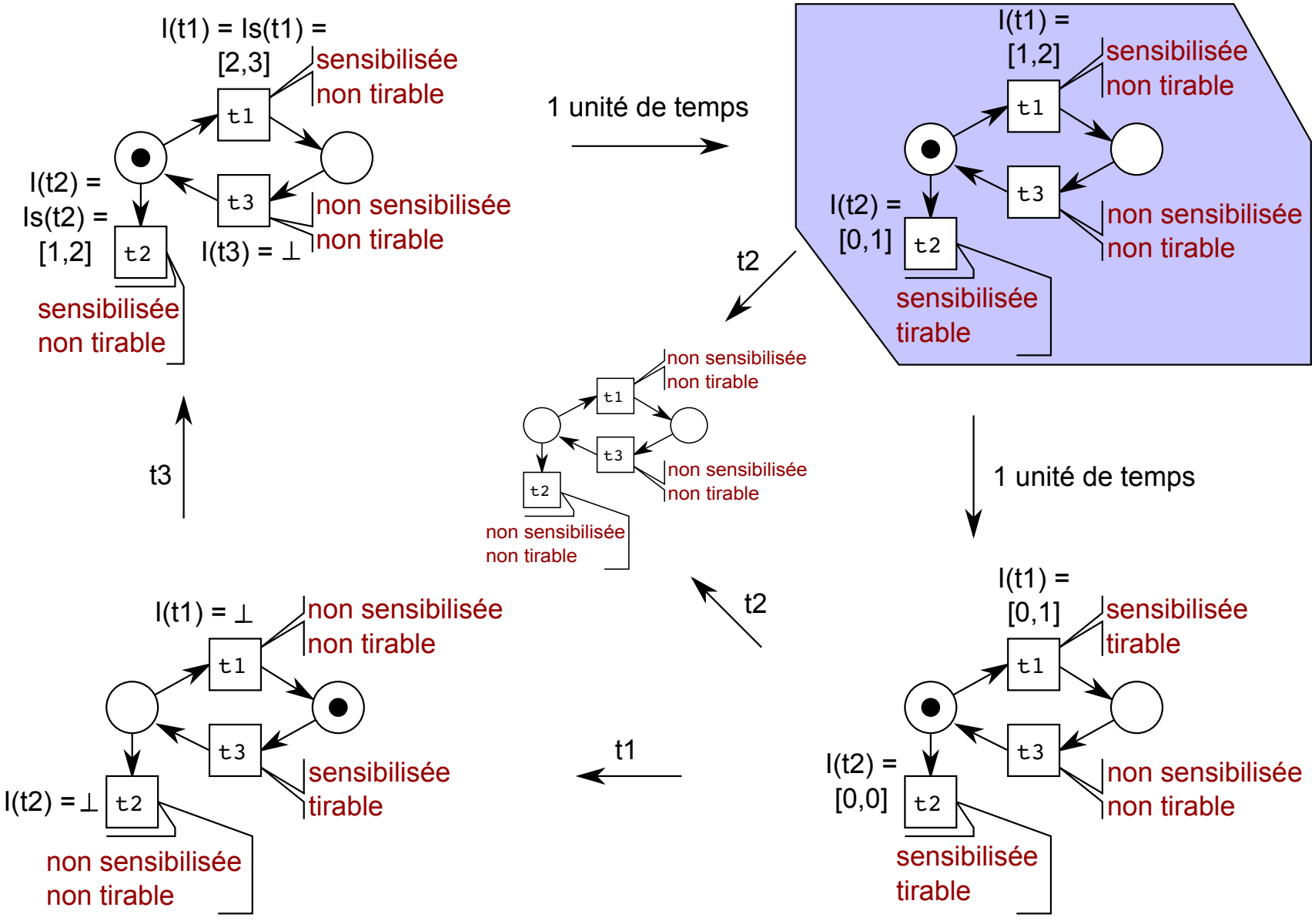


- P, T: ensemble de places et de transitions
- Pre, Post: pré/post-conditions
- $\mathcal{M}_0$: marquage initial
- I_S: fonction intervalle de tir statique
- Σ: alphabet d'actions
- $\mathcal{L}$: fonction d'étiquetage

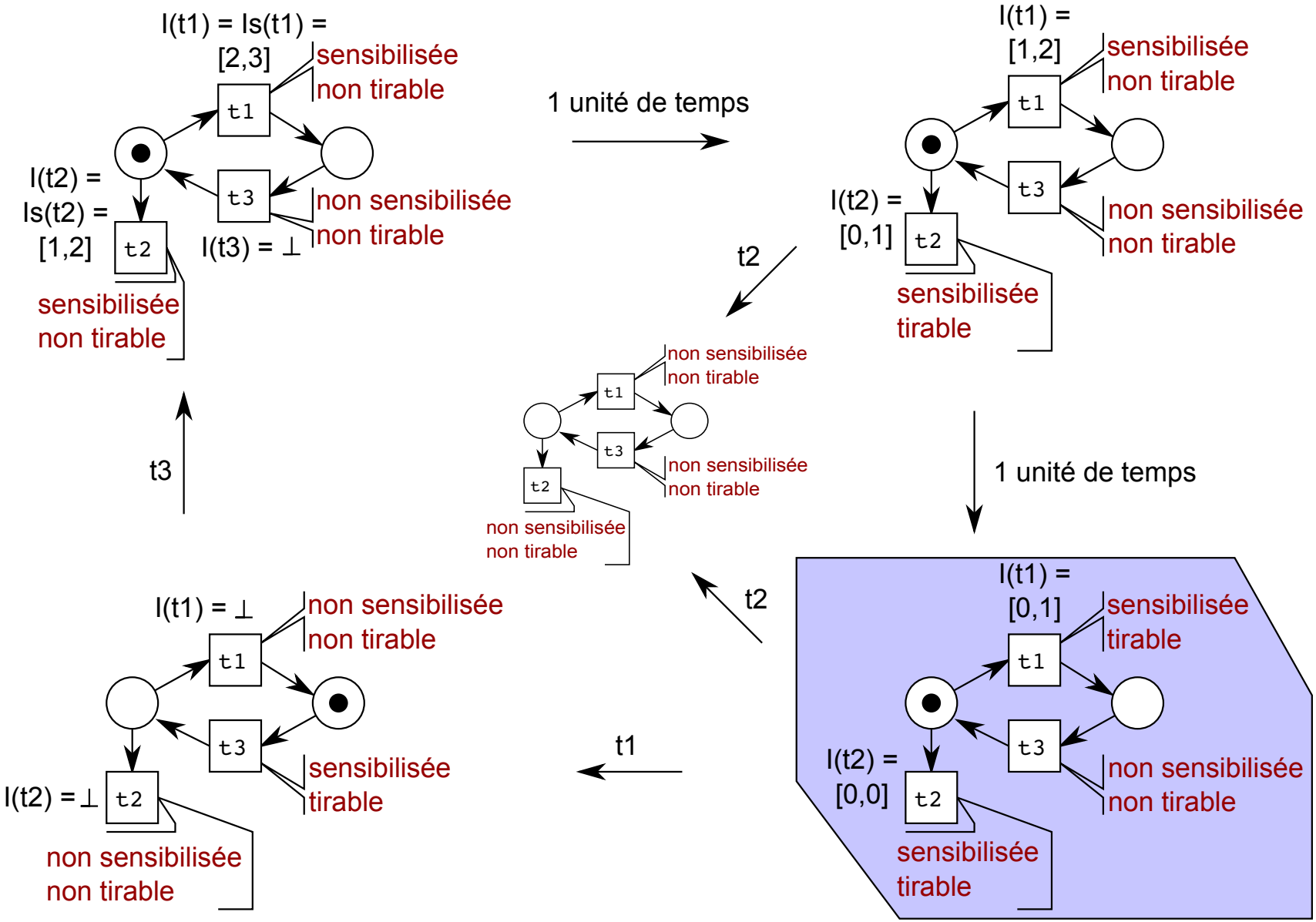
Dynamique des réseaux de Petri temporels



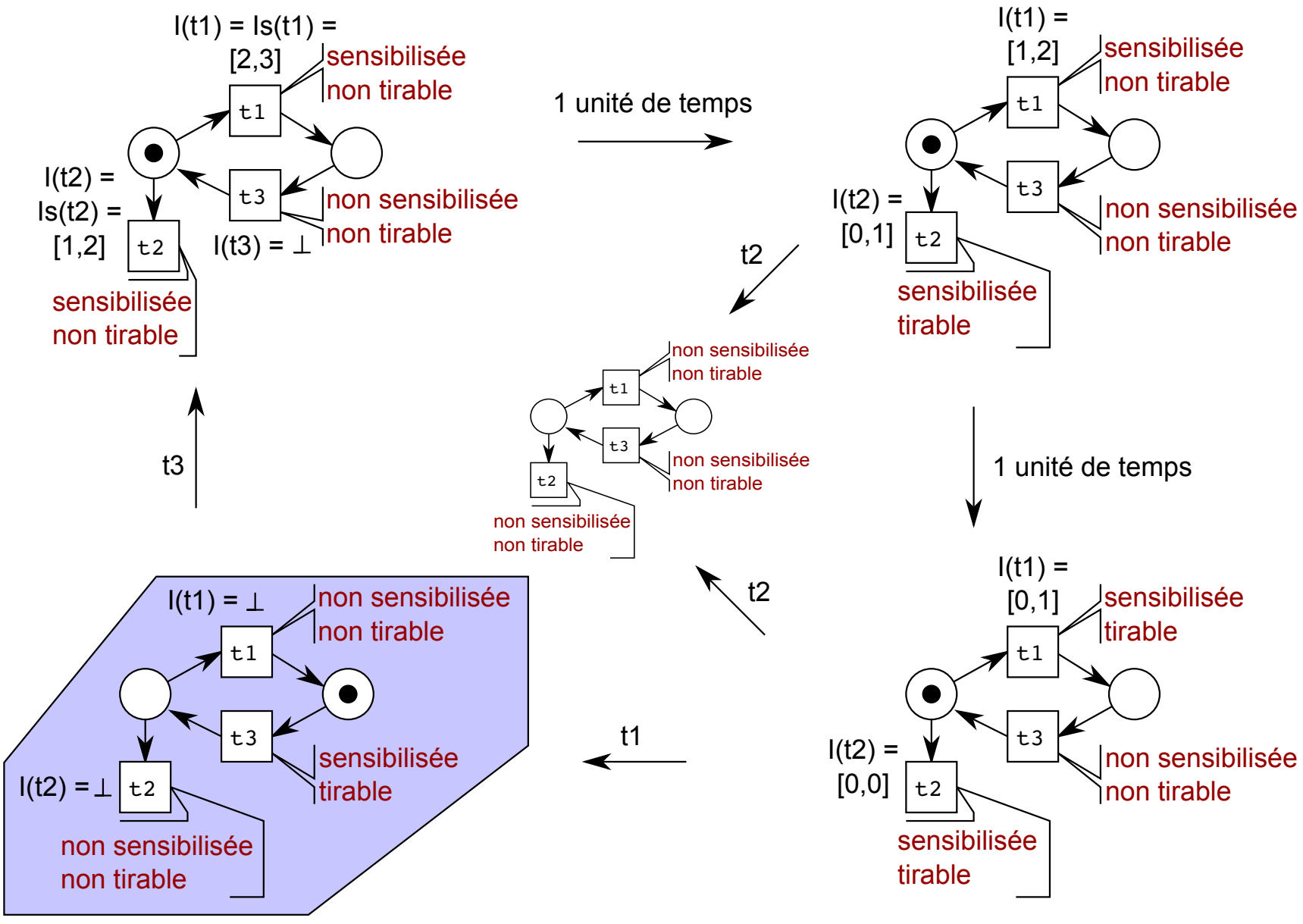
Dynamique des réseaux de Petri temporels



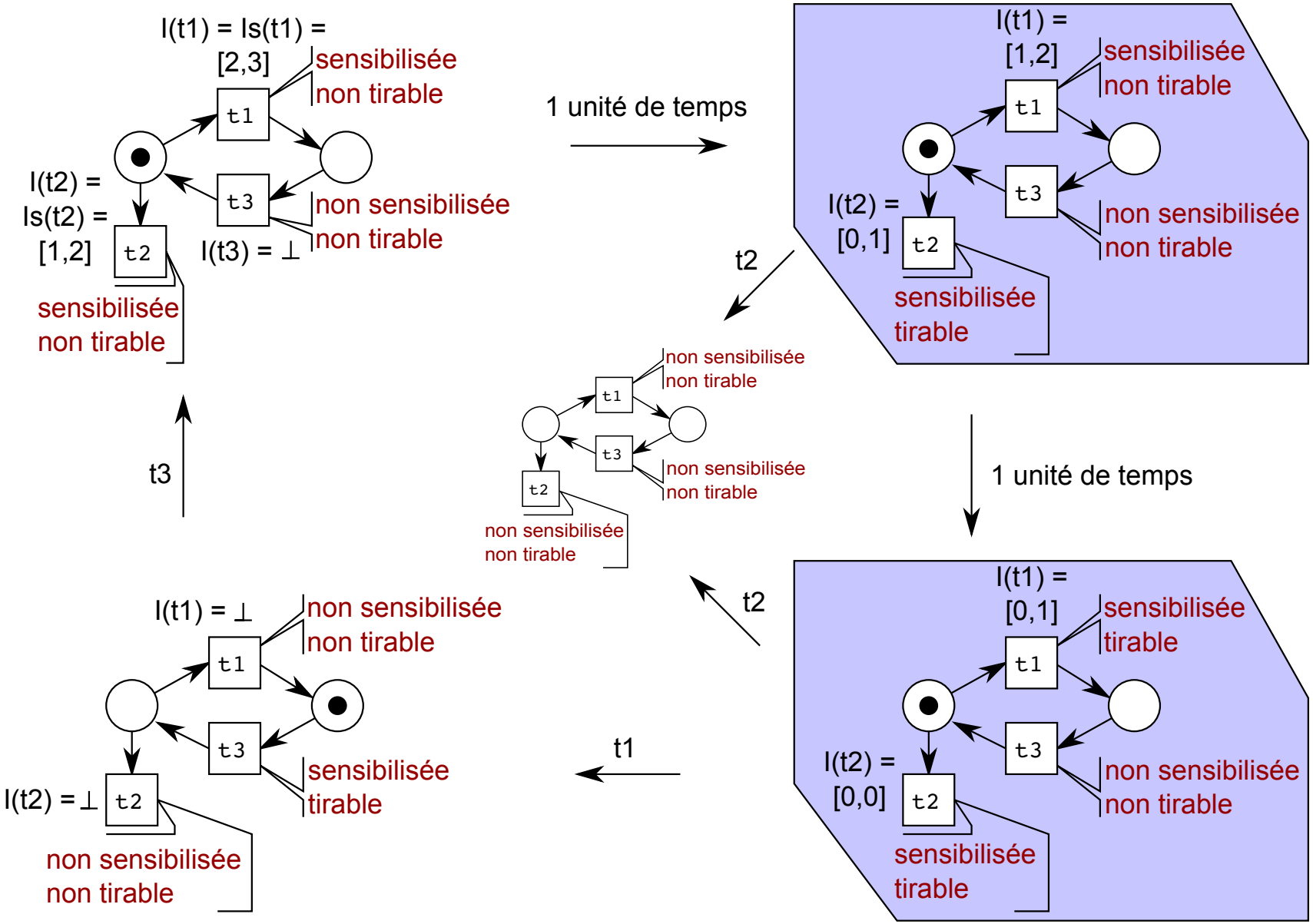
Dynamique des réseaux de Petri temporels



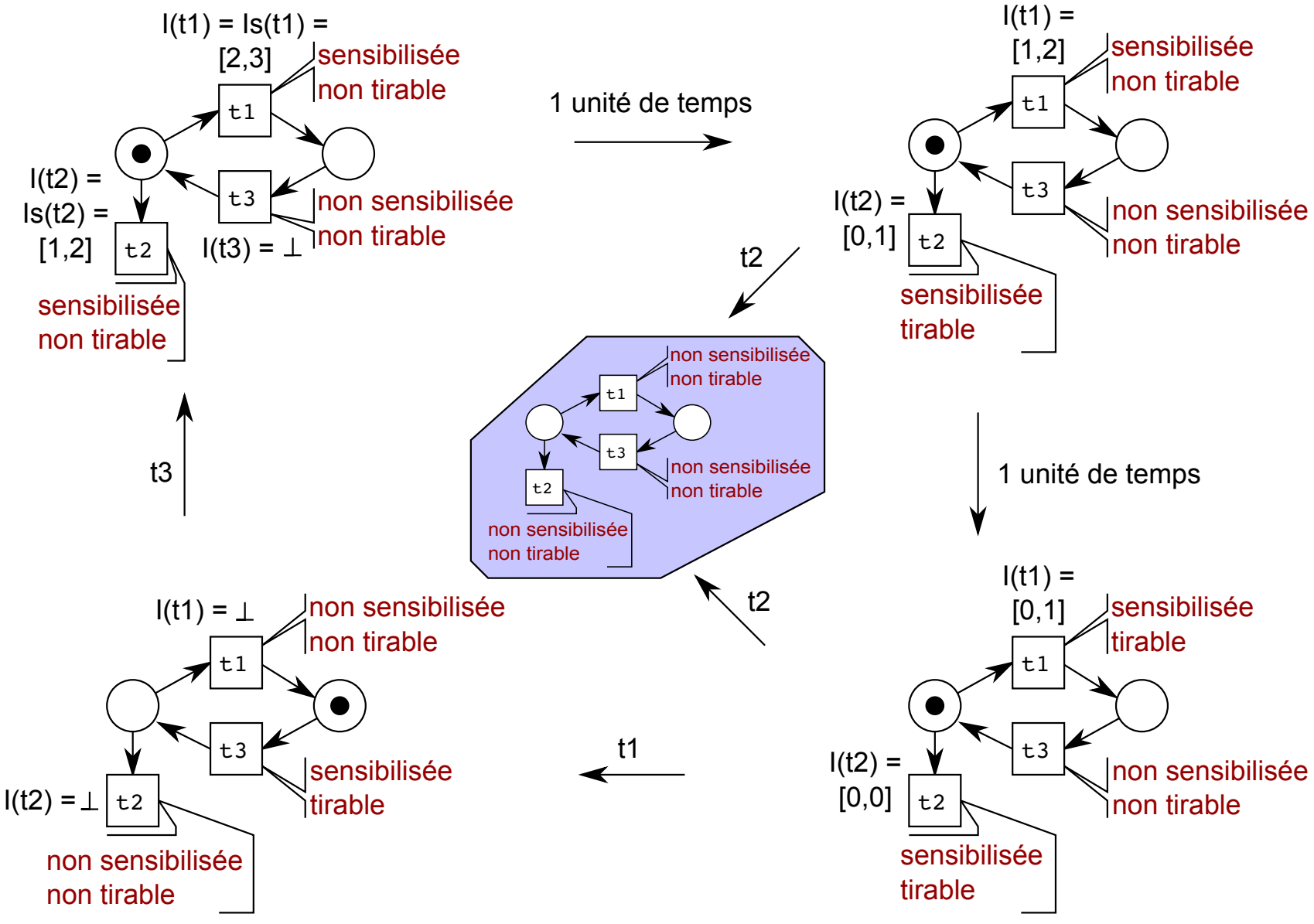
Dynamique des réseaux de Petri temporels



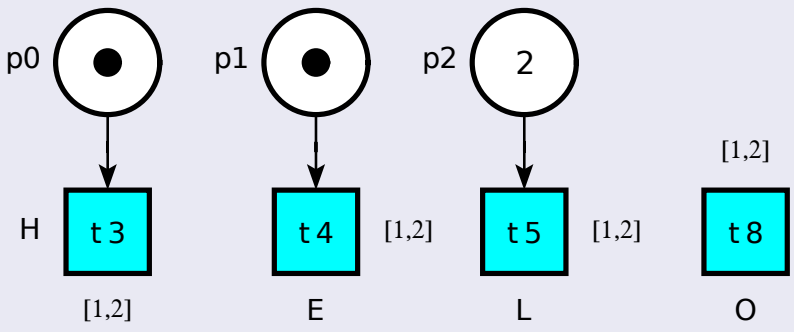
Dynamique des réseaux de Petri temporels



Dynamique des réseaux de Petri temporels



Les TPN pour les systèmes temps réel



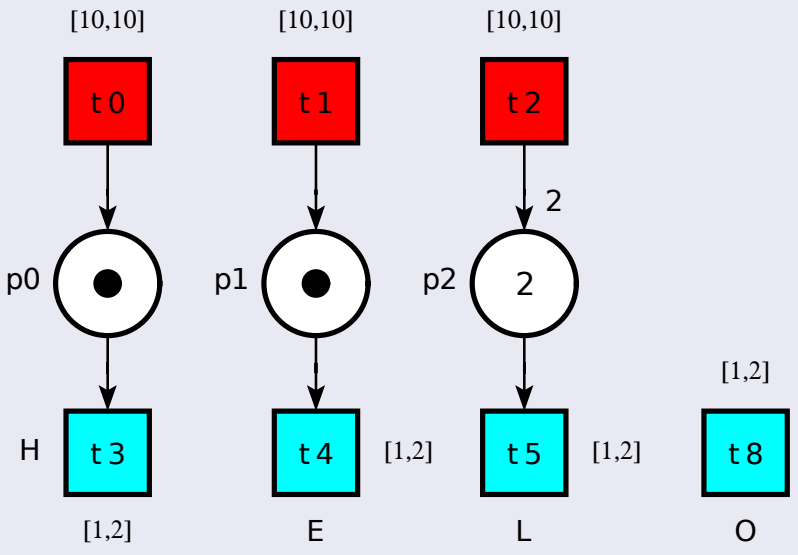
Légende

- : tâches

Comportement

$(H.E.L.L.O)^{\omega}$

Les TPN pour les systèmes temps réel



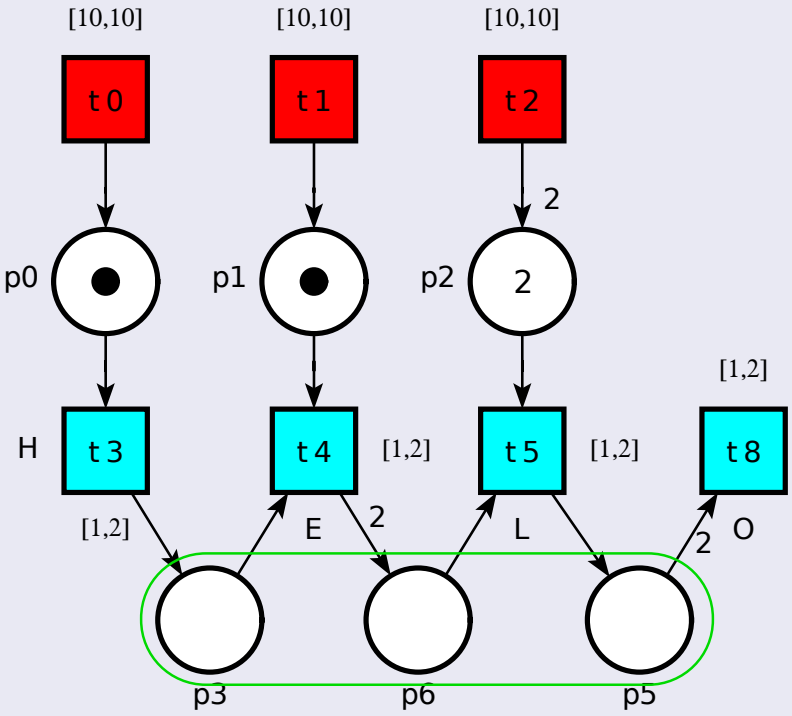
Légende

- : tâches
- : périodes

Comportement

$(H.E.L.L.O)^{\omega}$

Les TPN pour les systèmes temps réel



Légende

- : tâches
- : périodes
- : dépendances entre tâches

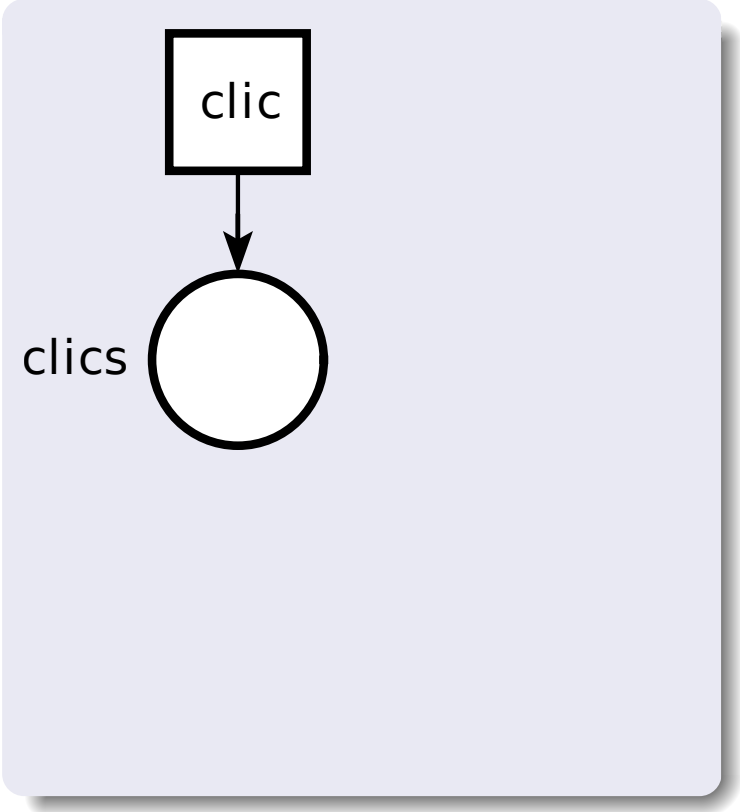
Comportement

$(H.E.L.L.O)^{\omega}$

Le double clic: un exemple problématique

Enoncé

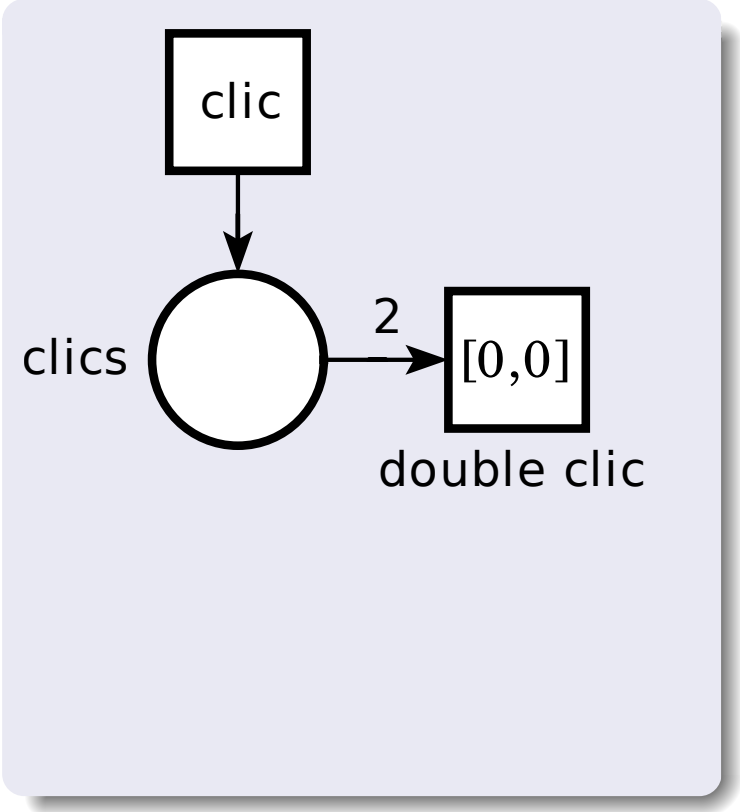
Si deux évènements *clic* se succèdent en moins de trois unités de temps, alors générer un *double clic*, sinon générer deux *simple clic*.



Le double clic: un exemple problématique

Enoncé

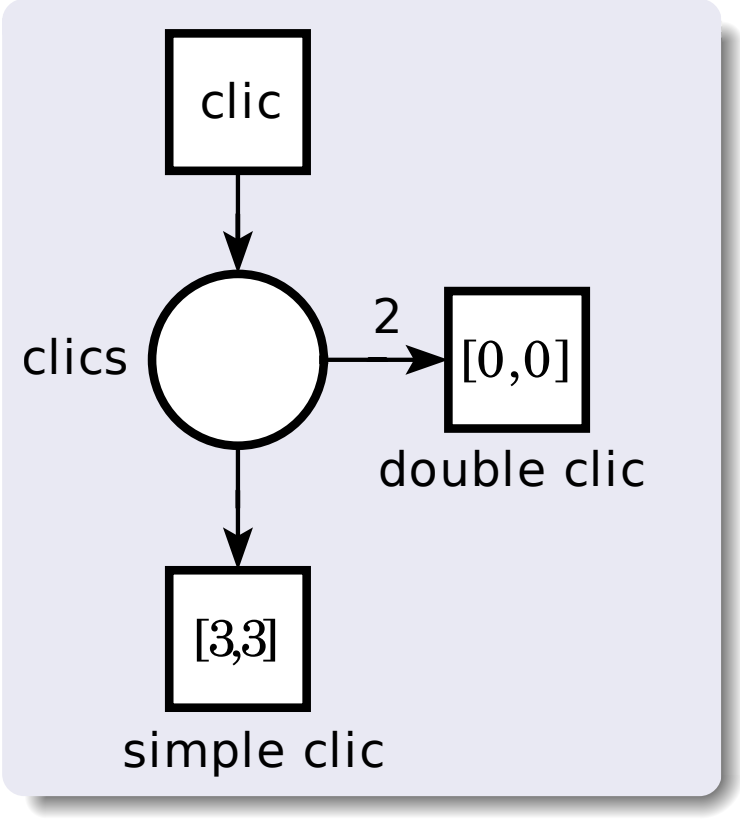
Si deux évènements *clic* se succèdent en moins de trois unités de temps, alors générer un *double clic*, sinon générer deux *simple clic*.



Le double clic: un exemple problématique

Enoncé

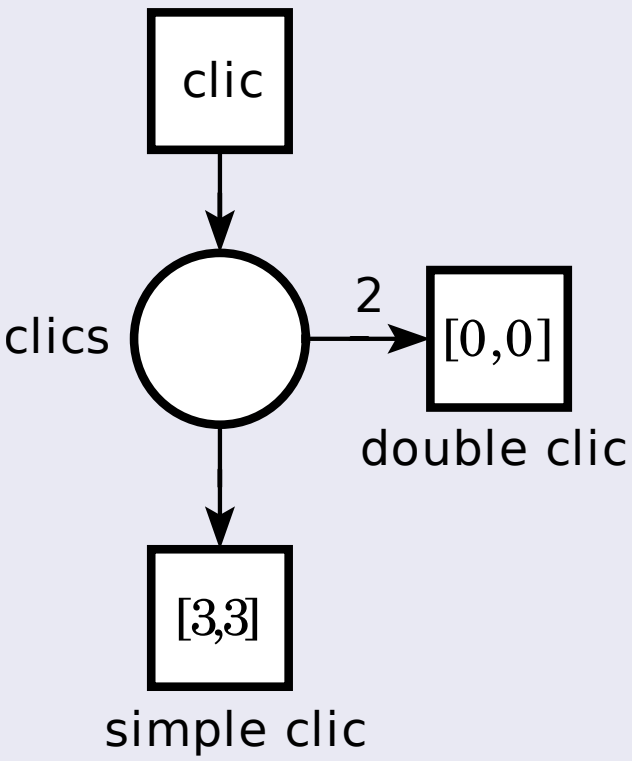
Si deux évènements *clic* se succèdent en moins de trois unités de temps, alors générer un *double clic*, sinon générer deux *simple clic*.



Le double clic: un exemple problématique

Enoncé

Si deux évènements *clic* se succèdent en moins de trois unités de temps, alors générer un *double clic*, sinon générer deux *simple clic*.



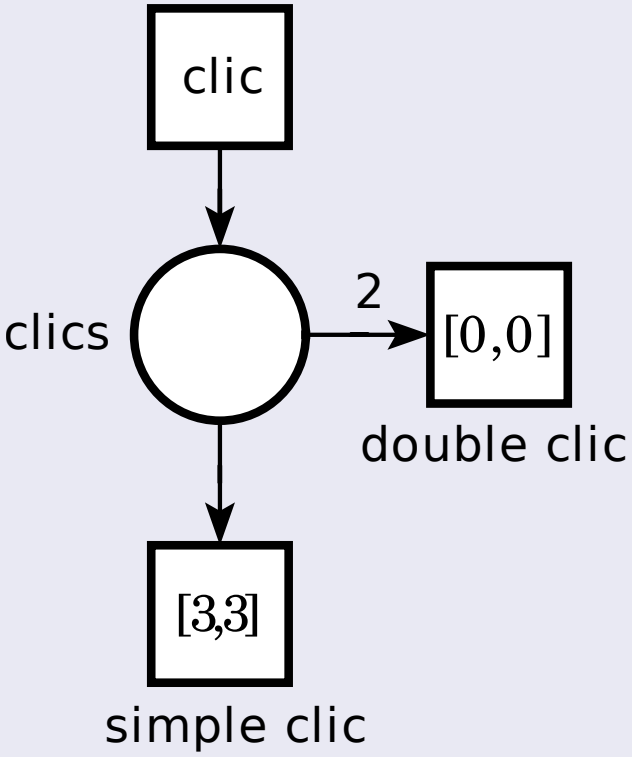
Problème: choix $\notin$ énoncé

double clic toujours possible à 3

Le double clic: un exemple problématique

Enoncé

Si deux évènements *clic* se succèdent en moins de trois unités de temps, alors générer un *double clic*, sinon générer deux *simple clic*.



Problème: choix $\notin$ énoncé

double clic toujours possible à 3

Caractéristique des TPN

Empêcher le tir d'une transition tirable revient à la désensibiliser

Conséquence

Impossible de coder cet exemple tout en conservant le choix *simple*/*double* clic tout au long des exécutions.

Plan

Réseaux de Petri temporels à inhibitions/permissions (ipTPN)

Définition

Un ipTPN est un tuple $\langle P, T, Pre, Post, \mathcal{M}_0, I_S, F, A, \Sigma, \mathcal{L} \rangle$ où

- $\langle P, T, Pre, Post, \mathcal{M}_0, I_S, \Sigma, \mathcal{L} \rangle$ est un TPN
- $F : T \times T$ est la *relation d'inhibition*, notée $\multimap$
- $A : T \times T$ est la *relation de permission*, notée $\bullet\!\!-$

T-sensibilisation

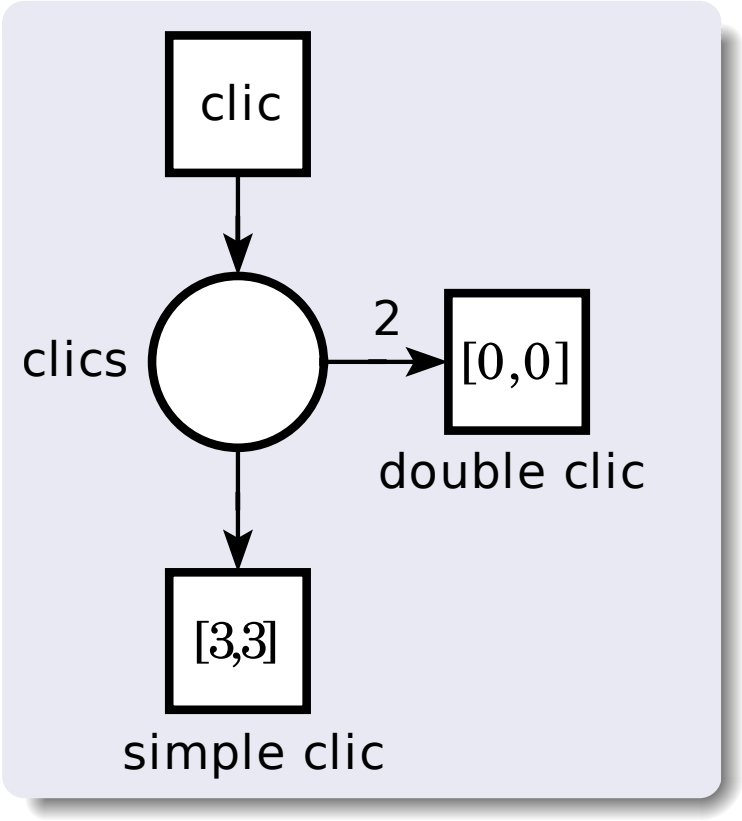
Une transition t est T-sensibilisée ssi elle sensibilisée et $0 \in I(t)$

Sémantique (ajout à celle des TPN)

Ajout des règles suivantes, contraignant le tir de la transition t:

- $(\forall i \in T)(i \multimap t \Rightarrow 0 \notin I(i))$
- $(\forall j \in T)(j \bullet\!\!- t \Rightarrow 0 \in I(j))$

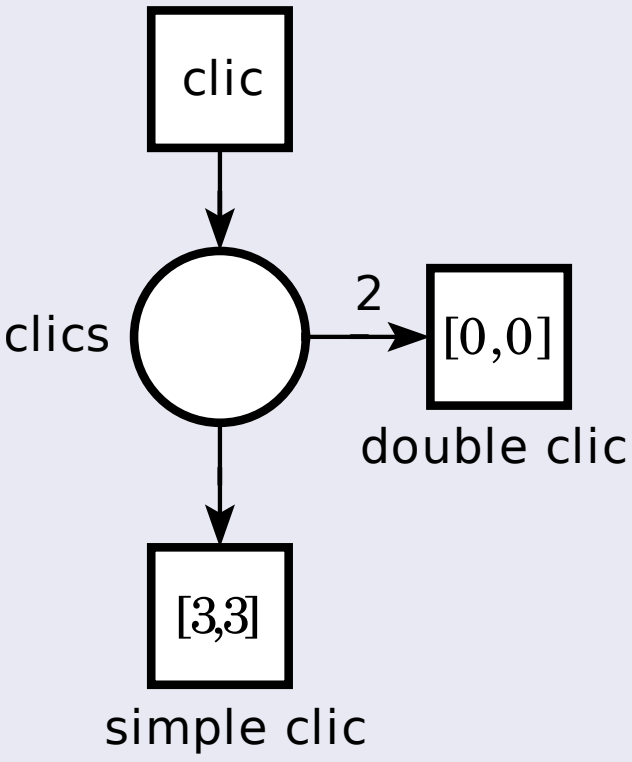
Double clic revisité



Problème

double clic toujours possible à 3

Double clic revisité



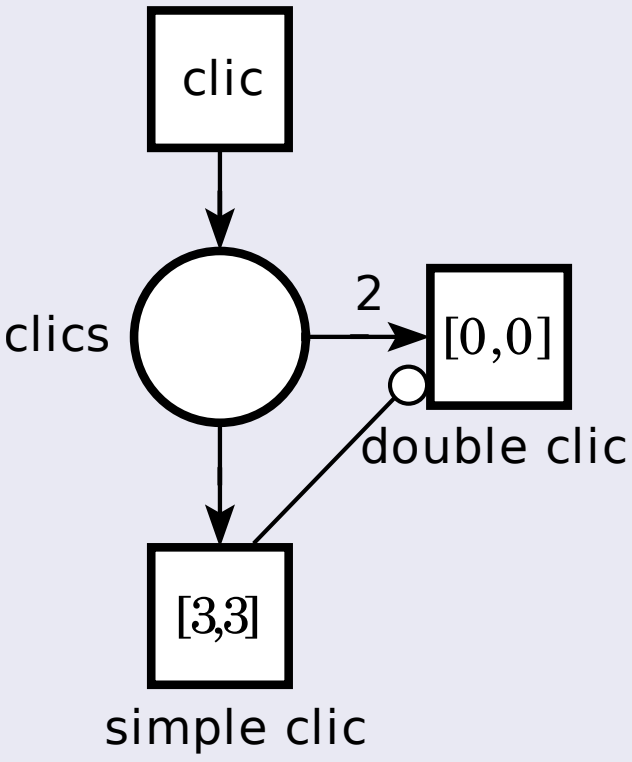
Problème

double clic toujours possible à 3

Solution

Lorsque *simple clic* est tirable, empêcher le tir de *double clic*

Double clic revisité



Problème

double clic toujours possible à 3

Solution

Lorsque *simple clic* est tirable, empêcher le tir de *double clic*

$\Leftrightarrow$ Ajouter la contrainte *simple clic* $\multimap$ *double clic*

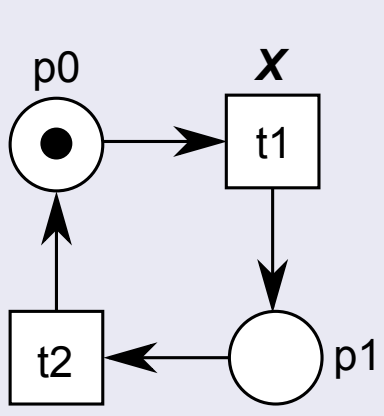
Composition des réseaux de Petri

Composition parallèle de réseaux

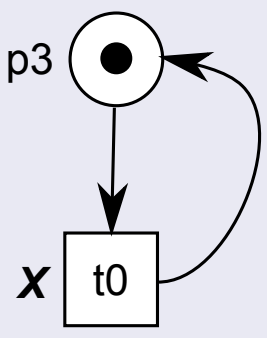
Réunion de deux réseaux selon les étiquettes portées par les transitions

Composition de transitions

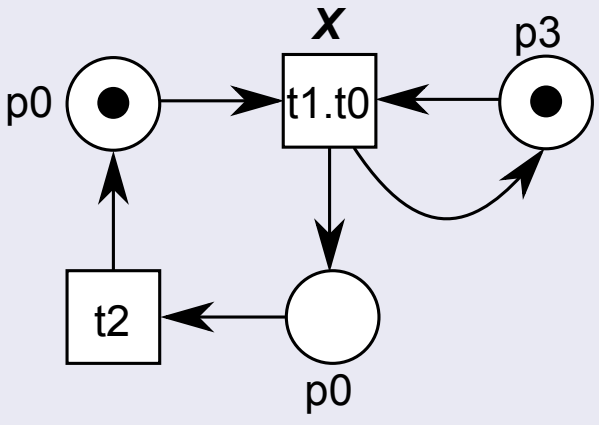
Fusion des informations des transitions participantes



Réseau A



Réseau B

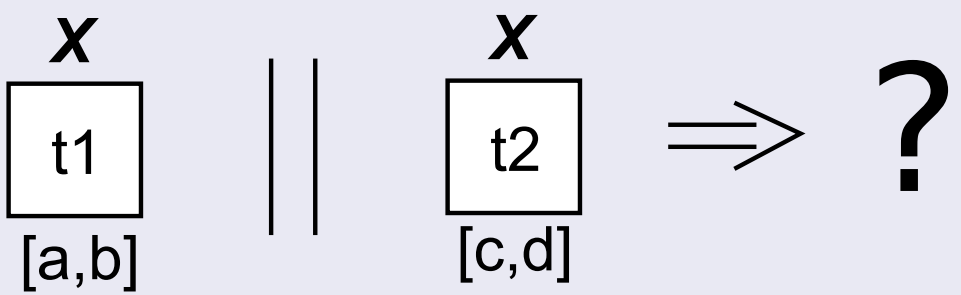


Réseau A || B

Composition des réseaux de Petri temporels

Problème

Les informations temporelles sont portées par les transitions.
Comment fusionner les intervalles? Pour quel résultat ?



Définition: TPN composable

Un TPN est composable si les transitions étiquetées ne portent pas d'informations temporelles

Rendre les TPN composables: (manière endogène)

Transformation TPN $\longrightarrow$ TPN composable est possible si

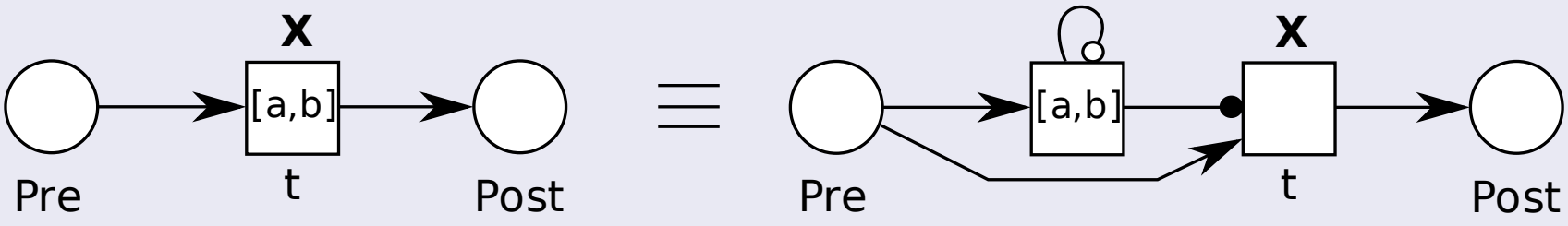
- borne de tir maximale non stricte
- réseau borné

Mais, cette transformation

- peut donner des réseaux exponentiellement plus gros
- ne préserve que la bisimulation temporisée faible

Rendre les TPN composables: (manière exogène)

Traduction TPN → ipTPN composable



Résultat

$(\forall x \in (ip)TPN)(\exists y \in ipTPN)(\lceil x \rceil \sim^{sb} \lceil y \rceil \wedge y \text{ composable})$, où $\lceil \alpha \rceil$ est le comportement de α

Comparaison TA vs TPN

Automates temporisés (TA)

Automate à états fini classique + horloges + gardes temporelles (+ invariants de locations)

Resultats

- $\text{TA}_{inv \in \{\leq, <\}} \sim^{tl}$ TPN (Bérard, Cassez, Haddad, Lime, Roux, 05)
- $\text{TA}_{inv \in \{\leq\}} \sim^{wb}$ PrTPN (Berthomieu, Peres, Vernadat, 06)
- $(\forall x \in TA)(\exists y \in ipTPN)(\lceil x \rceil \sim^{sb} \lceil y \rceil)$ (Peres, Berthomieu, Vernadat, 09)
- $\text{TA}_{inv \in \{\leq\}} \sim^{wb}$ ipTPN

Exploration exhaustive des états

Systèmes de différences

- Réseaux de Petri temporels (TPN)
- Réseaux de Petri temporels à priorités (PrTPN)
- Réseaux de Petri temporels à inhibitions/permissions (ipTPN)

Polyèdres

Tout TPN ou PrTPN ou ipTPN étendu avec les arcs chronomètres

Implémentation

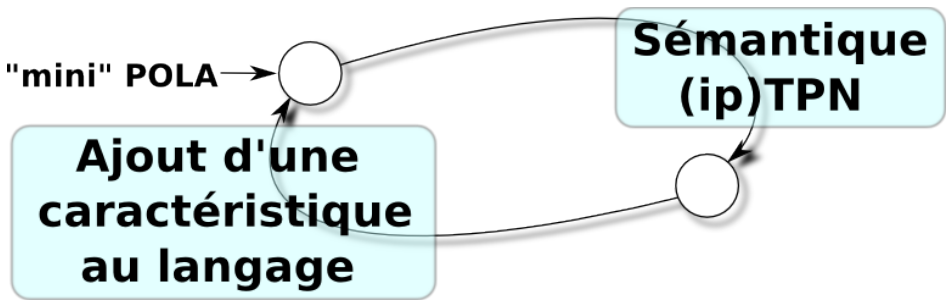
PrTPN, swPrTPN, ipTPN et swipTPN implémentés dans TINA

Plan

Pola et les systèmes de tâches temps réel

Double Objectif

- Modélisation et vérification **aisée** des STR
- Etude des forces et faiblesses des (ip)TPN en regard des STR



Manifeste de conception

- modélisable = vérifiable ($\neq$ AADL, MARTE, SysML)
- ordonnanceur $\in$ modèle ($\neq$ automates de tâches/Times)
- peu de concepts, mais les plus clairs et généraux possibles
- technique unique de vérification ($\neq$ méthodes analytiques)

Caractéristiques de POLA

```
system foo is
  res imprimante is not preempable
  res dualcore is preemptable pool of 2
  task t1 is
    ...
  end
  not preemptable task t2 is
    ...
  end
  policy RM is min P
  allocation allocation1 is
    ...
  allocation allocation2 is
    ...
end
```

Encapsulation système

Supporte la préemption

Caractéristiques de Pola

```
system foo is
  res imprimante is not preempable
  res dualcore is preemptable pool of 2
  task t1 is
    ...
  end
  not preemptable task t2 is
    ...
  end
  policy RM is min P
  allocation allocation1 is
    ...
  allocation allocation2 is
    ...
end
```

Ressources:
- préemptable ou non
- en groupe d'éléments indistinguables (pool) ou non

Caractéristiques de POLA

```
system foo is
  res imprimante is not preempable
  res dualcore is preemptable pool of 2
  task t1 is
    action a1 in [1,2] with allocation1
    period [5,5]
    deadline 4
    policy RM
    offset [2,3]
    level 13
  end
  not preemptable ta
    . . .
  end
  policy RM is min P
  allocation allocation
    . . .
  . . .
end
```

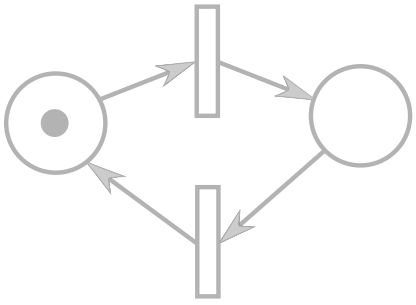
Tâches :
- actions = capacité de calcul
- périodicité = intervalle
- échéance
- politique
- décalage initial
- niveau défini par l'utilisateur

Caractéristiques de Pola

system foo **is**

. . .

not preemptable task t2 **is**
action a1 **in** [2,2] **with** allocation1
action a2 **in** [1,1] **with** allocation2 **endoftask**
period [10,10]
deadline 10
policy RM
behavior is



end

. . .

end

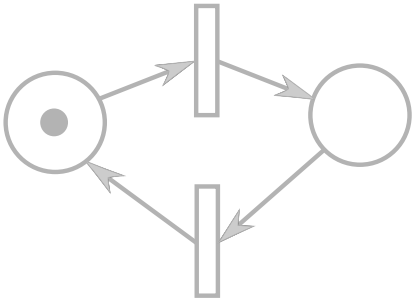
Tâche :
- peut être non préemptable
- plusieurs actions
- possibilité d'allouer différemment les ressources selon les actions
- comportements spécifiques

Caractéristiques de POLA

system foo **is**

. . .

not preemptable task t2 **is**
action a1 **in** [2,2] **with** allocation1
action a2 **in** [1,1] **with** allocation2 **endoftask**
period [10,10]
deadline 10
policy RM
behavior is

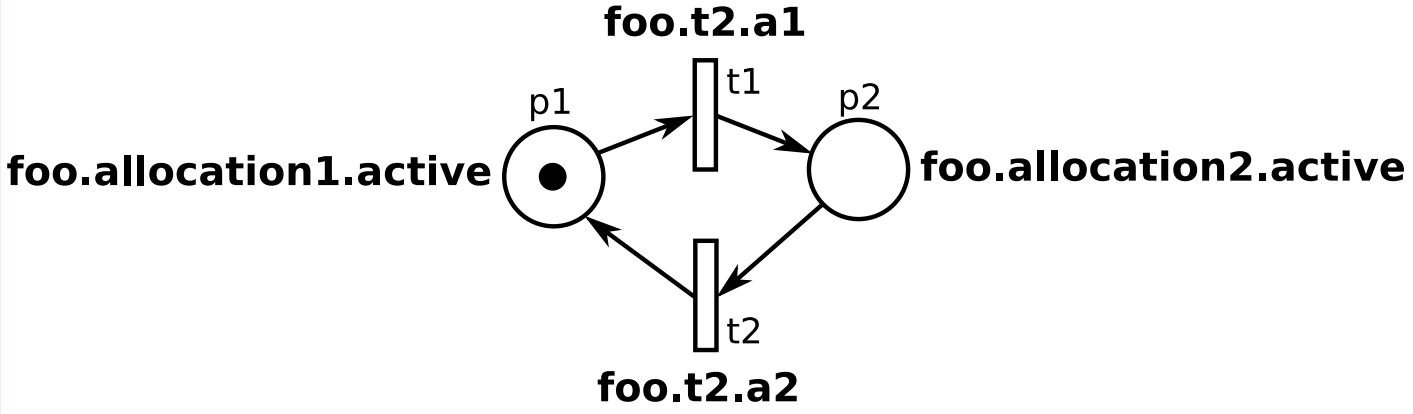


end

. . .

end

Comportements spécifiques



Caractéristiques de Pola

```
system foo is
  res imprimante is not preempable pool of 10 res
  res dualcore is pree
  task t1 is
    . . .
  end
  not preemptable t
    . . .
  end
  policy RM is min P
  allocation allocatio
    . . .
  allocation allocatio
    . . .
end
```

Politiques d'ordonnancement
(Migge, 1999)

Combinaison linéaire des caractéristiques statiques (P,D,C,L), triée selon min ou max

Rate Monotonic: min P

---

caractéristiques dynamiques (p,d,c)

Earliest Deadline First: min (D - d)

Caractéristiques de POLA

Allocations

Quelles ressources sont à allouer à quelles tâches ?

L'allocation ne connait pas le niveau action

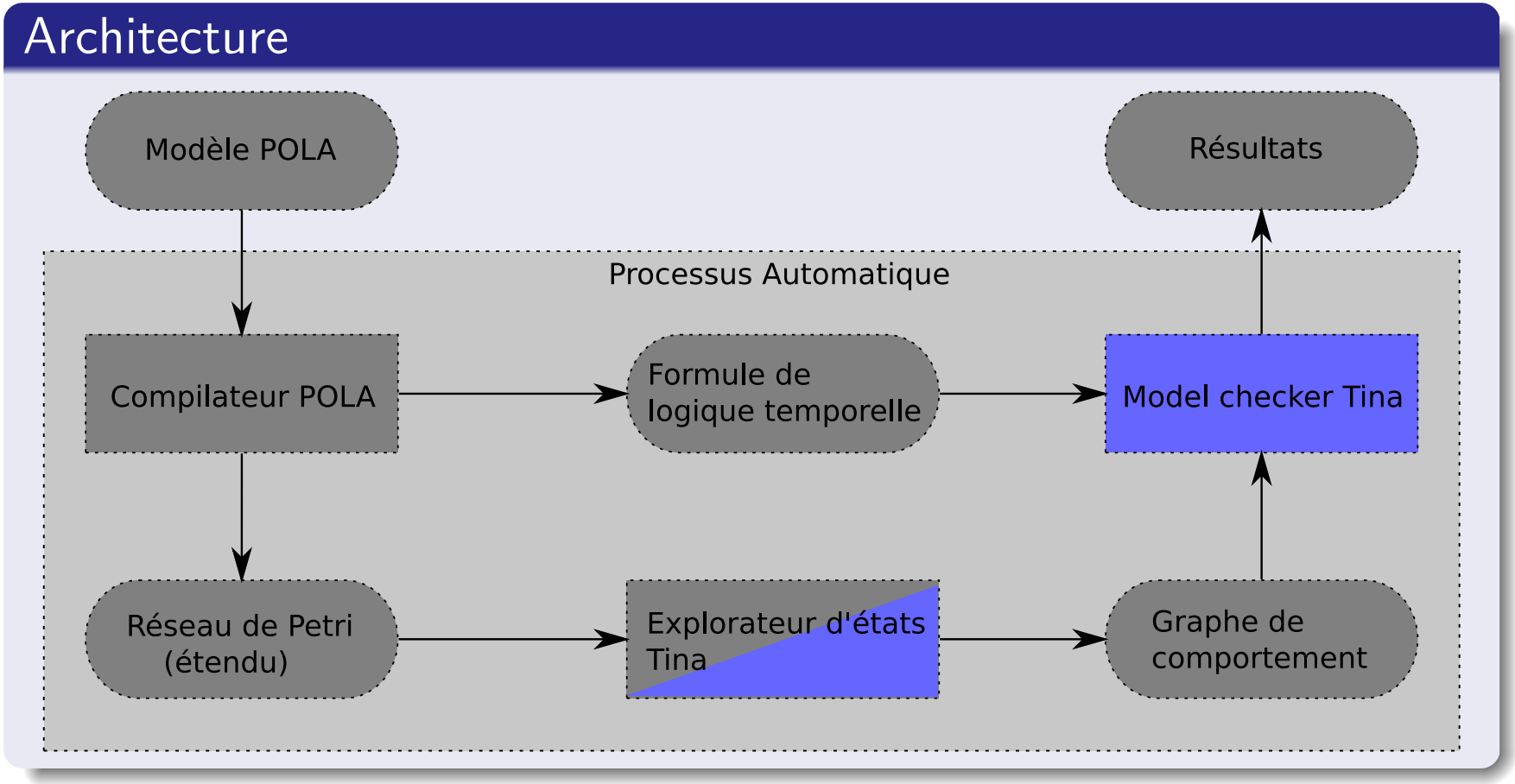
L'utilisateur doit se charger de leurs activations

```
system
  res im
  res du
  task t
    ...
  end
  not pr
    ...
  end
  policy RM is ////// ;
  allocation allocation1 is
    resources dualcore
    tasks t1,t2
  allocation allocation2 is
    resources imprimantes, dualcore
    tasks t2
.end
```

Plan

Chaine de vérification

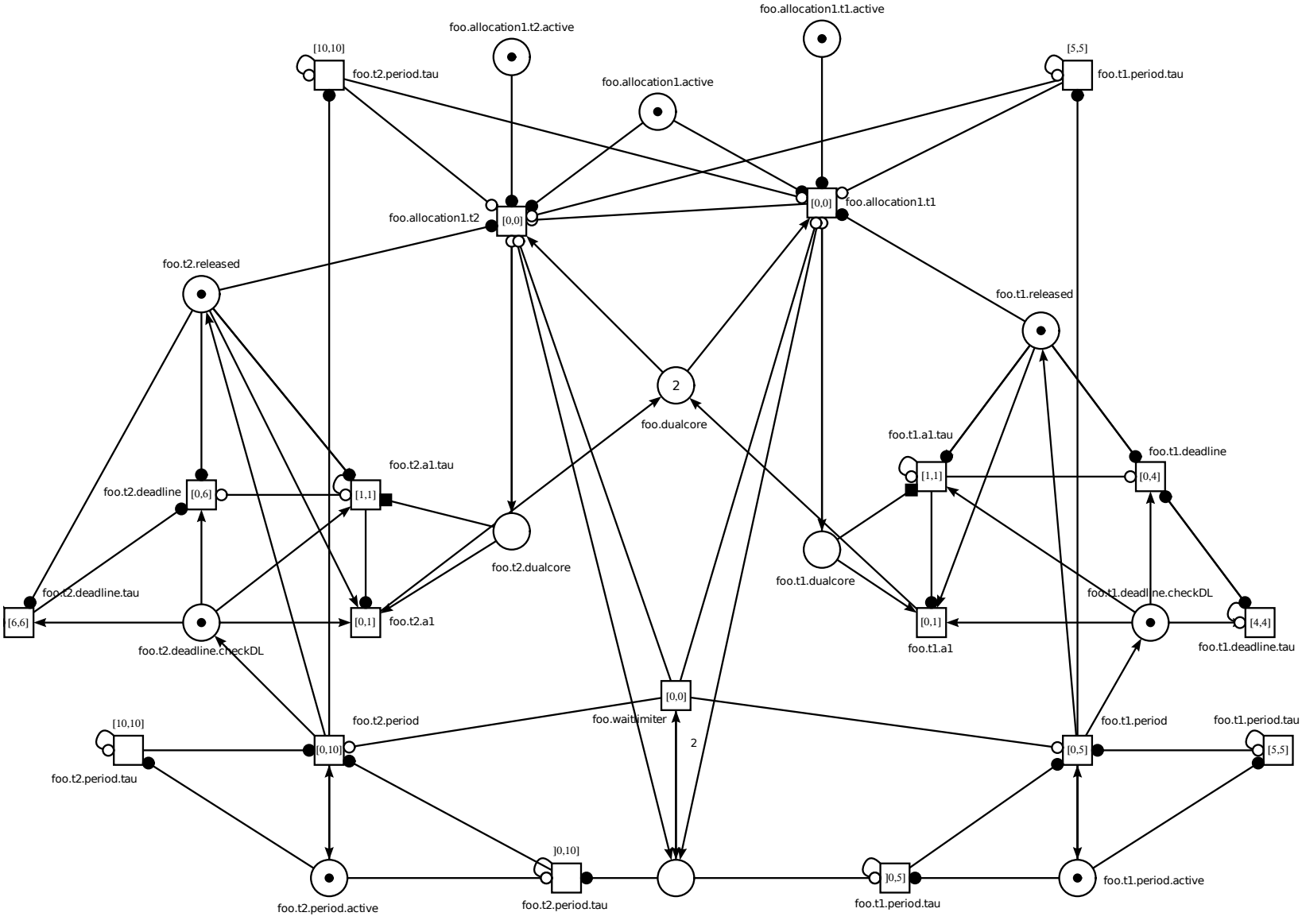
Architecture



Un exemple de traduction - modèle POLA

```
system foo is
 res dualcore is preemptable pool of 2
 task t1 is
  action a1 in [1,1] with allocation1
  period [5,5]
  deadline 4
  policy RM
 end
 not preemptable task t2 is
  action a1 in [1,1] with allocation1
  period [10,10]
  deadline 6
  policy RM
 end
 policy RM is min P
 allocation allocation1 is
  resources dualcore
  tasks t1,t2
end
```

Un exemple de traduction - modèle ipTPN



Résultats d'analyse

```
# net noname, 15 places, 10 transitions                              #
# bounded, not live, possibly reversible                             #
# abstraction        count      props      psets       dead       live #
#      states           17         15         13          0         17 #
# transitions           22         10         10          2          8 #
**************************
*Deadline Misses Checking*
**************************
Loading graph behavior ... DONE
Is there any deadline miss in the system ?  NO
Does task foo.t1 miss its deadline ?     NO
Does task foo.t2 miss its deadline ?     NO
********************
*Liveness Checking*
********************
Loading graph behavior ... DONE
For all possible executions, will the system execute action foo.t1.a1 ? YES
For all possible executions, will the system execute action foo.t2.a1 ? YES
Loading graph behavior ... DONE
Is live foo.t1.a1 ? TRUE
Is live foo.t2.a1 ? TRUE
```

L'ordonnançabilité n'est qu'une propriété parmi tant d'autres

Plan

Conclusions

ipTPN

- nouvelles possibilités de contraintes temporelles
- modèle composable (modulo une transformation simple)
- équivalents aux automates temporisés

Pola

- sémantique sous forme d'(ip)(sw)TPN
- multi-ressources
- intervalles temporels
- politiques statiques
- attribution des ressources
- prototype opérationnel
- études de cas : groupes OSEK et partitions ARINC

Perpectives et travaux futurs

ipTPN

- vérification de propriétés (temporelles) à l'aide d'observateurs
- politiques dynamiques
- exploiter la compositionnalité pour la vérification

Pola

- dépendances entre actions et tâches
- *behavior*: (ip)TPN $\Rightarrow$ FIACRE
- inversions de priorités: protocoles à héritage de priorité