Cours Algorithmique des systèmes parallèles et distribués
Exercices
Série 1 Modélisation, programmation et vérification en TLA$^+$
par Dominique Méry
20 janvier 2026

Modélisation et vérification avec TLA$^+$

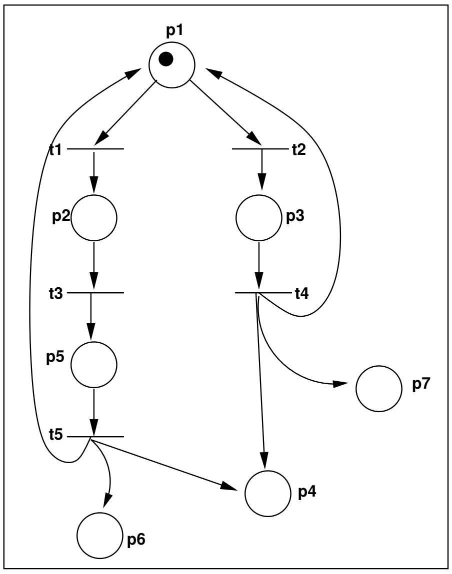
RAPPELS

Un réseau de Petri est un uple R=(S,T,F,K,M,W) tel que

- S est l'ensemble (fini) des places.
- T est l'ensemble (fini) des transitions.
- $S \cap T = \varnothing$
- F est la relation du flôt d'exécution : $F \subseteq S \times T \cup T \times S$
- K représente la capacité de chaque place : $K \in S \rightarrow Nat$.
- M représente le initial marquage chaque place :
  $M \in S \rightarrow Nat$ et vérifie la condition $\forall s \in S : M(s) \leq K(s)$.
- W représente le poids de chaque arc : $W \in F \rightarrow Nat$
- un marquage M pour R est une fonction de S dans Nat :
  $M \in S \rightarrow Nat$ et respectant la condition $\forall s \in S : M(s) \leq K(s)$.
- une transition t de T est activable à partir de M un marquage de R si
  1. $\forall s \in \{ s' \in S \mid (s',t) \in F \} : M(s) \geq W(s,t)$.
  2. $\forall s \in \{ s' \in S \mid (t,s') \in F \} : M(s) \leq K(s) - W(s,t)$.
- Pour chaque transition t de T, Pre(t) est l'ensemble des places conduisant à t et Post(t) est l'ensemble des places pointées par un lien depuis t :
  $Pre(t) = \{ s' \in S : (s',t) \in F \}$ et $Post(t) = \{ s' \in S : (t,s') \in F \}$
- Soit une transition t de T activable à partir de M un marquage de R :
  1. $\forall s \in \{ s' \in S \mid (s',t) \in F \} : M(s) \geq W(s,t)$.
  2. $\forall s \in \{ s' \in S \mid (t,s') \in F \} : M(s) \leq K(s) - W(s,t)$.
- un nouveau marquage M' est défini à partir de M par : ' $\forall s \in S$,

$$M'(s) = \begin{cases} \text{M(s)}-\text{W(s,t), si s} \in \text{Pre(t) - Post(t)} \\ \text{M(s)}+\text{W(t,s), si s} \in \text{Post(t) - Pre(t)} \\ \text{M(s)}-\text{W(s,t)}+\text{W(t,s), si s} \in \text{Pre(t)}\cap\text{Post(t)} \\ \text{M(s), sinon} \end{cases}$$

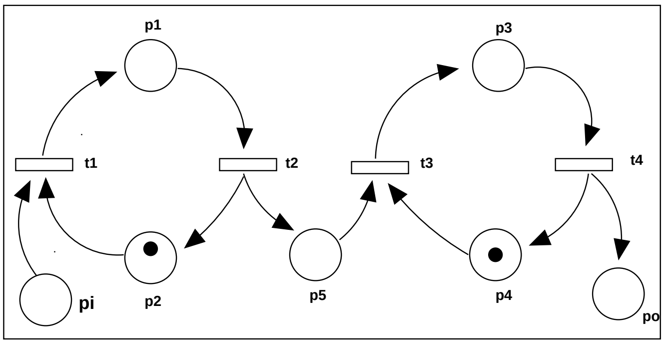
Exercice 1 *(petri13.tla)*
Soit le réseau de Petri suivant :



Question 1.1 *Modéliser ce réseau de Petri avec TLA$^+$.*

Question 1.2 *Etudier ce réseau en proposant et en vérifiant des invariants Ã l'aide des outils.*

Exercice 2 *(petri10.tla)*
On considère le réseau suivant :



Question 2.1 *Traduire ce réseau en un module TLA$^+$.. Pour cela, on donnera la définition des quatre transitions $t1$, $t2$, $t3$,$t4$. On ne tiendra pas compte de la capacité des places : les places ont une capacité d'au plus un jeton, sauf la place pi qui peut contenir N jetons, la place p5 peut contenir au plus B jetons et la place po peut contenir au plus Q.*

Question 2.2 *Donner une relation liant les places po,p1,p3,p5,pi et la valeur N. Justifier la réponse.*

Question 2.3 *Si on suppose que la place po peut contenir au plus Q jetons, donnez une condition sur Q pour que tous les jetons de pi soient consommés un jour. Justifier la réponse.*

Question 2.4 *Expliquer ce que modélise ce réseau de Petri.*

Exercice 3 *(petri14.tla)*
La figure 1 est un réseau de Petri modélisant le système des philosophes qui mangent des spaghetti.

Question 3.1 *Traduire le réseau de Petri sous la forme d'un module TLA.*

Question 3.2 *Est-ce que le réseau peut atteindre un point de deadlock ? Expliquezr votre réponse.*

Question 3.3 *Proposer une propriété TLA pour répondre à la question suivante, en donnant des explications.*
Est-ce que deux philosophes voisins peuvent manger en même temps ?

Exercice 4 *(disapp_td1_ex1.tla)*

Question 4.1 *Modéliser sous forme d'un module TLA$^+$ le réseau de Petri de la figure 2. Donner une instanciation possible des constantes. Préciser les conditions initiales correspondant à un système avec cinq (5) processeurs et deux (2) bus.*

Question 4.2 *On désire analyser le comportement de ce réseau et, pour cela, on souhaite savoir si la place $p5$ contiendra au moins un jeton. Expliquer comment on doit procéder pour obtenir une réponse en un temps fini. Préciser le message donné par le système TLAPS.*

Question 4.3 *Est-ce que le réseau peut atteindre un point de deadlock ?*

Question 4.4 *Enoncez trois propriétés de sûreté de ce réseau établissant une relation entre au moins deux places.*

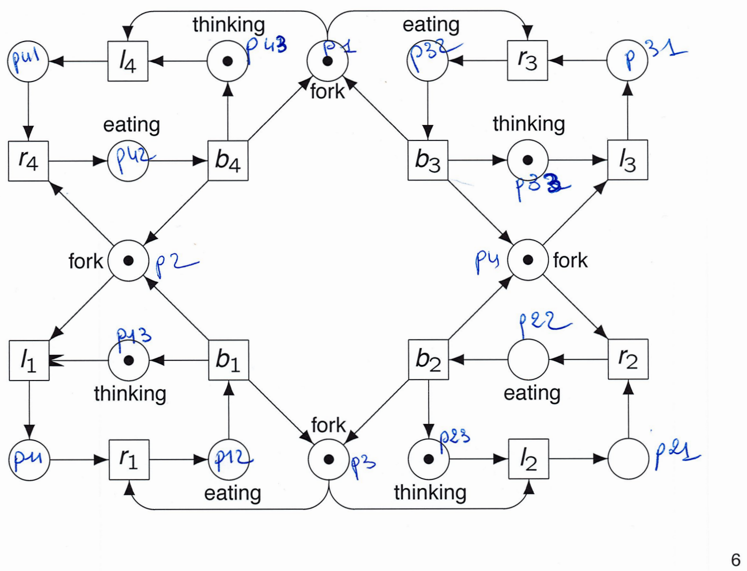


FIGURE 1 – Réseau de Petri

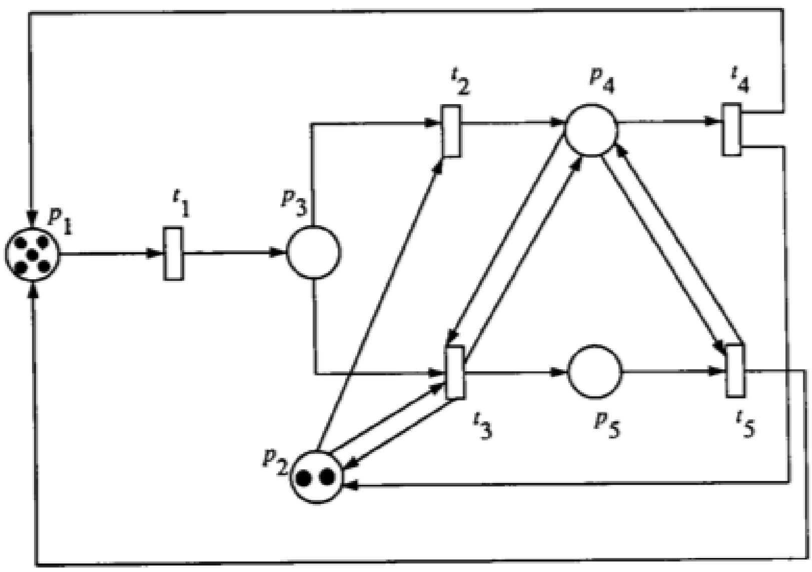


Fig. 14. A Petri-net model of a multiprocessor system, where tokens in p_1 represent active processors, p_2 available buses, p_3, p_4, and p_5 processors waiting for, having access to, queued for common memories, respectively.

FIGURE 2 – Réseau de Petri