

Cours Algorithmique des systèmes parallèles et distribués
Exercices

Série 1 Modélisation, programmation et vérification en TLA⁺
par Dominique Méry
12 février 2026

Modélisation et vérification avec TLA⁺

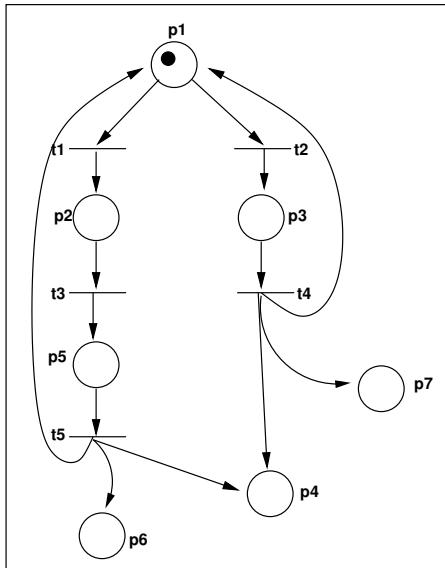
RAPPELS

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.
- $S \cap T = \emptyset$
- F est la relation du flot d'exécution : $F \subseteq S \times T \cup T \times S$
- K représente la capacité de chaque place : $K : S \rightarrow \text{Nat}$.
- M représente le initial marquage chaque place :
 $M : S \rightarrow \text{Nat}$ et vérifie la condition $\forall s \in S : M(s) \leq K(s)$.
- W représente le poids de chaque arc : $W : F \rightarrow \text{Nat}$
- un marquage M pour R est une fonction de S dans Nat :
 $M : S \rightarrow \text{Nat}$ 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 , $\text{Pre}(t)$ est l'ensemble des places conduisant à t et $\text{Post}(t)$ est l'ensemble des places pointées par un lien depuis t :
 $\text{Pre}(t) = \{ s' \in S : (s',t) \in F \}$ et $\text{Post}(t) = \{ s' \in S : (t,s') \in F \}$
- 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 : $\forall s \in S$,
$$M'(s) = \begin{cases} M(s) - W(s,T), & \text{SI } s \in \text{PRE}(T) - \text{POST}(T) \\ M(s) + W(T,s), & \text{SI } s \in \text{POST}(T) - \text{PRE}(T) \\ M(s) - W(s,T) + W(T,s), & \text{SI } s \in \text{PRE}(T) \cap \text{POST}(T) \\ M(s), & \text{SINON} \end{cases}$$

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 à l'aide des outils.

```

----- MODULE petri13 -----
EXTENDS Naturals ,TLC
-----
VARIABLES M
-----
CONSTANT Places ,N
-----
TypeOK == M \in [Places -> Nat]
-----
/* condition de t1
t1 ==
  /\ M["p1"] \geq 1
  /\ M' = [[M EXCEPT! ["p1"] = @-1] EXCEPT! ["p2"] = @+1]

t2 ==
  /\ M["p1"] \geq 1
  /\ M' = [[M EXCEPT! ["p1"] = @-1] EXCEPT! ["p3"] = @+1]

t3 ==
  /\ M["p2"] \geq 1
  /\ M' = [[M EXCEPT! ["p2"] = @-1] EXCEPT! ["p5"] = @+1]

```

```

t4 ==
    /\ M["p3"] \geq 1 /\ M["p4"] < N
    /\ M' = [[[ M EXCEPT! ["p1"] = @+1] EXCEPT! ["p7"] = @+1] EXCEPT! ["p4"] =
    /\ M["p5"] \geq 1 /\ M["p4"] < N
    /\ M' = [[[M EXCEPT! ["p1"] = @+1] EXCEPT! ["p6"] = @+1]
EXCEPT! ["p4"] = @+1] EXCEPT! ["p5"] = @-1]

-----
(* defining the system *)
Init == M = [p \in Places |> IF p \in {"p1"} THEN 1 ELSE 0]
Next == t1 \vee t2 \vee t3 \vee t4 \vee t5
-----

(* analysing properties of the system *)
P1 == M["p4"] = M["p6"]+M["p7"]
P2 == M["p1"] + M["p2"]+M["p3"]+M["p5"] =1
=====

/* SPECIFICATION
/* Uncomment the previous line and provide the specification name if it's declared
/* in the specification file. Comment INIT / NEXT parameters if you use SPECIFICATION

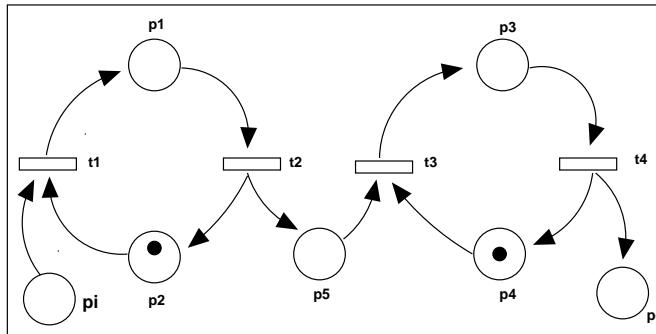
CONSTANTS
greeting = "Hello"
Places = {"p1", "p2", "p3", "p4", "p5", "p6", "p7"}
intmax = 100000

INIT Init
NEXT Next

/* PROPERTY
/* Uncomment the previous line and add property names

INVARIANT
/* Uncomment the previous line and add invariant names
P1
P2

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 t_1, t_2, t_3, t_4 . On ne tiendra pas compte de la capacité des places : les places ont une capacité d'au plus un jeton, sauf la place p_i qui peut contenir N jetons, la place p_5 peut contenir au plus B jetons et la place p_0 peut contenir au plus Q .

Question 2.2 Donner une relation liant les places p_0, p_1, p_3, p_5, p_i et la valeur N . Justifier la réponse.

Question 2.3 Si on suppose que la place p_0 peut contenir au plus Q jetons, donnez une condition sur Q pour que tous les jetons de p_i soient consommés un jour. Justifier la réponse.

Question 2.4 Expliquer ce que modélise ce réseau de Petri.

← Solution de l'exercice 2 _____

```
----- MODULE petri13 -----
EXTENDS Naturals ,TLC
-----
VARIABLES M
-----
CONSTANT Places ,N
-----
TypeOK == M \in [Places -> Nat]
-----
/* condition de t1
t1 ==
  /\ M["p1"] \geq 1
  /\ M' = [[M EXCEPT! ["p1"] = @-1] EXCEPT! ["p2"] = @+1]

t2 ==
  /\ M["p1"] \geq 1
  /\ M' = [[M EXCEPT! ["p1"] = @-1] EXCEPT! ["p3"] = @+1]
```

```

t3 ==
    /\ M["p2"] \geq 1
    /\ M' = [[M EXCEPT! ["p2"] = @-1] EXCEPT! ["p5"] = @+1]

t4 ==
    /\ M["p3"] \geq 1 /\ M["p4"] < N
    /\ M' = [[[ M EXCEPT! ["p1"] = @+1] EXCEPT! ["p7"] = @+1] EXCEPT! ["p4"] = @+1]

t5 ==
    /\ M["p5"] \geq 1 /\ M["p4"] < N
    /\ M' = [[[ M EXCEPT! ["p1"] = @+1] EXCEPT! ["p6"] = @+1]
EXCEPT! ["p4"] = @+1] EXCEPT! ["p5"] = @-1]

-----
(* defining the system *)
Init == M = [p \in Places |-> IF p \in {"p1"} THEN 1 ELSE 0]
Next == t1 \wedge t2 \wedge t3 \wedge t4 \wedge t5
-----
(* analysionsg properties of nthe system *)
P1 == M["p4"] = M["p6"]+M["p7"]
P2 == M["p1"] + M["p2"]+M["p3"]+M["p5"] =1
=====

/* SPECIFICATION
/* Uncomment the previous line and provide the specification name if it's declared
/* in the specification file. Comment INIT / NEXT parameters if you use SPECIFICATION

CONSTANTS
    greeting = "Hello"
    Places = {"p1", "p2", "p3", "p4", "p5", "p6", "p7"}
    intmax = 100000

INIT Init
NEXT Next

/* PROPERTY
/* Uncomment the previous line and add property names

INVARIANT
/* Uncomment the previous line and add invariant names
P1
P2

```

Fin 2

Exercice 3 (*petri14.tla*)

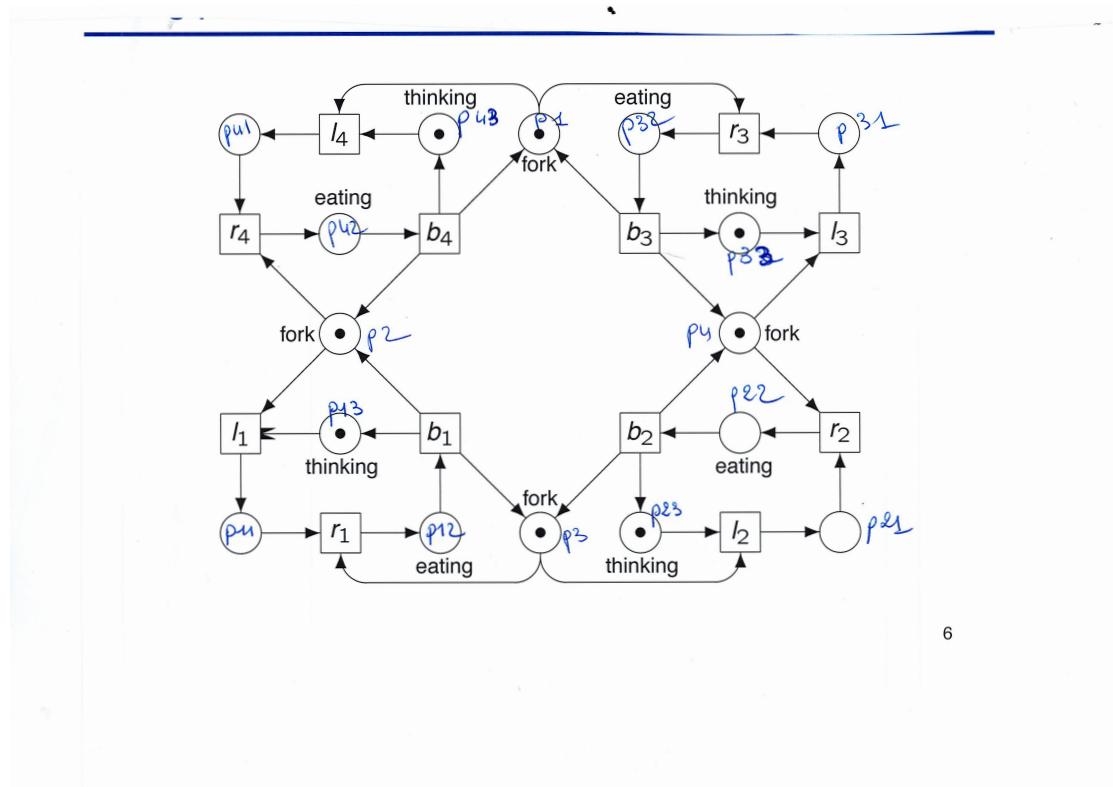
La figure 1 est un réseau de Petri modélisant le système des philosophes qui mangent des spaghetti.

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 ? Expliquez 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 ?



6

FIGURE 1 – Réseau de Petri

----- MODULE petri14 -----
EXTENDS Naturals , TLC

```

-----
CONSTANTS Places (* d\'esigne l'ensemble des places du r\'eseau de Petri *)
-----
VARIABLES M (* la variable d'\\'etat indiquant o\`u se trouvent les jetons *)
-----
ASSUME
    Places \subseteqq {"p1", "p11", "p12", "p13", "p2", "p21", "p22", "p23",
    "p3", "p31", "p32", "p33", "p4", "p41", "p42", "p43"}
-----
l1 ==
    M["p2"] >= 1 /\ M = [ [[M EXCEPT!["p2"] = M["p2"] - 1] EXCEPT!["p11"] = M["p11"] + 1]
    /\ l1
-----
Init == M = [p \in Places |-> IF p \in {"p1", "p2", "p3", "p4", "p13", "p23", "p33"
Next ==
    \/\ l1
=====

'
\* SPECIFICATION
\* Uncomment the previous line and provide the specification name if it's declared
\* in the specification file. Comment INIT / NEXT parameters if you use SPECIFICATION
CONSTANTS
    greeting = "Hello"
    Places = {"p1", "p11", "p12", "p13", "p2", "p21", "p22", "p23", "p3", "p31", "p32"
INIT Init
NEXT Next

\* PROPERTY
\* Uncomment the previous line and add property names

\* INVARIANT
\* Uncomment the previous line and add invariant names

```

Exercice 4 (disapp_td1_ex1.tla)

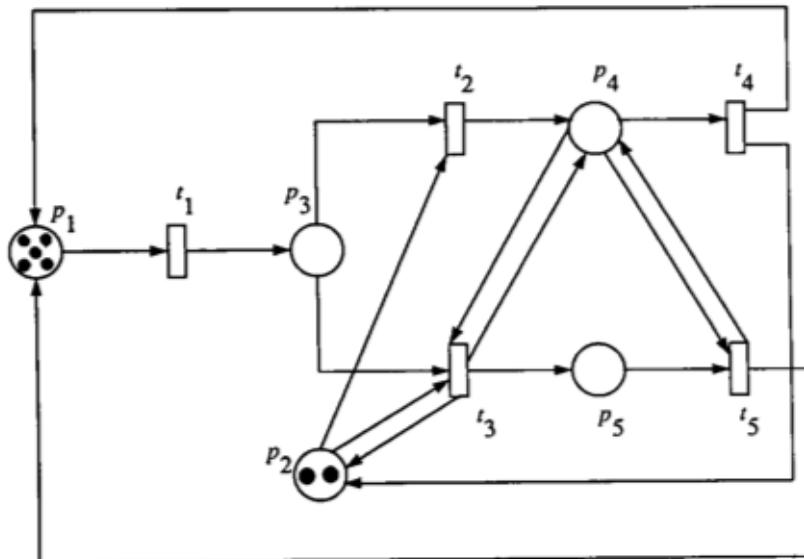


Fig. 14. A Petri-net model of a multiprocessor system, where tokens in p_1 represent active processors, p_2 available buses, p_3 , p_4 , and p_5 processors waiting for, having access to, queued for common memories, respectively.

FIGURE 2 – Réseau de Petri

Question 4.1 Modéliser sous forme d'un module TLA⁺ le réseau de Petri de la figure 2. Donner une instantiation 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 p_5 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.

----- MODULE petri15 -----
(* Petri-net for the multiprocessor system *)

```

EXTENDS Naturals ,TLC
CONSTANTS Places ,P,B
VARIABLES M
-----
(* Assumptions over Places namely Stellen *)
K == [p \in FPlaces |-> IF p \in {"p3"} THEN 1 ELSE 0 ]

TypeOK == M \in Places --> Nat
-----
t1 ==
  /\ M["p1"] \geq 1 /\ M["p3"]+1 \leq K["p3"]
  /\ M=[[M EXCEPT!["p3"]=M["p3"]+1] EXCEPT!["p1"]=M["p1"]-1]

t2 ==
  /\ M["p2"] \geq 1 /\ M["p3"] \geq 1
  /\ M=[[M EXCEPT!["p3"]=M["p3"]-1]
         EXCEPT!["p2"]=M["p2"]-1]
         EXCEPT!["p4"]=M["p4"]+1]

t3 ==
  /\ M["p2"] \geq 1 /\ M["p3"] \geq 1 /\ M["p4"] \geq 1
  /\ M=[[M EXCEPT!["p3"]=M["p3"]-1]
         EXCEPT!["p5"]=M["p5"]+1]

t4 ==
  /\ M["p4"] \geq 1
  /\ M=[[M EXCEPT!["p4"]=M["p4"]-1]
         EXCEPT!["p2"]=M["p2"]+1]
         EXCEPT!["p1"]=M["p1"]+1]

t5 ==
  /\ M["p4"] \geq 1 /\ M["p5"] \geq 1
  /\ M=[[M EXCEPT!["p5"]=M["p5"]-1]
         EXCEPT!["p1"]=M["p1"]+1]

-----
Init == M = [p \in Places |->
             IF p \in {"p1"} THEN P
             ELSE IF p \in {"p2"}
             THEN B ELSE 0 ]

```

```

Next == t1 \vee t2 \vee t3 \vee t4 \vee t5
Petri == Init /\ [] [Next]_<<M>>

I0 == \A p \in Places : M[p] \geq 0

I1 == 0 \leq M["p3"] /\ M["p3"] \leq 1
I2 == M["p1"] \leq 5 /\ M["p2"] \leq 2
I3 == \A p \in {"p5"} : 0 \leq M[p] /\ M[p] \leq 5
I4 == \A p \in {"p4"} : 0 \leq M[p] /\ M[p] \leq 2

AP4 == M["p4"] = 0
AP5 == M["p5"] = 0
Test == I0 /\ I1 /\ I2 /\ I3 /\ I4
=====
```