

Epreuve TP ASPD

Vous déposerez cette archive via Arche et vous enverrez votre archive nom-prenom-aspd2023.zip; vous attendrez avant de partir un accusé de réception de la part de Dominique Méry avant de quitter la salle d'épreuve.

Exercice 1

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

Question 1.1 Traduire le réseau de Petri sous la forme d'un module TLA, en utilisant le fichier petri2023.tla. En particulier, il faut compléter l'initialisation.

Question 1.2 Est-ce que le réseau peut atteindre un point de deadlock? Expliquez votre réponse.

Question 1.3 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?

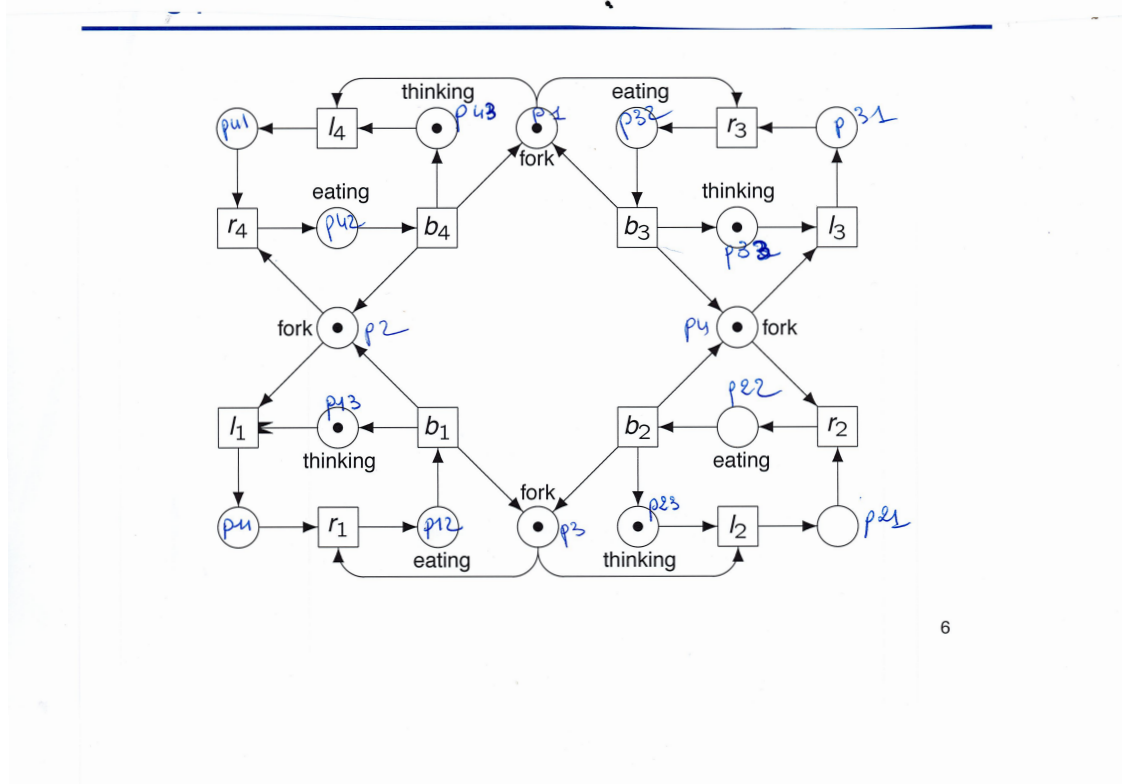


Figure 1: Réseau de Petri

```
----- MODULE examen2023q1 -----
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 \subseteq {"p11", "p12", "p13", ...}
-----
l1 ==
r1 ==
```

```

b1 ==
.....

Init ==  M = [p \in Places |-> IF p \in {"p1","p2","p3","p4"} THEN 1 ELSE IF .... ]
Next == t1 \/ t2      \/ t3      \/ t4      \/ t5

```

=====

Exercice 2 Compléter le module *pluscalaspd1.tla* en proposant une assertion *P1* correcte.

```

----- MODULE pluscalaspd1 -----
EXTENDS Integers, Sequences, TLC, FiniteSets
(*
--wf
--algorithm ex1{
variables x = 0;

process (one = 1)
{
  A:
    x := x - 1;
};

process (two = 2)
{
  C:
    x := x + 1;
  D:
    assert P1;
};

}
end algorithm;

*)

```

=====

Exercice 3 Compléter le module *pluscalaspd2.tla* en proposant une assertion *Q1* correcte.

```

----- MODULE pluscalaspd2 -----
EXTENDS Integers, Sequences, TLC, FiniteSets
(*
--wf
--algorithm ex1{
variables x = 0;

process (one = 1)
{
  A:
    x := x + 1;
  B:
    x := x +1;
};

process (two = 2)
{

```

```

    C:
      x := x - 1;
    D:
      assert \ldots;
  };

}
end algorithm;

*)

```

====

Exercice 4 Compléter le module *pluscalaspd3.tla* en proposant deux assertions *R1* et *R2* correctes.

```

----- MODULE pluscalaspd3 -----
EXTENDS Integers, Sequences, TLC, FiniteSets
(*
--wf
--algorithm ex3{
variables x = 0, y = 2;

process (one = 1)
{
  A:
    x := x + 1;
  B:
    y := y -1;
  C:
    assert Q1;
};

process (two = 2)
{
  D:
    x := x - 1;
  E:
    y:=y+2;
  F:
    x:= x+2;
  G:
    assert Q2;
};

}
end algorithm;

*)

```

====