

Université Mohamed Boudiaf de M'Sila
Faculté de Mathématiques et d'Informatique
Département d'Informatique
Module de Logique et Algèbres de Processus (M2IA)

Examen Final

Le 10/02/2022

Durée : 1h30 minutes

Exercice n° 1 (6 points) :

- a) En utilisant les règles d'inférence dites si les deux expressions A et B suivantes sont équivalentes ou non (2 points)

A = a; (b; stop [] c; stop), B=a; b; stop [] a; c; stop

Donner une explication (2 points)

- b) En utilisant les règles d'inférence donner le résultat de l'expansion de **(a; b; stop) [[b]] (b; c; stop)** (2 points)

Exercice n° 2 (6 points) :

En utilisant les règles d'inférence dessiner l'arbre (STE) du processus LOTOS suivant :

process P [a,b,c]

(a; b; stop) [[b]] (c; b; stop)

endproc

Barème : 4 points pour les règles d'inférence (l'expansion) et 2 points pour le graphe

Exercice n° 3 (8 points) :

Ecrire la spécification LOTOS pour le circuit logique (5 points):

d = or(a, and(b, c))

Dessiner le système à transitions étiquetées (3 points)

Bonne Chance
Pr. M. Bourahla

Corrigé Type

Module : Logique et Algèbres de Processus LAP (M2IA)

Exercice n° 1 :

$A = a; (b; \text{stop} \parallel c; \text{stop})$

$A \xrightarrow{a} b; \text{stop} \parallel c; \text{stop} \xrightarrow{b} \text{stop}$
 $\qquad \qquad \qquad b; \text{stop} \parallel c; \text{stop} \xrightarrow{c} \text{stop}$ 1 point

$B = a; b; \text{stop} \parallel a; c; \text{stop}$

$B \xrightarrow{a} b; \text{stop} \parallel b; \text{stop} \xrightarrow{b} \text{stop}$
 $B \xrightarrow{a} c; \text{stop} \parallel c; \text{stop} \xrightarrow{c} \text{stop}$ 1 point

Le choix de l'action a dans le comportement A transforme ce dernier en $(b; \text{stop} \parallel c; \text{stop})$. Tandis que le choix de a dans le comportement B le transforme ou bien en $b; \text{stop}$ ou bien en $c; \text{stop}$, selon l'a qui est choisi. 2 points

$(a; b; \text{stop}) \parallel [b] (b; c; \text{stop}) \xrightarrow{a} (b; \text{stop}) \parallel [b] (b; c; \text{stop}) \xrightarrow{b} (\text{stop}) \parallel (c; \text{stop}) \xrightarrow{c} (\text{stop}) \parallel (\text{stop}) = a; b; c; \text{stop}$ 2 points

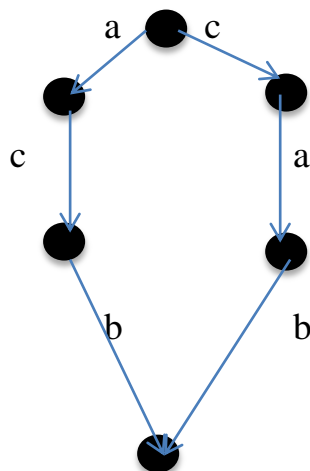
Exercice n° 2:

$(a; b; \text{stop}) \parallel [b] (c; b; \text{stop}) \xrightarrow{a} (b; \text{stop}) \parallel [b] (c; b; \text{stop}) \xrightarrow{c} (b; \text{stop}) \parallel [b] (b; \text{stop}) \xrightarrow{b} (\text{stop}) \parallel (\text{stop}) \xrightarrow{b} \text{stop} = (a; c; b; \text{stop})$

$(a; b; \text{stop}) \parallel [b] (c; b; \text{stop}) \xrightarrow{c} (a; b; \text{stop}) \parallel [b] (b; \text{stop}) \xrightarrow{a} (b; \text{stop}) \parallel [b] (b; \text{stop}) \xrightarrow{b} (\text{stop}) \parallel (\text{stop}) \xrightarrow{b} \text{stop} = (c; a; b; \text{stop})$

Donc

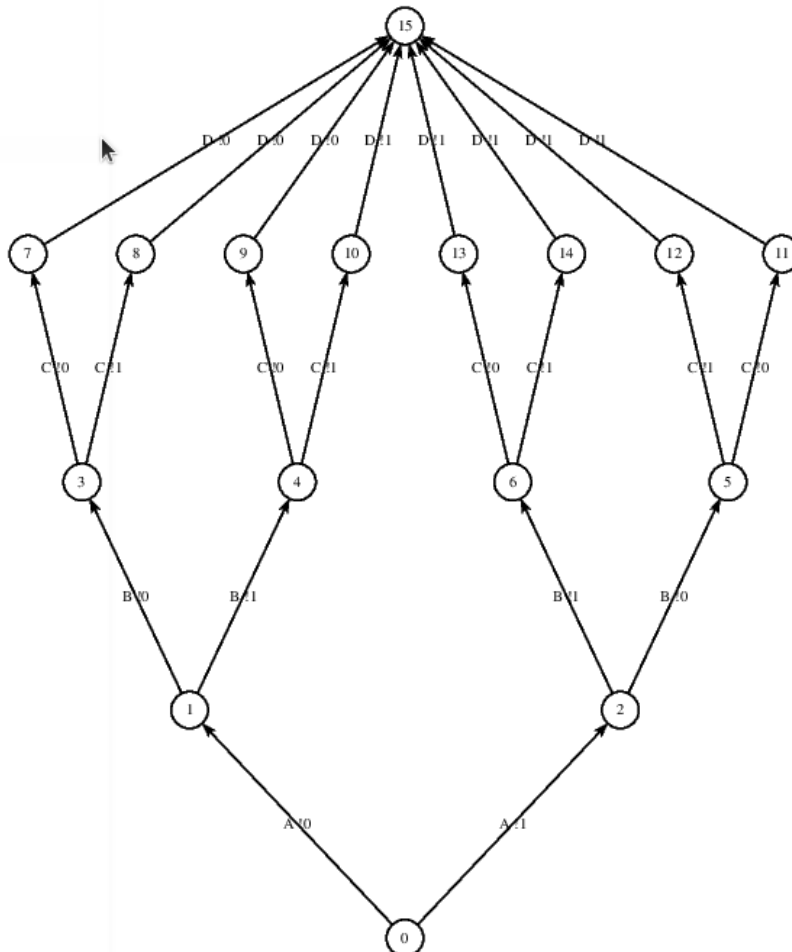
$(a; b; \text{stop}) \parallel [b] (c; b; \text{stop}) = (a; c; b; \text{stop}) \parallel (c; a; b; \text{stop})$ 4 points



2 points

Exercice n° 3:

```
specification circuit_logique_and [a, b, c, d] : noexit
type BIT is
  sorts BIT
  opns 0 (*! constructor *),
       1 (*! constructor *) : -> BIT
  and : BIT, BIT -> BIT
  or  : BIT, BIT -> BIT
eqns
  ofsort BIT
    and (0, 0) = 0;
    and (0, 1) = 0;
    and (1, 0) = 0;
    and (1, 1) = 1;
    or  (0, 0) = 0;
    or  (0, 1) = 1;
    or  (1, 0) = 1;
    or  (1, 1) = 1;
endtype
behaviour
  or_and[a, b, c, d]
where
  process or_and[a, b, c, d] : noexit :=
    a ?aa:Bit; b ?bb:Bit; c ?cc:Bit; d !or(aa, and(bb, cc)); stop
  endproc
endspec    5 points
```



3 points