Master 2 : Intelligence Artificielle

Logique et Algèbres de Processus

Pr. Mustapha Bourahla

Département d'Informatique Université de M'Sila

Exercices corrigés

Exercice n° 1: Lois algébriques pour []

- Les lois algébriques de LOTOS sont une conséquence des règles d'inférence
- Persuadez-vous que ceci est vrai pour les trois lois que nous avons mentionné pour l'opérateur []
  - A[]B = B[]A
  - A[]stop = stop[]A = A
  - A[](B[]C) = (A[]B)[]C

Solutions: Lois algébriques pour []

- A[]B = B[]A

$$\frac{B1 \xrightarrow{a1} B1'}{B1 \,[]\, B2 \xrightarrow{a1} B1'}$$

$$\frac{B2 \xrightarrow{a2} B2'}{B1 \,[]\, B2 \xrightarrow{a2} B2'}$$

$$\frac{A \xrightarrow{a} A'}{A \,[]\, B \xrightarrow{a} A'}$$

$$\frac{A \xrightarrow{a} A'}{B \,[]\, A \xrightarrow{a} A'}$$

Donc A[]B = B[]A

Solutions: Lois algébriques pour []

A[]stop = stop[]A = A

Évidemment, si A ou B est un stop (c'est-à-dire qu'il n'a pas d'action), l'action devra venir de l'autre comportement :

A [] stop = A

stop [] A = A

donc une seule des deux dérivations ci-dessus est possible.

Solutions: Lois algébriques pour []

$A[](B[]C) = (A[]B)[]C$

$$\frac{B1 \xrightarrow{a1} B1'}{B1 \,[]\, B2 \xrightarrow{a1} B1'}$$

$$\frac{B2 \xrightarrow{a2} B2'}{B1 \,[]\, B2 \xrightarrow{a2} B2'}$$

$$\frac{A \xrightarrow{a} A'}{A[](B[]C) \xrightarrow{a} A'}$$

$$\frac{A \xrightarrow{a} A'}{(A[]B)[]C \xrightarrow{a} A'}$$

Donc $A[](B[]C) = (A[]B)[]C$

Exercice n° 2: Lois algébriques pour ||

- Les lois algébriques de LOTOS sont une conséquence des règles d'inférence
- Persuadez-vous que ceci est vrai pour les trois lois que nous avons mentionné pour l'opérateur ||
  - A||B = B||A
  - A||stop = stop||A = stop
  - A||(B||C) = (A||B)||C

Enfin, si vous faites le travail vraiment bien, vous verrez que la deuxième loi n'est pas vraie dans un cas particulier…

Solutions: Lois algébriques pour II

Pour A||B = B||A

$$\frac{A \xrightarrow{a} A' \qquad B \xrightarrow{a} B'}{A \,||\, B \xrightarrow{a} A' ||\, B'} \qquad \frac{B \xrightarrow{a} B' \qquad A \xrightarrow{a} A'}{B \,||\, A \xrightarrow{a} B' ||\, A'}$$

Donc A||B = B||A car nous avons les mêmes conditions

- Pour A||stop = stop||A = stop, stop n'a pas d'action (n'est pas vraie dans les cas des actions internes)
- Pour A||(B||C) = (A||B)||C

$$\frac{A - a > A' \quad B - a > B' \quad C - a > C'}{A \,||\, B \,||C \xrightarrow{a} A' ||\, B' \,||\, C'}$$

Exercice n° 3:

En utilisant les règles d’inference dessiner les arbres (STE) de

process one [a,b,c]

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

endproc

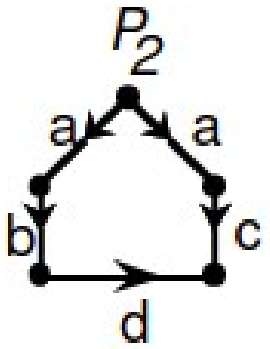
process two [a,b,c]

a; b; stop [] a; c; stop

Endproc

P1 := a; (b; d; stop [] c; stop)

P2 := a; b; d; stop [] a; c; stop



Solutions:

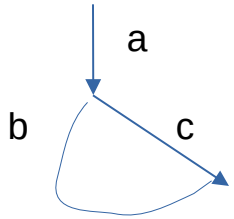
En utilisant les règles d'inference dessiner les arbres (STE) de

process one [a,b,c] a; (b; stop [] c; stop) endproc

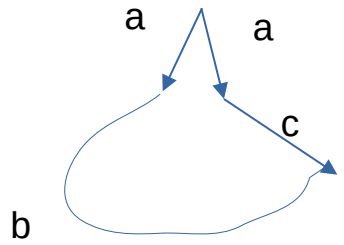
a; ((b; stop [] c; stop)) --- a > (b; stop [] c; stop)

(b; stop [] c; stop) --- b > stop

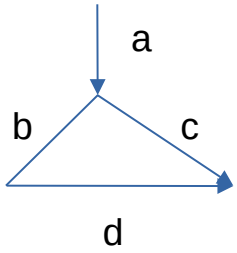
(b; stop [] c; stop) --- c > stop



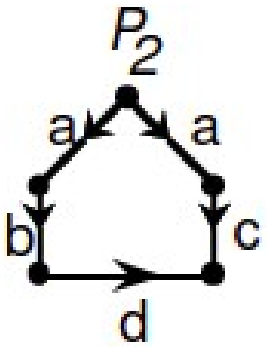
process two [a,b,c] a; b; stop [] a; c; stop endproc



P1 := a; (b; d; stop [] c; stop)



P2 := a; b; d; stop [] a; c; stop



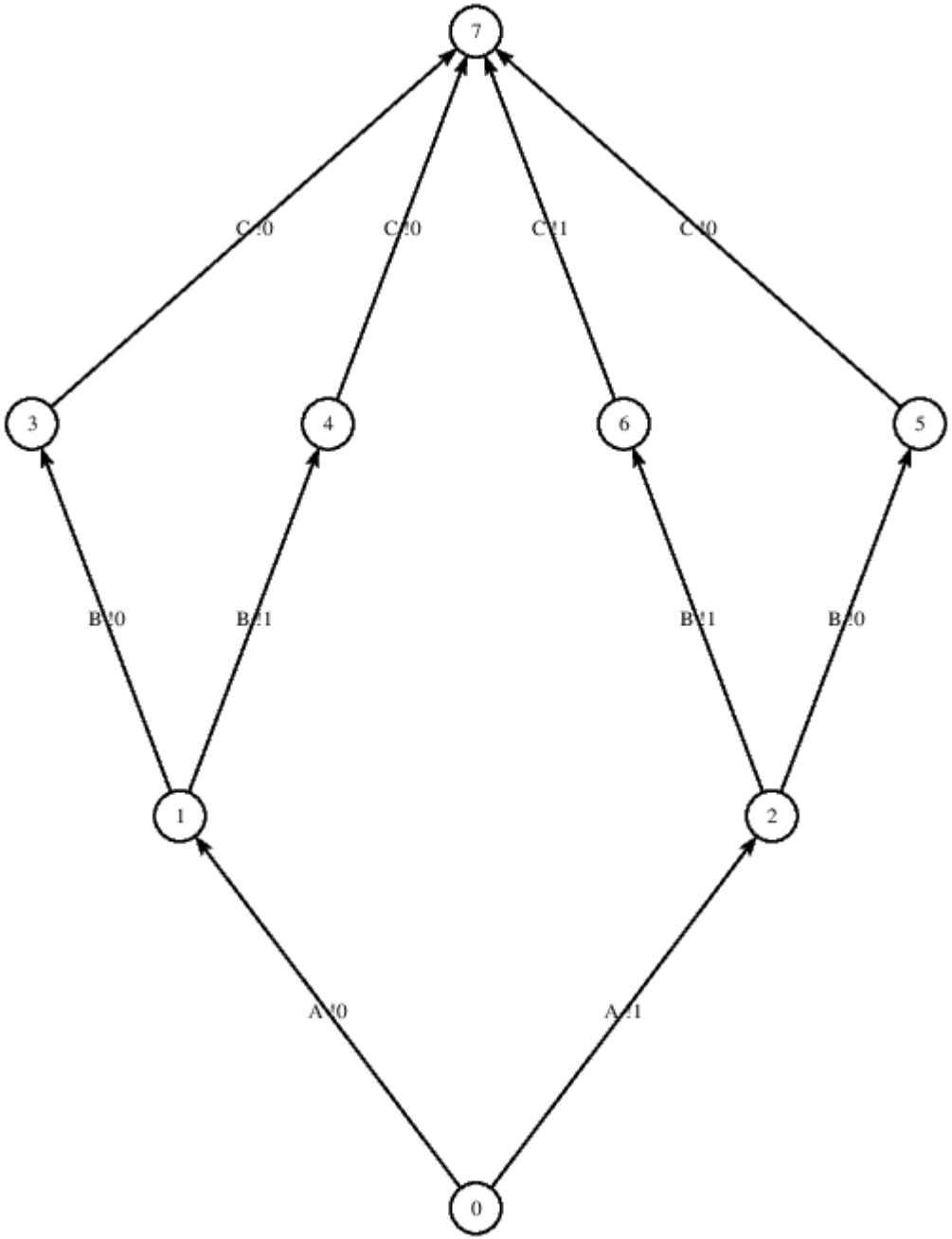
Exercice n° 4:

Ecrire des spécifications lotos pour les circuits logiques: and, or et xor.

Dessiner leurs systèmes à transitions étiquettées

```
Solutions: and
specification circuit_logique [a,b,c] : noexit
type BIT is
   sorts BIT
   opns 0 (*! constructor *),
        1 (*! constructor *) : -> BIT
        and : BIT, BIT -> BIT
   eqns
      ofsort BIT
         and (0, 0) = 0;
         and (0, 1) = 0;
         and (1, 0) = 0;
         and (1, 1) = 1;
endtype

behaviour
   gate_and[a, b, c]
where
   process gate_and[a, b, c] : noexit :=
     a ?aa:Bit; b ?bb:Bit; c !and(aa, bb); stop
   endproc
endspec
```


7
C:0
C:0
C:1
C:0
3
4
6
5
B:0
B:1
B:1
B:0
1
2
A:0
A:1
0