Universite de M'sila
Département math et informatique
Master 2 IA
logique et algebre de processus

Gherabi Amira

November 20, 2021

exercice n01: la demonstration des trois lois pour l'operateur []

1. A[]B = B[]A → A   Commutativité

∗ on a A[]B $\xrightarrow{a} A\ et\ B[]A \xrightarrow{a} A\, alors\, A[]B = B[]A$

2. A[]stop = stop[]A = A →   A Zéro absorption

∗ on a A[]stop $\xrightarrow{a} A\ et\ stop[]A \xrightarrow{a} A\, alors\, stop[]A = A[]stop = A$

3. A[](B[]C) = (A[]B)[]C →   Associativité
   on a B[]C $\xrightarrow{a} B\,(1)$
   $on\, a\, A[]B \xrightarrow{a} B\,(2)$
   $depuis\,(1)\,et\,(2)\,on\,trouve\,que:\ A[](B[]C) \xrightarrow{a} B \quad (A)$
   $et(A[]B)[]C \xrightarrow{a} B \quad (B)$
   $depuis\,(A)\,et\,(B)\,on\,trouve\,que:\ A[](B[]C) = (A[]B)[]C$

exercice n02: la demonstration des trois lois pour l'operateur ——

1. A$\|B = B\|A$→ A   Commutativité

∗ on a A$\|B \xrightarrow{a} A\ et\ B\|A \xrightarrow{a} A\, alors\, A\|B = B\|A$

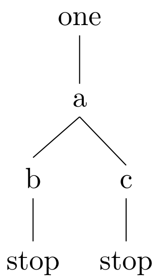
2. A$\|stop = stop\|A = A$→   A Zéro absorption

∗ on a A$\|stop \xrightarrow{a} stop\ et\ stop\|A \xrightarrow{a} stop\, alors\, stop\|A = A||stop = stop$

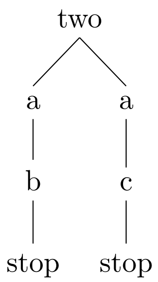
3. A$\|(B\|C) = (A\|B)\|C$→   Associativité

exercice03

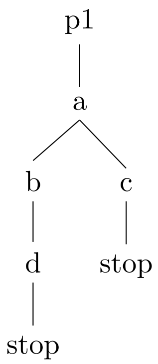
process one [a,b,c] a; (b; stop [] c; stop) endproc



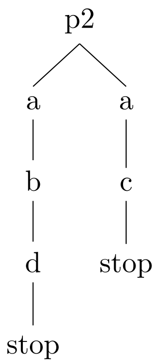
process two [a,b,c] a; b; stop [] a; c; stop Endproc



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



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



exercice04

Les spécifications lotos pour les circuits logiques: or, and et xor

la spécification lotos pour le circuit logique OR

spec OR [a,b,c] :noexit
type BIT is
sorts BIT
opns 0 (*! constructor *),
1 (*! constructor *) : $\rightarrow BIT$
$or : BIT, BIT \rightarrow BIT$
$eqns$
$ofsortBIT$
$or(0,0) = 0;$
$or(0,1) = 1;$
$or(0,1) = 1;$
$or(1,1) = 1;$
$endtype$
$behaviour$
$gateOR[a,b,c]$
$where$
$process\, gate - OR[a,b] : noexit :=$
$a\,?aa : Bit;\ b\,?bb : Bit;\ c\,!or(aa,bb);\ stop$
$endproc$
$endspec$

la spécification lotos pour le circuit logique AND

spec AND [a,b,c] :noexit
type BIT is
sorts BIT
opns 0 (*! constructor *),
1 (*! constructor *) : $\rightarrow BIT$
$and : BIT, BIT \rightarrow BIT$
$eqns$
$of sortBIT$
$and(0,0) = 0;$
$and(0,1) = 0;$
$and(0,1) = 0;$
$and(1,1) = 1;$
$endtype$
$behaviour$
$gate_and[a,b,c]$
$where$
$processgate - AND[a,b] : noexit :=$
$a?aa : Bit; andb?bb : Bit; andc!and(aa,bb); andstop$
$endproc$
$endspec$

la spécification lotos pour le circuit logique XOR

spec XOR [a,b,c] :noexit
type BIT is
sorts BIT
opns 0 (*! constructor *),
1 (*! constructor *) : $\rightarrow BIT$
$xor : BIT, BIT \rightarrow BIT$
$eqns$
$of sortBIT$
$xor(0,0) = 0;$
$xor(0,1) = 1;$
$xor(0,1) = 1;$
$xor(1,1) = 0;$
$endtype$
$behaviour$
$gate - XOR[a,b,c]$
$where$
$processgate_XOr[a,b] : noexit :=$
$a?aa : Bit; andb?bb : Bit; andc!xor(aa,bb); stop$

endproc
endspec