Cours - Systèmes de Transition

Louis Thevenet

Table des matières

1. Mise en pratique : La factorielle

```
-------- MODULE Fact0 -------

EXTENDS Naturals
CONSTANT N
VARIABLE res

Init == res = Fact[N]
Next == UNCHANGED res (*ou FALSE*)
Spec == Init \land [Next]_res
==============
```

Liste 1. – 0 transition

```
-------- MODULE Fact1 -------

EXTENDS Naturals
CONSTANT N
ASSUME N \in Nat
VARIABLES res, i

Init ==
    /\ res = 1
    /\ i = 1

Mult ==
    /\ i <= N
    /\ res' = res * i
    /\ i' = i + 1

Next == Mult

Spec == Init \land [Next]_{res,i}
==============
```

Liste 2. – Avec transitions

```
-------- MODULE Fact1 -------

EXTENDS Naturals
CONSTANT N
ASSUME N \in Nat
VARIABLES res, factors

Init ==
    /\ res = 1
    /\ factors = 1..N

Mult(i) ==
    /\ res' = res * i
    /\ factors' = factors \ {i}


Next == \E i \in factors : Mult (i)

Spec == Init \land [Next]_{res,factors}
==============
```

Liste 3. – Sans ordre particulier

```
-------- MODULE Fact1 -------

EXTENDS Naturals
CONSTANT N
ASSUME N \in Nat
VARIABLES res, factors

Init ==
    /\ res = 1
    /\ factors = 1..N

Mult(I) ==
    /\ res' = (*on multiplie les éléments de I à res*)
    /\ factors = 1..N

Next == \E I \in SUBSET factors : Mult (i)
Spec == Init \land [Next]_{res,factors}
==============
```

Liste 4. – Sans ordre particulier

2. Homme-Loup-Mouton-Chou

On doit les faire passer d'une rive à l'autre d'une rivière.

- Il faut un homme pour ramer
- Sans la surveillance de l'homme
  - le mouton mange le chou
  - le loup mange le mouton

```
-------- MODULE hlmc -------

    VARIABLES h, m, c, l
    RIVES == {"G", "D"}

    Inv(r) ==
        IF r = "G"
            THEN "D"
            ELSE "G"

    TypeInvariant == {h, l, m,c} \subseteq RIVES

    Init ==
        /\ h = "G"
        /\ l = "G"
        /\ m = "G"
        /\ c = "G"
        (*/\ PasMiam*)

    PasMiam ==
        /\ (l = m => h = m)
        /\ (c = m => h = m)

    MoveH ==
        /\ h' = Inv(h)
        /\ UNCHANGED <<l, m, c>>
        /\ PasMiam'

    MoveHL ==
        /\ h' = Inv(h)
        /\ l' = Inv(l)
        /\ h = l
        /\ UNCHANGED << m, c >>
        /\ PasMiam'

    MoveHM ==
        /\ h' = Inv(h)
        /\ m' = Inv(m)
        /\ h = m
        /\ UNCHANGED << l, c >>
        /\ PasMiam'

    MoveHC ==
        /\ h' = Inv(h)
        /\ c' = Inv(c)
        /\ h = c
        /\ UNCHANGED << l, m >>
        /\ PasMiam'

    Next ==
        \/ MoveH
        \/ MoveHL
        \/ MoveHM
        \/ MoveHC

    Spec ==
        /\ Init
        /\ [Next]_<<h,l,m,c>>

    But == [](~ {h,l,m,c} = {"D"})
=================================================
```

Liste 5. – Sans ordre particulier