

Quelques commentaires et exercices sur LUSTRE

1

Charles André - UNSA

Flots d'entrée

- Le compilateur LUSTRE v4 n'admet pas les nœuds sans flot d'entrée (comme par exemple le diviseur de l'horloge de base)

- Solution:

```
node GenClock (dummy:bool) returns (c:bool);
let
  c = true -> not pre c;
tel
```

dummy non utilisé dans les équations.

2

Charles André - UNSA

Assert vs. Masquage

- assert** spécifie une condition qui est supposée toujours satisfaite. Lors des analyses, les évolutions, qui ne satisfont pas les assertions, ne sont considérées.
- Conjonction logique de tous les asserts.
- On peut décider de ne pas tester (ou observer) une propriété lorsqu'une condition est satisfaite. Les évolutions ne sont alors pas contraintes.
- Voir la question 2.5 du TD1

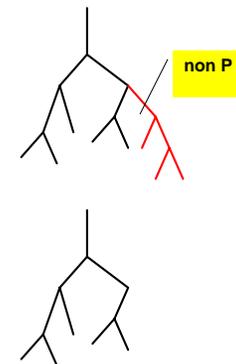
3

Charles André - UNSA

Assert vs. Masquage (2)

- assert P**
- E.g.


```
assert not (S and R);
```

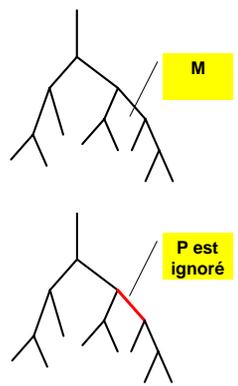


4

Charles André - UNSA

Assert vs. Masquage (2)

- $ok = P \text{ or } M$
- E.g.
 $M = true \rightarrow pre\ M \text{ and not } (S \text{ or } R);$



5

Charles André - UNSA

If then else

node COUNTER (init:int) returns (c:int);
 let c = init -> pre c + 2; tel

node DoubleCall (even:bool) returns (n:int);
 let
 n = if even then COUNTER(0) else
 COUNTER(1);
 tel

Réponse à la séquence: even = ff ff tt tt ff ff tt tt ff ?

6

Charles André - UNSA

Cohérence d'horloges

- Les entrées sont sur l'horloge de base
- Les constantes sont sur l'horloge de base
- Pour tout opérateur op , l'expression $(X \text{ op } Y)$ a pour horloge H si X et Y ont pour horloge H .
- $(X \text{ when } H)$ est correct si X et H sont sur la même horloge. L'horloge de $(X \text{ when } H)$ est alors H .
- L'horloge de $(\text{current } X)$ est l'horloge de X .

7

Charles André - UNSA

Importance des traces

Equation LUSTRE
 $n = i \rightarrow 1 + pre(n);$

Formules de récurrence:

$$n_1 = i_1$$

$$n_k = 1_k + n_{k-1} \text{ pour } k > 1$$

Instants	1	2	3	4	5
1	1	1	1	1	1
n	0	1	2	3	4
i	0	-	-	-	-

->
1 + pre(n)

8

Charles André - UNSA

-> et pre

Que fait $n = 0 \rightarrow 1 \rightarrow 2$?

\rightarrow est associatif

$$X \rightarrow Y \rightarrow Z = (X \rightarrow Y) \rightarrow Z = X \rightarrow (Y \rightarrow Z)$$

Vérification avec Lesar

```

node verif(X,Y,Z:bool) returns (ok:bool);
var l,m:bool;
let
  l = X -> (Y -> Z);
  m = (X -> Y) -> Z;
  ok = (l = m);
tel
    
```

9

Charles André - UNSA

Associativité du ->

Utilisons des variables intermédiaires :

$$A = X \rightarrow Y \quad \text{et} \quad B = Y \rightarrow Z$$

$$l = X \rightarrow B \quad \text{et} \quad m = A \rightarrow Z$$

Passons aux équations de récurrence :

$$l_1 = X_1, B_1 = Y_1 \Rightarrow l_1 = X_1$$

$$\text{Pour } k > 1: l_k = B_k, B_k = Z_k \Rightarrow l_k = Z_k$$

$$m_1 = A_1, A_1 = X_1 \Rightarrow m_1 = X_1$$

$$\text{Pour } k > 1: m_k = Z_k, A_k = Y_k \Rightarrow m_k = Z_k$$

10

Charles André - UNSA

Que fait ?

$$X = A \rightarrow \text{pre}(X);$$

Instants	1	2	3	4	5
A	A1	A2	A3	A4	A5
X	A1	A1	A1	A1	A1
pre(X)	⊥	A1	A1	A1	A1

$$X = (A1, A1, A1, A1, A1,$$

11

Charles André - UNSA

Que fait ?

$$X = A \rightarrow \text{pre}(B \rightarrow \text{pre}(X));$$

Instants	1	2	3	4	5
A	A1	A2	A3	A4	A5
B	B1	B2	B3	B4	B5
pre(X)	⊥	A1	B1	A1	B1
B->pre(X)	B1	A1	B1	A1	B1
pre(B->pre(X))	⊥	B1	A1	B1	A1
X	A1	B1	A1	B1	A1

$$X = (A1, B1, A1, B1, A1,$$

12

Charles André - UNSA

Fibonacci

$$F_n = F_{n-1} + F_{n-2}; F_0 = F_1 = 1$$

Solution incorrecte :

```
F = 1 -> 1 -> pre F + pre pre F;
```

Une solution correcte :

```
F = 1 -> pre ( F + (0 -> pre F) );
```