Examen
Langages Temps Réel - LUSTRE (duration 1h30)
(All documents are allowed)

Question 1 Petite question de compréhension

Let us consider the following Lustre programs:

```
node integrator (X : int) returns (S : int);
let
     S = X + (0 -> pre(S));
tel;

node Prog1 (X : int) returns (S1, S2, S3 : int);
var B1, B2 : bool;                   B1的初始值为T,之后的值为(初始值为f -> 之后为上一值B1)
let      初始值   上一值 初始值      上一个值
     B1 = true -> pre(false -> pre(B1));
     B2 = true -> pre(false -> pre ( false -> pre(B2)));
     S1 = current (integrator (X when B1));
     S2 = current (integrator (X) when B2);
     S3 = current (S1 when B2);
tel;
```

Let us consider the input flow X = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1...) (X is always equal to 1).
Question= what are the values of:

- `integrator(X)`
- `B1`
- `B2`
- `S1`
- `S2`
- `S3`

You can answer with a time table giving the values of the flows over time. You can directly answer on the sheet given in the appendix (question 1), or use your own sheet.

Question 2 A mysterious Lustre program...

Let us consider the following Lustre program:

```
node Mysterious (X : int) returns (Y : int)
let
     Y = X -> pre ( X -> (Y + pre(Y))) ;
tel
```

Let us consider the input flow X = (0,1,2,3,4,5,6,7,8....).
Question: what is the output flow Y in response to X ?

Here again, you can answer with a time table giving the values of the flows over time. You can directly answer on the sheet given in the appendix (question 2), or use your own sheet.

Question 3 An avionic system...

The speed (Mach number) of an aircaft is computed by the following formula:

$$M = \boxed{\sqrt{\frac{PT}{PS} - 1}}$$

where PT is the "Total Pressure" and PS is the "Static Pressure" measured outside the aircraft. The avionic computer which calculates M takes as input PT and PS. It produces M and Boolean flow called BM. BM is the validity status of M:

- BM must be true in normal conditions, that is, when $0<PS\leq PT$
- BM must be false in abnormal conditions, that is, when $PS>PT$ or $PS\leq 0$ of $PT\leq 0$.

Write a Lustre program, called ADIRS, such that:

- ADIRS takes as input PT and PS and returns M and BM
- In normal conditions, ADIRS computes M and BM according to the specification above
- In abnormal conditions, ADRIS returns false for BM and the last correct value for M.
- In abnormal conditions, ADIRS does not compute illegal operations, that is, negative square root and division by zero.

The skeleton of this program is given in the appendix (question 3). You can complete this program directly on the appendix, or copy it and complete it on your sheet.

Question 4 A railway control system...

The aim of this exercise is to automatize a railway crossing. Let us consider a crossing as shown in Figure 3. This railway crossing is composed of the one-way tracks (note that the trains always run in the same direction).

Access to the crossing section is protected by two traffic lights Fi (i=1.2), one on each track. To simplify, we consider that each Fi is a Boolean flow

- equal to true to mean that the traffic light is green
- and equal to false to mean that it is red.

In front of each traffic light, there is sensor Ei (i=1.2). The arrival of a train in front of the traffic light activates the corresponding sensor.
To simplify, we consider each Ei is a Boolean flow:

- Ei is true when a train arrives in front of Fi, and it remains true as long as the train is blocked in front of the traffic light (that is, as along as the traffic light is red).
- Ei is false if there is no train in front of Fi.

For simplicity, we assume that the that the length of the trains is zero, that is, as soon as a train has crossed the traffic light, it enters the crossing section and Ei becomes immediately equal to false.

Let us consider a third sensor called C which detects the presence of a train in the crossing section. The presence of a train in the crossing section (whatever the track) set C to true. Contrary, the absence of train in the section set C to false.

For instance, imagine that a train arrives on the first track in front of F1, then E1 become true. Let us imagine that F1 is true (that is, green), then the train enters the crossing section, E1 becomes false (the is no more train in front of F1), and C becomes true (there is a train in one of the two tracks of the crossing section). Let us imagine that after few minutes the train leaves the section, the C becomes again ~~true~~. F

Write a Lustre program to control the states of the traffic lights F1 and F2 such that:

- **No-collision property**: There at most one train at each time in the crossing section (in order to avoid any collision between two trains)
- **No-starvation property**: Any train arriving in front of the traffic light will eventually be allowed to enter the crossing section (in order to avoid any train being blocked indefinitely before the traffic lights).

We assume that the trains respect the traffic lights.

Note: This problem cannot be reduced to a simple mutual exclusion problem. We indeed accept that two trains traveling on the same track (for example track 1) are sufficiently close so that at a given moment the first is in the crossing section while the second is in front of F1. There can therefore be two trains waiting for the crossing section (one on each track) while a third train is in the crossing section (as shown in Figure 1). However, we suppose that there is at most one train waiting in front of each traffic light.

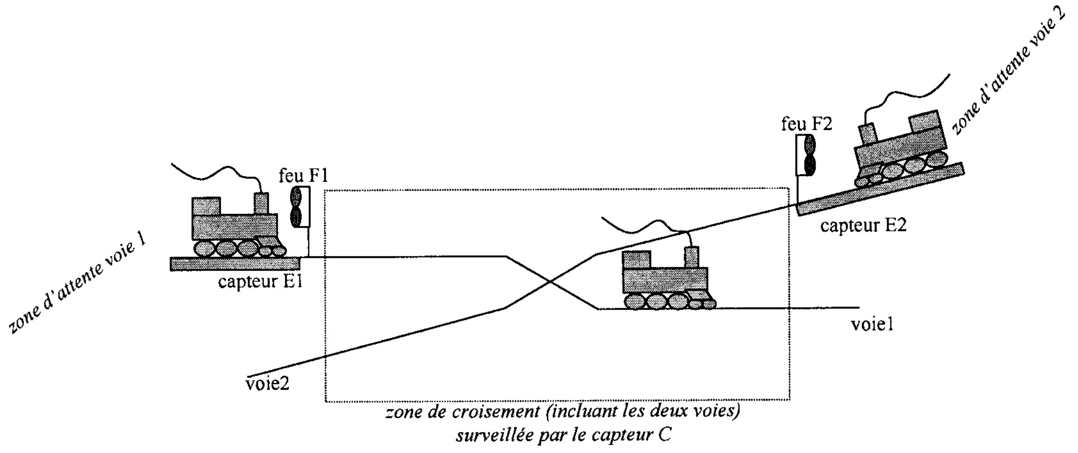

Figure 1.