Exercise : session N°1

## (Complement for the Fourth Net)

F.Kordon - MATCH School

## **1.** Fourth Net (correction)

In the model of Figure 1, two classes of processes are communicating together : a producer (P1, on the left) and a consumer (on the right). As the guard on T2a tells, producers and Consumers must have the same Id to communicate. Once the message sent, the producer waits for an acknowledge to send another message. In that example, there are two processes for each class. A priori, acknowledge are provided using an untyped signal.



You where asked to unfold that model to get an equivalent P/T net and check for bounds of places on this unfolded net. Unfolding provides you with the P/T equivalent net of Figure 2. When bounds are computed on this net, you get the following result :

(i) P2b\_proc\_1 : number of token in [0 ... 1] P2b proc 2: number of token in  $[0 \dots 1]$ **(ii)** (iii) P2a\_proc\_1: number of token in [0 ... 1] P2a\_proc\_2: number of token in [0 ... 1] (iv) P1b proc 1: number of token in [0 ... 1] **(v)** (vi) P1b\_proc\_2: number of token in [0 ... 1] (vii) P1a proc 1: number of token in  $[0 \dots 1]$ (viii) P1a\_proc\_2: number of token in [0 ... 1] (ix)  $Msg_proc_1$ : number of token in [0 ... 2] *Msg\_proc\_2* : number of token in [0 ... 2] **(x)** Ack : (xi) number of token in [0 ... 2]

We can be surprise by the bounds of places  $Msg\_proc\_1$  and  $Msg\_proc\_2$  (bounds *ix* and *x*). It means there could be in place Msg two messages from producer 1 to consumer 1 (or two messages from producer 2 to consumer 2). There should be a problem because a producer has to wait for an acknowledgment before sending the next message.



Having a look on the reachability graph of that Petri net, you can observe several path leading to such a state. Figure 3 shows one of them (they can be obtained using an appropriate query on a model checking tool like PROD). From this path, we get the following scenario :

**step 1:** producer 2 sends a message to consumer 2,

step 2: producer 1 sends a message to consumer 1,

step 3: consumer 1 takes the message from consumer 1,

step 4: consumer 1 sends an acknowledge to consumer 1,

step 5: producer 2 takes the acknowledge message,

step 6: producer 2 sends a second message to consumer 2.



Figure 3: An interesting path in the Reachability Graph of the Petri net model.

It is obvious that, if we want to force producers to send messages only after having got the appropriate acknowledge, then, acknowledges have to be typed after the identity of the couple <producer, consumer>. This modification leads to the model of Figure 4. The place *Ack* is now colored and a guard (similar to the one of transition T2a) added.



If we unfold this new net, we get the P/T net of Figure 4 that is obviously 1-bounded.

*Exercise: session N°1 (correction for the fourht net)* 

