

Mini-projet, module SVS

La technique du contrôle des modèles est l'une des techniques prometteuses pour la vérification automatique des systèmes informatiques. Les réseaux Petri (RDP) figurent parmi les formalismes les plus utilisés dans le domaine du génie logiciel. Malheureusement, les outils de modélisation et d'analyse à l'aide des RDP sont rarement dotés de techniques de vérification efficaces. De plus, la plupart de ces outils sont soit des outils non maintenus, soit ils utilisent des techniques de vérification limitées.

Dans ce mini-projet, nous proposons d'utiliser l'outil NuSMV [1] pour vérifier les modèles en réseaux de Petri.

L'objectif de ce mini-projet est double : au départ, nous cherchons à construire le graphe de marquage d'un réseau Petri. Ensuite, dans une deuxième étape, on transforme le graphe de marquage obtenu en un code NuSMV équivalent.

L'algorithme de transformation du graphe de marquage d'un RDP vers du code NuSMV, est montré sur la figure 1.

La deuxième Figure illustre un exemple d'utilisation de l'algorithme de la figure 1.

```

1: add MODULE main statement
2: add VAR keyword
3: for all  $M_i \in \mathcal{R}(M_0)$  do
4:   add  $s_i$  to the set of states  $s$                                  $\triangleright s : \{s_0, s_1, \dots\};$ 
5: end for
6: for all  $p_i \in P$  do
7:   if  $\forall_{M_j \in \mathcal{R}(M_0)} M_j(p_i) \leq 1$  then
8:     add Boolean variable  $p_i$                                      $\triangleright p_i : \text{boolean};$ 
9:   else
10:    add bounded Integer variable  $p_i$ 
11:                                      $\triangleright p_i : 0..k; \text{ where } \forall_{p_i \in P, M_j \in \mathcal{R}(M_0)} M_j(p_i) \leq k$ 
12:   end if
13: end for
14: add ASSIGN keyword
15: init  $s$  variable                                                 $\triangleright \text{init}(s) = s_0;$ 
16: open transition relation switch statement                         $\triangleright \text{next}(s) := \text{case}$ 
17: for all  $s_i \in s$  do
18:   add case  $s = s_i$ 
19:   for all  $s_j \in s$  do
20:     if  $\exists_{t \in T} M_i \xrightarrow{t} M_j$  then
21:       add  $s_j$  to  $s_i$  successors list
22:     end if
23:   end for
24:                                      $\triangleright s = s_i : \{s_{j1}, s_{j2}, \dots\};$ 
25: end for
26: close transition relation switch statement                       $\triangleright \text{esac};$ 
27: for all  $p_i \in P$  do
28:   open labelling function switch statement                       $\triangleright p_i := \text{case}$ 
29:   for all  $s_j \in s$  do
30:     if  $M_j(p_i) > 0$  then
31:       assign  $m$  to case  $s = s_j$ , where  $m = M_j(p_i)$            $\triangleright s = s_j : m;$ 
32:     end if
33:   end for
34:   if  $p_i$  is of Boolean type then
35:     set default value to FALSE                                   $\triangleright \text{TRUE} : \text{FALSE};$ 
36:   else
37:     set default value to 0                                       $\triangleright \text{TRUE} : 0;$ 
38:   end if
39: close labelling function switch statement                       $\triangleright \text{esac};$ 
40: end for

```

Figure 1 : Algorithme de transformation d'un graphe de marquage vers du code NuSMV [2]

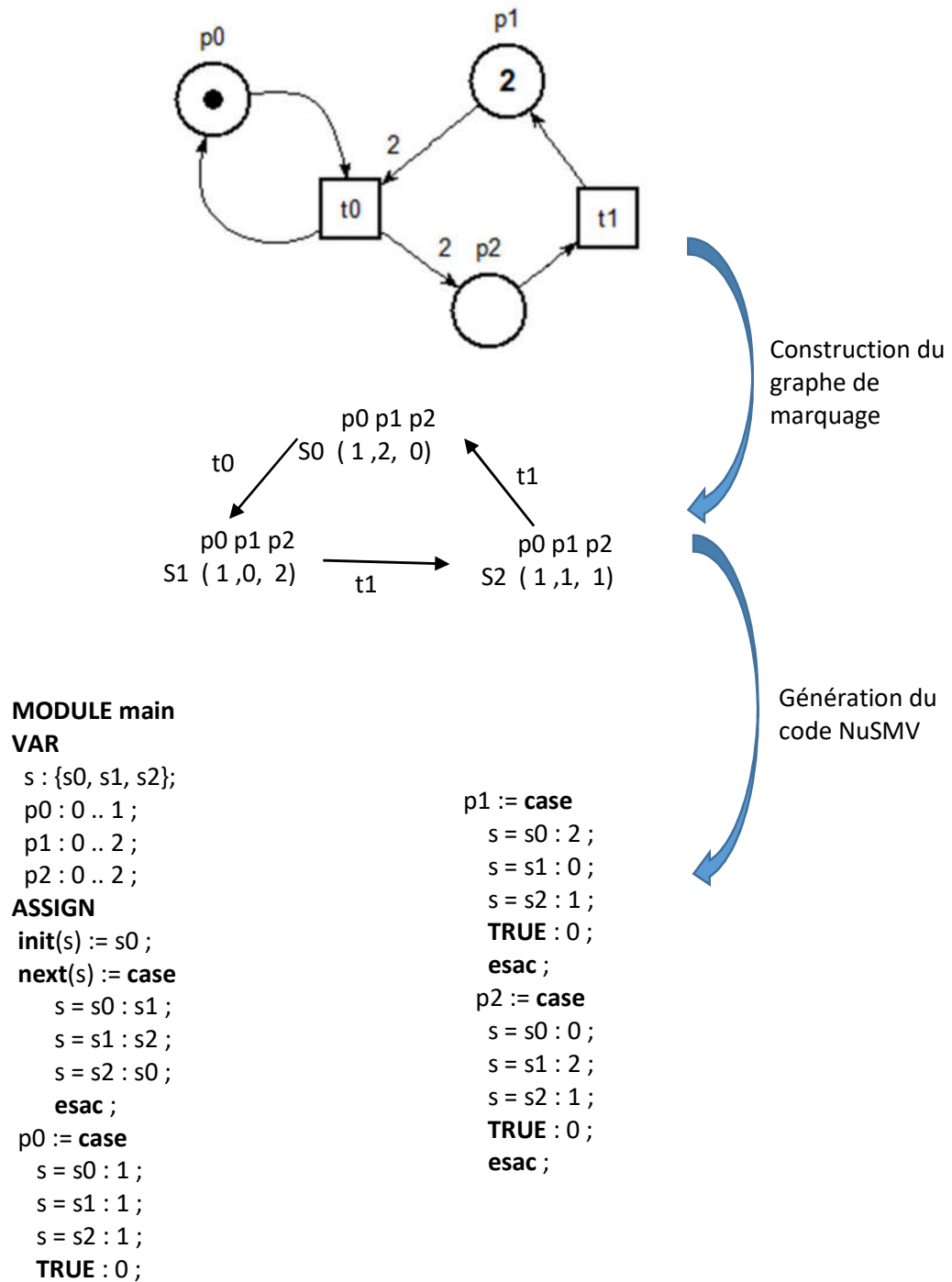


Figure 2 : une utilisation de l’algorithme de la figure 1 pour générer le code NuSMV

Références

1. Cimatti, A., Clarke, E., Giunchiglia, F., & Roveri, M. (2000). NuSMV: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer*, 2, 410-425.
2. Szpyrka, M., Biernacka, A., & Biernacki, J. (2014, September). Methods of Translation of Petri Nets to NuSMV Language. In *CS&P* (pp. 245-256).