TP3 - Ordonnancement de tâches sur deux machines

Paul Chaignon - Ulysse Goarant

21 février 2014

Listing 1 – taches.ecl

```
:- lib(ic).
:- lib(ic_symbolic).

:- local domain(machines(m1, m2)).

/**
 * Question 3.1
 * taches(?Taches)
 */
taches(Taches):-
  Taches = [](tache(3, [],      m1, _),
        tache(8, [],      m1, _),
        tache(8, [4, 5],  m1, _),
        tache(6, [],      m2, _),
        tache(3, [1],     m2, _),
        tache(4, [1, 7],  m1, _),
        tache(8, [3, 5],  m1, _),
        tache(6, [4],     m2, _),
        tache(6, [6, 7],  m2, _),
        tache(6, [9, 12], m2, _),
        tache(3, [1],     m2, _),
        tache(6, [7, 8],  m2, _)).

/**
 * Question 3.2
 *
 */
affiche(Taches):-
  (foreachelem(Tache, Taches)
  do
    writeln(Tache)
  ).

/**
 * Question 3.3
 * domaines(+Taches, ?Fin)
 */
domaines(Taches, Fin):-
  (foreachelem(tache(Duree, _, Machine, Debut), Taches),
  param(Fin)
  do
    Machine &:: machines,
    Debut #>= 0,
```

```
    Debut #=< Fin - Duree
  ).

/**
 * Question 3.4
 * getVarList(+taches, ?Fin, ?List)
 */
getVarList(Taches, Fin, [Fin|List]):-
  (foreachelem(tache(_, _, _, Debut), Taches),
  fromto([], In, Out, List)
  do
    Out = [Debut|In]
  ).

/**
 * Question 3.5
 * solve(?Fin)
 */
solve(Fin):-
  taches(Taches),
  domaines(Taches, Fin),
  precedences(Taches),
  conflits(Taches),
  getVarList(Taches, Fin, List),
  labeling(List),
  affiche(Taches).

/**
 * Question 3.6
 * precedences(+Taches)
 */
precedences(Taches):-
  (foreachelem(tache(_, Precedences, _, Debut), Taches),
  param(Taches)
  do
    (foreach(Precedence, Precedences),
    param(Debut), param(Taches)
    do
      tache(DureePred, _, _, DebutPred) is Taches[Precedence],
      Debut #>= DebutPred + DureePred
    )
  ).

/**
 * Question 3.7
 * conflits(+Taches)
 */
conflits(Taches):-
  (for(I, 1, 12),
  param(Taches)
  do
    (for(J, I+1, 12),
    param(Taches), param(I)
    do
      tache(Duree1, _, Machine1, Debut1) is Taches[I],
      tache(Duree2, _, Machine2, Debut2) is Taches[J],
      (Machine1 &= Machine2) => (Debut2#>=Debut1+Duree1 or
          Debut2#=<Debut1-Duree2)
```

```
101     )
102   ).
103
104 /**
105  * Question 3.8
106  */
107 Fin #< 43, solve(Taches, Fin).
```

Question 3.8

La première solution trouvée par ECLiPSe est sûrement la meilleure parce qu'il commence le labeling avec les valeurs les plus petites. Nous devrions donc trouver en premier la solution finissant la plus tôt. Cela n'est cependant pas sûr. Nous pouvons nous en assurer en ajoutant une contrainte à la recherche de solutions (voir code 3.8).