TP3 - Ordonnancement de tâches sur deux machines

Paul Chaignon - Ulysse Goarant

17 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,
```

```
44      Debut #=< Fin - Duree
45    ).
46
47  /**
48   * Question 3.4
49   * getVarList(+taches, ?Fin, ?List)
50   */
51  getVarList(Taches, Fin, [Fin|List]):-
52    (foreachelem(tache(_, _, _, Debut), Taches),
53    fromto([], In, Out, List)
54    do
55      Out = [Debut|In]
56    ).
57
58  /**
59   * Question 3.5
60   * solve(?Fin)
61   */
62  solve(Fin):-
63    taches(Taches),
64    domaines(Taches, Fin),
65    precedences(Taches),
66    conflits(Taches),
67    getVarList(Taches, Fin, List),
68    labeling(List),
69    affiche(Taches).
70
71  /**
72   * Question 3.6
73   * precedences(+Taches)
74   */
75  precedences(Taches):-
76    (foreachelem(tache(_, Precedences, _, Debut), Taches),
77    param(Taches)
78    do
79      (foreach(Precedence, Precedences),
80      param(Debut), param(Taches)
81      do
82        tache(DureePred, _, _, DebutPred) is Taches[Precedence],
83        Debut #>= DebutPred + DureePred
84      )
85    ).
86
87  /**
88   * Question 3.7
89   * conflits(+Taches)
90   */
91  conflits(Taches):-
92    (for(I, 1, 12),
93    param(Taches)
94    do
95      (for(J, I+1, 12),
96      param(Taches), param(I)
97      do
98        tache(Duree1, _, Machine1, Debut1) is Taches[I],
99        tache(Duree2, _, Machine2, Debut2) is Taches[J],
100       (Machine1 &= Machine2) => (Debut2#>=Debut1+Duree1 or
            Debut2#=<Debut1-Duree2)
```

```
    )
  ).

/**
 * Question 3.8
 */
%Fin #< 43, solve(Taches, Fin).
```

Question 3.8