

TD de *Programmation logique et par contraintes* n° 3

Le problème du M1

L'UFR d'informatique de Paris Diderot propose les trois *parcours* suivants pour le M1 informatique:

- *Information et Mobilité: Programmation, Algorithmes pour Internet, Répartition et Systèmes* (IM-PAIRS).
- *Langages et Programmation* (LP) .
- *Informatique Recherche* (IR) .

Un *parcours* est caractérisé par trois ensembles d'UE (*unités d'enseignement*) : les UE *obligatoires*, les UE *majeures* et les UE *suggérées* du parcours.

Une UE est caractérisée par: le code de l'UE, les *ects* (*european credit transfer system*) de l'UE, le semestre de l'UE.

Les données du problème, en Prolog, sont fournies page 2.

Un *projet de parcours* est un ensemble de UE.

- Un projet de parcours P est *cohérent* si:
 - P contient exactement une UE parmi *an1* et *an2*.
 - P contient au moins une UE parmi *se*, *pp* et *tr*.
 - la somme des *ects* des UE de P, pour chaque semestre, est au moins égale à 30.
- Un projet de parcours P est *en accord avec* un parcours si
 - P contient toutes les UE obligatoires du parcours.
 - P contient aux moins trois UE majeures du parcours (sauf pour le parcours IR, qui n'a pas d'UE majeures)
 - la somme des *ects* des UE du projet qui sont obligatoires, majeures ou suggérées pour le parcours est au moins égale à 42.
- Un projets de parcours P est *acceptable* s'il est cohérent et il est en accord avec au moins un parcours.

Exercice 1 Ecrire un programme qui fournit comme interface les prédicats *coherent/1*, *en_accord/2* et *acceptable/1*, spécifiés ci-dessus.

Voici une spécification détaillée d'une implantation possible de ces prédicats.

Dans la suite, un projet de parcours est décrit par un liste de codes d'UE.

1. Copier de `/ens/bucciare/pub/prolog/m1.pl` les définition des prédicats *ue/3* et *parcours/4*.
2. Définir un prédicat *anglais/1* tel que *anglais(L)* réussit si et seulement si L est un projet de parcours qui contient exactement une UE parmi *an1* et *an2*.

3. Définir un prédicat `fin_etudes/1` tel que `fin_etudes(L)` réussit si et seulement si `L` est un projet de parcours qui contient au moins une UE parmi `se`, `pp`, et `tr`.
4. Définir un prédicat `ects_sem/3` tel que `ects_sem(1,L,N)` réussit si et seulement si `N` est le nombre d'ects du premier semestre dans le projet de parcours `L`, et `ects_sem(2,L,N)` réussit si et seulement si `N` est le nombre d'ects du deuxième semestre dans le projet de parcours `L`.
5. Définir le prédicat `coherent/1`.
6. Définir un prédicat `inclus/2` tel que `inclus(L1,L2)` réussit ssi tous les éléments `e` la liste `L1` sont aussi des éléments de la liste `L2`.
7. Définir un prédicat `inter/3` tel que `inter(L1,L2,L3)` réussit ssi la liste `L3` contient exactement les éléments communs aux listes `L1` et `L2`.
8. Définir un prédicat `union/3` tel que `inter(L1,L2,L3)` réussit ssi la liste `L3` contient l'union des éléments des listes `L1` et `L2`.
9. Définir un prédicat `compte/2` tel que `compte(L,N)` réussit ssi `N` est la somme des ects des UE de la liste `L`.
10. Définir le prédicat `en_accord/2`.
11. Définir le prédicat `acceptable/1`.
12. Appliquer ces prédicats a votre propre projet de parcours.

Les UE:

```

ue(plc,6,1). /*Programmation Logique et par Contraintes */
ue(cc,6,1). /*Calculabilité et complexité */
ue(alg,6,1). /*Algorithmique*/
ue(res,6,1). /*Protocoles réseaux*/
ue(xml,6,1). /*Formats de documents et XML*/
ue(ca,6,1). /*Circuits et architecture */
ue(int,3,1). /*Interprétation des programmes*/
ue(ia,6,1). /*Introduction à l'Intelligence Artificielle et à la théorie des jeux*/
ue(pf,6,1). /*Programmation Fonctionnelle Avancée */
ue(po,6,1). /*Langages à objets avancés*/
ue(tg,3,1). /*Théorie des graphes*/
ue(pcm,6,1). /*Programmation des composantes mobiles*/
ue(ig,6,1). /*Infographie */
ue(an1,3,1). /* Anglais */
ue(an2,3,2). /* Anglais */
ue(ill,6,2). /*Introduction aux Logiciels Libres */
ue(tpc,6,2). /*Théorie et pratique de la concurrence */
ue(bd,6,2). /*Bases de données avancées */
ue(sem,6,2). /*Sémantique */
ue(ntw,3,2). /*Nouvelles tendances du web */
ue(alga,6,2). /*Algorithmique Avancée */

```

```

ue(auta,6,2). /*Automates Avancés */
ue(comp,6,2). /*Compilation */
ue(di,3,2). /*Droit de l'Informatique */
ue(gl,6,2). /*Génie Logiciel */
ue(ps,6,2). /*Programmation système avancée */
ue(intg,6,2). /*Interfaces graphiques */
ue(pao,6,2). /*Preuves Assistées par Ordinateurs */
ue(te,6,2). /*Technologies émergentes */
ue(tece,3,2). /*Techniques d'expression */
ue(se,6,2). /*Stage*/
ue(pp,6,2). /*Projet de Programmation*/
ue(tr,6,2). /*Travaux de recherche encadrés */

```

Les parcours:

```

/*Langages et Prog.*/
parcours(lp,
[int], /*OBL*/
[plc,pf,po,gl], /* MAJ*/
[cc,alg,xml,ca,pcm,ill,tpc,bd,sem,ntw,comp,di,ps,intg,pao,te,tece,se,pp]). /*SUG*/

/*Information et Mobilité: Prog., Algo. pour Internet, Répartition et Systèmes */
parcours(impairs,
[], /*OBL*/
[alg,res,bd,ps], /* MAJ*/
[cc,xml,ca,int,pf,po,tg,pcm,ig,ill,tpc,ntw,alga,comp,di,gl,intg,te,tece,se,pp]). /*SUG*/

/*Info. Recherche */
parcours(ir,
[cc,alg,int,tr], /*OBL*/
[], /* MAJ*/
[plc,ia,pf,tg,ig,tpc,sem,alga,auta,comp,pao]). /*SUG*/

```