

Travail pratique Prolog

Agencement

1. But

Le but de ce travail est de mettre en pratique les concepts de la programmation logique pour implanter un système qui réalise un agencement de meubles et d'appareils pour une cuisine.

2. Spécifications

Le fichier Agencement.pl comporte des définitions de prédicats permettant la gestion d'un agencement de meubles et d'appareils pour une cuisine. Vous devez compléter ce travail et définir les prédicats suivants:

- *lesIds(Blocs,Css)* unifie *Css* à la liste des listes des identificateurs des composants d'une liste de blocs.
- *lesComposantsDuType(IdAgencement,Ty,Compos)* unifie *Compos* à la liste des composants de l'agencement qui sont du type *Ty*.
- *estStrictementInclus(BlocY,BlocX)* est vrai si et seulement si *BlocY* est strictement inclus dans *BlocX* peu importe l'ordre des composants dans les blocs.
- *lesVoisins(Dir,Axe,Ag,Compo,Voisins)* où *Dir* est une direction, *Axe* est un axe, *Ag* est un agencement, *Compo* est un composant, *Voisins* s'unifie à la liste des voisins de *Compo* sur l'axe *Axe* et dans la direction *Dir*.

Ces prédicats existent déjà dans le fichier Agencement.pl, ils donnent cependant des résultats incorrects et doivent donc être modifiés.

3. Exemples

Dans les exemples on utilise les faits suivants :

```
composant(meuble ,b1,(1.0,1.0,1.0),position(0.0, 0.0, 0.0),coin).
composant(electro,b2,(1.0,1.0,1.0),position(1.0, 0.0, 0.0),laveVaisselle).
composant(meuble,b3,(1.0,1.0,1.0),position(0.0, 1.0, 0.0),bas).
composant(meuble,b4,(1.0,1.0,1.0),position(2.0, 0.0, 0.0),bas).
composant(meuble,h1,(1.0,1.0,0.5),position(2.0, 0.0, 2.0),haut).
composant(meuble,h2,(1.0,1.0,0.5),position(3.0, 0.0, 2.0),haut).
composant(meuble,b5,(1.0,1.0,1.0),position(4.0, 0.0, 0.0),haut).
composant(meuble,b6,(1.0,1.0,1.0),position(5.0, 0.0, 0.0),haut).
```

```
bloc(0,[axCmp(y,b1),axCmp(y,b3)]).
bloc(1,[axCmp(x,b1),axCmp(x,b2),axCmp(x,b4)]).
bloc(2,[axCmp(x,b5),axCmp(x,b6)]).
bloc(7,[axCmp(x,b1),axCmp(x,b2),axCmp(y,b3),axCmp(x,b4)]).
```

```
agencement(ag,[b1,b2,b3,b4,b5,b6,h1,h2]).
```

Les évaluations suivantes concernent les prédicats à réaliser, elles ont été établies à partir de versions correctes, similaires à celles que vous devez obtenir :

```
| ?- bloc(1,Bloc1),bloc(2,Bloc2),lesIds([Bloc1,Bloc2],Ids).
Ids = [[b1,b2,b4],[b5,b6]],
Bloc1 = [axCmp(x,b1),axCmp(x,b2),axCmp(x,b4)],
```

```

Bloc2 = [axCmp(x,b5),axCmp(x,b6)] ?
yes

| ?- lesComposantsDuType(ag,bas,Bs).
Bs = [b3,b4] ?
yes

| ?- bloc(1,B1),bloc(7,B7),estStrictementInclus(B1,B7).
B1 = [axCmp(x,b1),axCmp(x,b2),axCmp(x,b4)],
B7 = [axCmp(x,b1),axCmp(x,b2),axCmp(y,b3),axCmp(x,b4)] ?
yes

| ?- bloc(0,B0),bloc(1,B1),estStrictementInclus(B0,B1).
No

| ?- lesVoisins(droite,y,ag,b3,Bs).
Bs = [b1] ?
yes

| ?- lesVoisins(gauche,x,ag,b4,Bs).
Bs = [b2,b1] ?
yes

```

4. Directives pédagogiques

- Le fichier Agencement.pl constitue la base du travail.
- Le fichier Main.pl contient des tests. Vos prédicats, pour être considérés comme corrects, doivent **au minimum** vérifier les tests qui figurent dans ce fichier. Ces tests donnent aussi des indications précieuses sur la nature des données que doivent accepter vos prédicats ainsi que sur les résultats que vous devez obtenir. Pour réaliser ces tests, vous devez compiler le fichier Main.pl. Vous avez intérêt à produire aussi vos propres tests qui devraient couvrir des cas qui ne sont pas déjà considérés par les tests du fichier fourni.
- Le travail peut être réalisé en équipe de deux étudiants au plus.
- La remise du fichier Agencement.pl complété avec vos définitions doit être faite au plus tard le 26 avril 2013. Une pénalité de 5% sera appliquée à la note pour chaque jour de retard. Le délai maximum est de 7 jours, au delà, la note 0 sera attribuée au travail.
- Vous devez remettre, en plus, un listing DOCUMENTÉ du programme identifiant les membres du groupe (la qualité de la documentation sera un critère d'évaluation important). Vous devez commenter chaque prédicat que vous avez réalisés, ceux demandés et ceux que vous avez ajoutés et qui sont utilisés dans le code des prédicats demandés. Cette documentation doit porter sur le code ET donner toutes les directives nécessaires à l'utilisation de vos prédicats :
 - nature des arguments,
 - formes possibles d'utilisation,
 - restrictions d'utilisation,
 - exemples documentés d'utilisation.
- Ce listing pourra être remis dans la chute de courrier du département d'informatique au 4^e étage du

pavillon Président Kennedy. Indiquez au surligneur jaune ou en caractères gras les parties que vous avez ajoutées.

- Le fichier du programme devra être remis sur le site moodle du cours.
- Les remises du fichier et du listing sont nécessaires à la correction du TP et sont donc indispensables.
- Le programme remis doit pouvoir être compilé et exécuté **SANS MODIFICATION** à l'aide de Sicstus Prolog.