

Représentation de la Connaissance

Répétition 6

31 octobre 2006

Astuce du jour

Le prédicat prédéfini `bagof(X, P, L)` produit la liste L de tous les objets X qui satisfont le goal P . Il va de soit qu'il faut que X et P partagent au moins une variable commune pour que cela ait un sens. Soit par exemple, l'ensemble de faits suivant :

```
age(francois,25).  
age(arnaud,23).  
age(julien,21).  
age(robert,23).
```

Il est facile d'obtenir la liste des personnes ayant un age donné :

```
?- bagof(Name,age(Name,21),List).  
List=[julien,robert]
```

Il est tout aussi aisé d'obtenir les différentes listes correspondant aux différents ages possibles :

```
?- bagof(Name,age(Name,Age),List).
```

```
Age=25  
List=[francois];
```

```
Age=21  
List=[julien,robert];
```

```
Age=23  
List=[arnaud];
```

```
no
```

Si `bagof` ne trouve pas de solution on aura simplement un fail. Si le même objet X est trouvé plusieurs fois il apparaîtra autant de fois dans la liste.

Le prédicat `setof(X, P, L)` fonctionne comme `bagof` sauf qu'il trie la liste L à l'aide du prédicat prédéfini `@<` et élimine les doublons.

Lorsqu'on utilise le prédicat `findall(X, P, L)`, tous les X sont collectés indépendamment des différentes solutions possibles pour des variables de P non partagées avec X . Comparez l'exemple suivant avec l'appel à `bagof` correspondant vu plus haut :

```
?- findall(Name,age(Name,Age),List).  
List=[francois,julien,arnaud,robert]
```

Remarquez aussi que si aucun objet X ne satisfait P , `findall` est vrai et renvoie $L = []$.

Exercices mathématiques

1. idealbag

Après avoir réussi votre année, vous décidez de prendre des vacances en avion. La compagnie avec laquelle vous voyagez impose que votre valise ne pèse pas plus de 20 kilogrammes. Vous disposez d'une liste d'objets que vous désirez prendre avec mais le poids total de ce que vous comptiez prendre dépasse de loin les 20 kilogrammes. Il vous faut donc faire un choix en fonction de la masse et de l'utilité de chaque objet. On représentera un objet par `obj(nom,masse,utilité)`. Ecrire un prédicat `idealbag` dont vous choisirez vous même le nombre et le type des arguments et qui vous permettra d'obtenir la liste des objets les plus utiles à prendre avec.

Exercices proposés

2. Ecrire un prédicat permettant de calculer la transposée d'une matrice.
3. Ecrire un prédicat permettant de calculer le déterminant d'une matrice.