TP2 – Prolog & la résolution de problèmes

Lucas Foulon, Erwan Guillou, Nathalie Guin, Marie Lefevre

LE MONDE ANIMAL

Question 1 : Rédiger un programme Prolog capable de résoudre le problème suivant :

- la chèvre est un animal herbivore
- le loup est un animal cruel
- un animal cruel est carnivore
- un animal carnivore mange de la viande
- un animal herbivore mange de l'herbe
- un animal carnivore mange des animaux herbivores
- les carnivores et les herbivores boivent de l'eau
- un animal consomme ce qu'il boit ou ce qu'il mange

Y-a-t-il un animal cruel et que consomme-t-il ?

Vous devez obtenir que le loup est cruel et qu'il mange de la viande, des chèvres et de l'eau.

LE DISTRIBUTEUR DE BILLET

On s'intéresse à un distributeur automatique de boissons. Ce distributeur propose un certain nombre de boissons et peut rendre la monnaie. Pour fonctionner, l'utilisateur insère des pièces de monnaie, puis il sélectionne une boisson, dont le prix est affiché en Euros. L'utilisateur peut insérer des pièces de 10, 20 ou 50 centimes et des pièces de 1 ou 2 Euros. Les prix des boissons sont des multiples de 10 centimes.

Question 2 : Écrire le prédicat permettant de calculer la monnaie à rendre. Sachant que le distributeur a en réserve un certain nombre de chaque pièce, le prédicat pourra s'écrire :

```
monnaie(ArgentDonne, Prix, ArgentRendu, P2, P1, P50, P20, P10)
```

où les deux premiers paramètres reflètent les actions de l'utilisateur avec des montants en centimes, le troisième le détail des pièces que le distributeur va lui rendre, et les autres, le stock de chaque pièce dans le distributeur.

Vous pouvez procéder de deux manières :

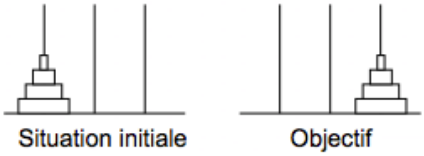
- soit faire un algorithme glouton qui donne une seule solution avec le moins de pièces possible,
- soit faire un algorithme de type « générer et tester » qui sera moins efficace mais permettra d'obtenir toutes les solutions possibles.

LES TOURS DE HANOÏ

Pour être sûr que vous ayez bien compris le principe de **résolution de problème par décomposition**, nous allons recoder le problème des tours de Hanoi que nous avons vu en cours.

Pour rappel, le problème est le suivant :

On dispose de 3 socles nommés respectivement socleA, socleB et socleC. Sur socleA sont empilés N disques circulaires, du plus grand au plus petit, formant une pyramide. Le but est de passer les disques de socleA à socleC, en s'aidant de socleB, de manière à obtenir exactement la même pyramide. La règle est que l'on ne déplace qu'un disque à la fois, et qu'on ne peut placer un disque sur un disque de diamètre inférieur.



Question 1 : Définir un prédicat `hanoi(N)` qui affiche la succession des déplacements à opérer d'un socle à l'autre.

```
?- hanoi(3).
Deplacement du disque du socle a vers le socle c
Deplacement du disque du socle a vers le socle b
Deplacement du disque du socle c vers le socle b
Deplacement du disque du socle a vers le socle c
Deplacement du disque du socle b vers le socle a
Deplacement du disque du socle b vers le socle c
Deplacement du disque du socle a vers le socle c
true ;
false.
```