Examen Prolog — ESSI2

Michel Rueher

4 Février 1997

Documents autorisés: *supports (et notes) de cours ou TD*

1 Générateur de plans pour robots

On voudrait déplacer trois cubes A, B et C disposés sur les emplacements 1, 2 et 3 (situation **Init**) pour obtenir la configuration finale suivante (situation **Final**). On dispose pour cela d'un petit robot qui sait déplacer les cubes: quand il reçoit la commande `transfert(X,I,J)`, le robot enlève le cube `X` à la position `I` et le pose à la position `J`. Le robot peut uniquement déplacer les cubes au sommet d'une pile.

Ecrivez le prédicat `plan_robot(P,N)` où `P` est une suite de commandes pour déplacer les cubes de la configuration initiale à la configuration finale et `N` le nombre maximal d'opérations.

```
                                                  ---
                                                 | A |
  ---                                             ---
 | A |                                           | B |
  ---      ---                                    ---
 | C |    | B |                                  | C |
  ---      ---                                    ---
----------------------------     -------------------------
   1        2        3             1        2        3

           (Init)                        (Final)
```

Exemple:

```
| ?- robot(A,5).

A = [t(a,c,b),t(c,1,3),t(a,b,1),t(b,2,c),t(a,1,b)]
```

2 Expliquez ...

Expliquer le fonctionnement du prédicat `prouve/2` ci-dessous. Vous pouvez utiliser un exemple.

```
prouve(Nil) :-
      Nil = vide ; Nil = [].
prouve([Goal|OtherGoal]) :-
      list_of_clauses(L),
      member([Goal,Tail],L),
      prouve(Tail),
      prouve(OtherGoal).
```

3 Un prédicat power_set/2...

Le prédicat `power_set(Set, Power_Of_Set)` doit calculer:

- l'ensemble des parties `Power_Of_Set` à partir de l'ensemble `Set`;
- l'ensemble `Set` à partir de l'ensemble de ses parties `Power_Of_Set`.

Algorithme Général:

```
 1. If Set = {}, Power_Of_Set = {{}}
 2. If Set = {Elt} U Rest, then
 2.1.      Let Power_Of_Rest = power_set(Rest)
 2.2.    Power_Of_Set = {{Elt} U X | X in Power_Of_Rest} U Power_Of_Rest
```

Exemples:

```
?- power_set(X, [[1,2],[1],[2],[]]).
X = [1, 2]

?- power_set(X, [[1,2],[2],[1],[]]).
fail.

?- power_set([1,2,3], X).
X = [[1, 2, 3], [1, 2], [1, 3], [1], [2, 3], [2], [3], []]
```

1. L'algorithme ci-dessus effectue un certain nombre de choix qui autorisent une efficacité raisonnable. Lesquels ?
2. Ecrire `power_set/2` en Prolog.