



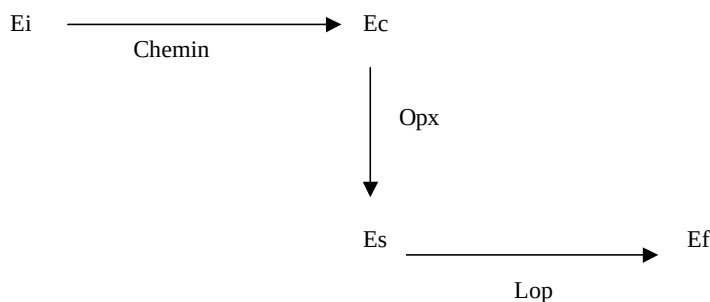
Résolution de problèmes par recherche dans un graphe d'états

On considère des problèmes du type recherche d'un chemin entre un état initial E_i et un état final E_f , avec des opérateurs de transition. L'ensemble des états qu'il est possible d'atteindre à partir de l'état initial en appliquant tous les opérateurs possibles constitue l'ensemble des sommets du graphe d'états, que l'on développe sous forme d'arborescence. On utilise ici une recherche en profondeur d'abord, particulièrement bien adaptée à PROLOG.

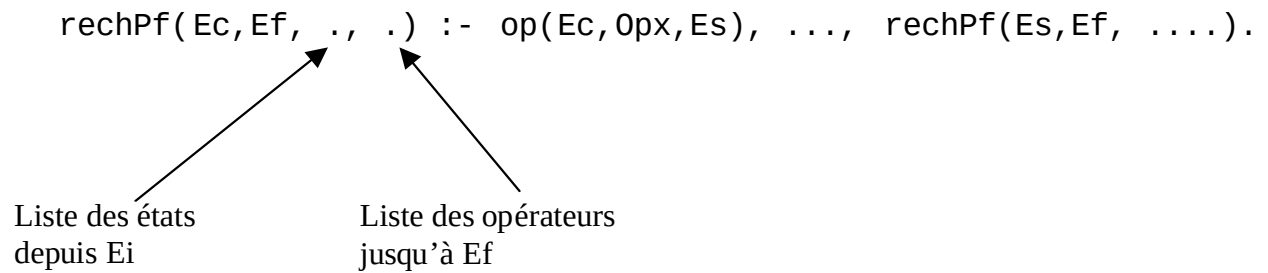
1 Recherche en profondeur dans un graphe d'états

Pour un problème, l'état initial est connu par le fait `initial( $E_i$ )`. L'état final est connu par le fait `final( $E_f$ )`. Les opérateurs sont connus par le prédicat `op( $E_c$ ,  $O_{px}$ ,  $E_s$ )`, E_c désignant l'état courant, O_{px} un opérateur de transition et E_s l'état de sortie, atteint par application de O_{px} .

- On atteint l'état E_c à partir de E_i en passant par une liste d'états mémorisés dans la liste `Chemin`.
- L'application de l'opérateur O_{px} à l'état courant E_c fait passer à un état E_s .
- `Lop` est la liste des opérateurs qu'il faut appliquer depuis E_s pour atteindre E_f .



1.1 Définir le prédicat de recherche en profondeur `rechPf` :



1.2 Définir le prédicat `resoudre(S)` qui donne la liste `S` des opérateurs à appliquer pour passer de l'état initial à l'état final.

On va maintenant appliquer le prédicat `rechPf` aux deux problèmes suivants, en définissant pour chacun d'entre eux l'état initial, l'état final et les opérateurs.

2 Le problème des flèches

2.1 Définition du problème

Six flèches sont dans la position de la figure 1. On souhaite les positionner comme dans la figure 2.

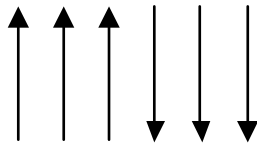


Figure 1

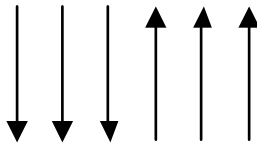


Figure 2

On symbolise ces deux états par « hhhbbb » et « bbbhhh » avec « h » pour « flèche haute » et « b » pour « flèche basse ».

On définit quatre opérateurs de transition :

- R1 : retournement de deux flèches hautes adjacentes (« hh » devient « bb »)
- R2 : retournement d'une flèche haute et d'une flèche basse adjacentes (« hb » devient « bh »)
- R3 : retournement d'une flèche basse et d'une flèche haute adjacentes (« bh » devient « hb »)
- R4 : retournement de deux flèches basses adjacentes (« bb » devient « hh »)

2.2 Modéliser le problème

- Définir un prédicat `rempl(S1, S2, L1, L2)` qui est satisfait si le remplacement de la sous-liste `S1` par `S2` dans la liste `L1` donne `L2`.
- Définir les prédicats `initial`, `final` et `op` pour le problème des flèches et utiliser le prédicat `résoudre` pour trouver des solutions au problème.

3 Le problème de la simplification d'un nombre

3.1 Définition du problème

On souhaite simplifier un nombre considéré comme une liste de chiffres. On prendra l'exemple du nombre 3413242341. L'objectif est d'arriver au mot vide. Les règles de simplification sont les suivantes :

- R1 : si deux chiffres adjacents sont égaux, on peut les supprimer tous les deux.
- R2 : on peut permuter deux chiffres adjacents si la valeur absolue de leur différence est supérieure à 1.
- R3 : la séquence $(n, n-1, n)$ peut être remplacée par la séquence $(n-1, n, n-1)$.

3.2 Résolution du problème

Définir les prédicats `initial`, `final` et `op` pour ce problème et utiliser le prédicat `résoudre` pour lui trouver des solutions.

4. Amélioration de la recherche

En observant la résolution du problème des flèches, on constate que le troisième état généré était très proche de l'état final. Or l'ordre figé des opérateurs et le fait que la première substitution possible est appliquée ont conduit le programme vers une solution plus longue.

Nous allons donc améliorer la recherche en profondeur en examinant tous les états auxquels peuvent conduire les opérateurs applicables. Dans le cas où un de ces états est l'état final, c'est la règle qui y conduit qui sera appliquée. Sinon, on utilise comme précédemment le premier opérateur applicable.

- Définir le prédicat `resoudre+`, qui utilise le prédicat `rechPf+` implémentant cette stratégie.

5. Pour aller plus loin

Pour ceux qui ont le temps d'en faire plus : on peut encore améliorer la recherche, en définissant pour chaque problème un prédicat calculant une distance entre deux états. Dans le prédicat de recherche, on considère tous les états auxquels peuvent conduire les opérateurs applicables, et l'on choisit l'opérateur qui conduit à l'état le plus proche de l'état final (au sens de la distance précédemment définie).

6. Procédure de notation du TP

Ce TP sera noté à la dernière séance des travaux dirigés. Un court rapport de synthèse et une démonstration seront demandés.