

# Programmation Logique : Prolog

Valentin NOEL et Romain Pascual

14 juin 2017

## 1 Problème 1 : Traversées de rivière

Les problèmes des traversées de rivière supposent généralement que l'on doit faire passer des individus d'un côté à l'autre de la rivière en ayant un nombre limité de personnage sur le bateau tout en interdisant à certains personnages de rester ensemble d'un même côté de la rivière.

Ici le problème est le suivant : "Lulu doit faire passer le chou, la chèvre et le loup de l'autre côté de la rivière et il n'a qu'une place sur son bateau. Si la chèvre et le chou sont ensemble sur une rive quand Lulu s'éloigne, la chèvre mange le chou. Et si le loup et la chèvre sont ensemble quand Lulu s'éloigne, le loup mange la chèvre."

On cherche à modéliser le problème en Prolog. La solution est donnée dans le fichier "rivière.pl"

La modélisation est la suivante : on utilise un tuple (*Lulu*, *Chou*, *Chevre*, *Loup*) de booléens tel que chaque variable vaut 0 si le personnage est du premier côté de la rivière et 1 s'il est du second. On cherche donc à transformer le tuple (0, 0, 0, 0,) en (1, 1, 1, 1). Pour cela on commence par définir les états interdits (le loup et la chèvre d'un même côté ou encore la chèvre et le chou du même côté). Ensuite on modélise les déplacements possibles : déplacement du chou, de la chèvre, du loup ou de Lulu en interdisant les déplacements qui conduisent à un état interdit.

On cherche ensuite la solution à l'aide d'un parcours en largeur implémenté par des listes.

Le problème est finalement résolu par analogie avec une réflexion sur les graphes où les nœuds représentent les différents états licites et les arcs les déplacements du bateau. La "meilleure" solution est alors obtenue à l'aide d'un parcours en largeur du graphe. Le code du parcours en largeur a été trouvé sur internet et adapté à la situation.

Pour exécuter le code, il suffit d'utiliser la commande `solution` ou la commande `solution_ecrite` dans Prolog après avoir chargé le fichier "rivière.pl". Dans le premier cas, on obtient la liste des situations depuis celle où tout le monde est sur la première berge jusqu'à celle où tout le monde est sur la seconde berge. Avec la deuxième commande, on obtient la liste écrite des déplacements de Lulu, comme le montre la sortie ci-dessous :

```

romain@romain-N551JK:~/Documents/Perso/RepsGit/ProjetProlog$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [riviere].
true.

?- solution.
[situation(0,0,0,0),situation(1,0,1,0),situation(0,0,1,0),situation(1,1,1,0),
situation(0,1,0,0),situation(1,1,0,1),situation(0,1,0,1),situation(1,1,1,1)]
true.

?- solution_ecrite.
Lulu se déplace avec la chevre de la berge 1 à la berge 2.
Lulu se déplace seul de la berge 2 à la berge 1.
Lulu se déplace avec le chou de la berge 1 à la berge 2.
Lulu se déplace avec la chevre de la berge 2 à la berge 1.
Lulu se déplace avec le loup de la berge 1 à la berge 2.
Lulu se déplace seul de la berge 2 à la berge 1.
Lulu se déplace avec la chevre de la berge 1 à la berge 2.
Tout le monde a traversé
true .

?- 

```

## 2 Problème 2 : Le compte est bon

"Le compte est bon est un jeu (télévisé) qui consiste à trouver une combinaison arithmétique de nombres afin d'obtenir un certain total."