Projet Prolog – L2 MIASHS – 2025

Comme indiqué le 5 février, le projet consiste à concevoir en équipe (de 1 à 3 étudiant.es) un programme Prolog qui joue au jeu indiqué ci-dessous contre un autre programme.

Règle du jeu

Ce jeu se joue à 2 joueurs. Le but est d'obtenir le score le plus élevé après plusieurs tours.

VERSION 1. Chaque joueur choisit un nombre entre 1 et 5 et les 2 joueurs les prononcent en même temps. Appelons A et B ces nombres. Si |A-B|=1 (donc si les deux nombres sont consécutifs), le joueur qui a proposé le nombre le plus petit obtient la valeur A+B et l'autre 0. Sinon, chaque joueur obtient la valeur de son nombre.

Par exemple, si le premier joueur dit 4 et le second dit 2, le premier gagne 4 points et le second gagne 2 points. Au tour suivant, si le premier joueur dit 4 et le second dit 3, le premier joueur a 0 points et le second joueur a 7 points. Les scores sont : 4 (joueur 1) vs 9 (joueur 2). Et on continue...

VERSION 2. Mêmes règles que la version 1 excepté le fait que lorsqu'un joueur prononce un nombre N qui est le même que le précédent, le score potentiel est multiplié par lui-même. Par exemple, si le deux nombres sont 4 et 2, les joueurs gagnent 4 et 2. Au tour suivant, si les deux nombres sont 4 et 1, le premier joueur gagne 4×4=16 et le second joueur gagne 1. Aux tours suivants, le gain est de nouveau multiplié en cas de nouvelle répétition ; donc si le joueur 1 dit de nouveau 4, il peut gagner 16×4=64 ! Sauf si l'adversaire prononce 3, auquel cas ce dernier gagne la somme des deux, donc 3+64 = 67 !!! La répétition d'un nombre permet de gagner beaucoup mais elle peut aussi faire perdre beaucoup.

Implémentation en Prolog

Chaque équipe doit avoir un nom unique, sans espaces, commençant par une minuscule et l'indiquer sur la page https://cloud.univ-grenoble-alpes.fr/s/RtwMtXAgdi7XAoM

Chaque équipe doit créer un prédicat `joue/3`.

- Le premier argument est le nom de l'équipe ;
- Le second argument est la liste des coups précédents, en commençant par le dernier. Un coup est une paire de chiffres entre 1 et 5. Vos coups précédents sont toujours en première position. Par exemple, si les 2 premiers coups ont été 1 pour vous et 4 pour l'adversaire, puis 3 pour vous et 2 pour l'adversaire, cette liste vaudra `[[3,2], [1,4]]`. Ce n'est pas à vous de gérer cette liste, elle va vous être donnée.
- Le troisième argument est votre coup, un entier entre 1 et 5.

Ce prédicat va être appelé par le gestionnaire du tournoi (moi) avec, dans l'ordre, 2 variables instanciées et une variable non instanciée que votre programme devra déterminer. Par exemple, je vais d'abord appeler votre prédicat avec cette requête : `joue(votreNom,[],C).` Puis, si vous renvoyez C=2 et que votre adversaire renvoie 4, je calculerai les scores et j'appellerai de nouveau votre prédicat : `joue(votreNom,[[2,4]],C).` Et ainsi de suite...

Voici par exemple un programme qui joue systématiquement au hasard :

```
joue(aleatwar,_,N):-random_between(1,5,N).
```

Voici un programme qui joue le premier coup au hasard puis ensuite le coup précédent de l'adversaire :

```
joue(titfortat,[],N):-random_between(1,5,N).
joue(titfortat,[[_,C]|_],C).
```

Voici un programme qui joue le premier coup au hasard puis répète son coup précédent avec une probabilité de 20% ou joue au hasard avec une probabilité de 80 % :

```
joue(gambler,[],N):-random_between(1,5,N).
joue(gambler,[[Prec,_]|_],C):-random(N),(N<0.2,C=Prec;random_between(1,5,C)).
```

Évidemment, on peut faire mieux que ça et notamment piéger la stratégie précédente.

Si vous avez besoin de prédicats supplémentaires, ajoutez le nom de votre équipe à ces prédicats parce que je vais mettre tous les programmes dans le même fichier. Donc par exemple : `inverserListe_miashsForEver(…)` ou `stockeResultat_weLoveProlog(…).`

Si vous voulez stocker des faits avec `assert`, indiquez également le nom de votre équipe dans le noms de ces faits.

Je vous joins le programme qui gère le tournoi. Les trois stratégies précédentes sont déjà implémentées. Vous pouvez y ajouter votre programme et le tester contre les 3 autres en lançant un tournoi. Tout est décrit en commentaire dans le code.

Évaluation

L'évaluation du projet va dépendre de la pertinence de la votre stratégie et de son implémentation en Prolog. Je regarderai aussi le succès de votre stratégie par rapport aux autres joueurs, mais je ne sais pas du tout où cela va nous mener et s'il existe une stratégie gagnante. C'est une UE de Prolog donc l'implémentation est importante. On organisera une première version du tournoi courant avril.

Bonne chance !