

TP - 3

25 septembre 2014

Dans ce TP nous allons continuer d'écrire notre puissance 4. La prochaine étape consiste à coder une intelligence artificielle pour pouvoir jouer contre l'ordinateur.

Exercice - 1 *Terminer le puissance 4 Humain contre Humain*

Avant de continuer, vous devez terminer le [TP d'hier](#).

Conseil : A chaque nouvelle fonction que vous écrivez, compilez votre projet en utilisant le Makefile. N'attendez pas !

1 Intelligence artificielle

Nous allons implémenter un [algorithme minmax](#). Le principe est relativement simple. Imaginons que l'on veuille calculer un "bon coup" pour le joueur Rouge :

- nous définissons une fonction d'évaluation du jeu "anti-symétrique" pour Jaune et Rouge. C'est à dire qu'en échangeant les couleurs nous devons obtenir une évaluation opposée. Cette fonction doit refléter les chances de victoire de Rouge. Par exemple si le jeu contient 4 jetons rouges alignés, son évaluation doit être très grande. Pour 3 jetons avec une case vide au bout, un peu moins, etc..
- nous calculons maintenant TOUTES les possibilités de jeu possible pour un nombre de coups donné P . Nous obtenons alors un arbre de profondeur P pour lequel chaque noeud représente un état du jeu où Jaune ou Rouge doit jouer. La racine dénote l'état du jeu initial où Rouge doit jouer. Tous les fils d'un noeud correspondent aux différents coups possible pour joueur ;
- Nous allons maintenant évaluer le $MinMax(_)$ sur cette arbre. Pour une feuille, $MinMax$ coïncide avec la fonction d'évaluation. Pour un noeud où Rouge doit jouer, $MinMax$ calcule le maximum des $MinMax$ des fils (*i.e.*, Rouge essaye de maximiser ses gains). Pour un noeud où Jaune doit gagner, $MinMax$ calcule le minimum des $MinMax$ des fils. (*i.e.*, Jaune essaye de maximiser ses gains qui sont opposés à ceux de Rouge) ;
- le coup que l'on choisit de jouer pour Rouge est le coup qui mène au fils possédant le plus grand $MinMax$.

L'algorithme est exponentiel en P . Il va falloir ajuster cette valeur pour obtenir les coups de l'IA dans un temps raisonnable. De même, la fonction d'évaluation de jeu doit bien être ajustée.

Exercice - 2 *IA*

1- Commencez par vous assurez que vous comprenez bien le fonctionnement d'un tel algorithme. Sortez papier-stylo et réfléchissez aux différentes fonctions que vous devrez implémenter.

2- Le but est maintenant d'implémenter un tel algorithme dans un fichier `ia.ml`. La fonction principale de ce fichier doit être `meilleurCoup` de type `jeu -> int`.

Téléchargez le *fichier de signature* `ia.mli` [ici](#) qui indique au compilateur quelle est la fonction attendue qui doit être implémentée dans `ia.ml`. Placez ce fichier dans votre dossier de travail avec les autres fichiers et créez un fichier `ia.ml` qui contient une définition de `meilleurCoup` qui ne fait rien pour le moment (qui renvoie toujours 0). Complétez votre Makefile (il faut d'abord compiler le `ia.mli` avec `ocamlc -c regles.ml ia.mli` puis compiler `ia.ml` comme d'habitude).

Une fois que le Makefile vous permet de compiler `ia.ml`, implémentez l'algorithme que vous aviez en tête.

- 3- Ajoutez dans `puissance4.ml` la possibilité de jouer contre l'ordinateur en utilisant l'IA.
- 4- (Bonus) Si vous voulez grandement améliorer votre IA, vous pouvez optimiser le MinMax en implémentant un [élagage \$\alpha - \beta\$](#) .
- 5- (Bonus) Ecrivez une fonction qui fait jouer une IA contre une autre. Ecrivez une fonction qui réalise un tournoi entre IAs.

Les règles de la compétition On essayera d'organiser un tournoi entre vos différents IA. Vous devez donc bien respecter le nom de la fonction `meilleurCoup` et son type. La seule dépendance autorisée est `regles.ml` comme indiquée dans `ia.mli`. Vous voyez ici l'utilité de `ia.mli` qui agit comme un contrat : on sait déjà que votre code satisfait ces contraintes car sinon la compilation échouerait.

Avant de me l'envoyer, vous renommerez votre fichier `ia.ml` en `iaXY.ml` où `XY` sont les deux premières lettres de votre nom de famille (toutes les deux en majuscule).

Votre fonction `meilleurCoup` doit répondre en moins de 3 secondes (sinon, une colonne arbitraire est choisie à votre place). Si on a le temps, on fera jouer toutes les paires d'IAs. On classera les IAs par nombre de victoires.

Deadline Ce projet (version Humain contre Humain + version Humain contre Ordinateur) sera à rendre. Je vous donnerai la deadline en fin de TP.