Beuque Eric
Etienne Christophe

Master 1 info

Rapport projet de SCS-MOIA
SIAM

Partie MOIA

table de matière

Idée générale:

Le but du programme Prolog est de calculer le prochain coup à jouer pour un joueur (RHINO ou ELEPH) à partir d'un état du plateau de jeu. C'est le prédicat *coup_suivant(TAB, JOUEUR, COUP)* qui est donc le prédicat principal qui est appelé à chaque tour de jeu. Dans notre implémentation, le calcul du coup suivant se fait en 3 étapes : on récupère d'abord la liste des pions du joueur dans une liste (ceux sur le plateau, plus un parmis ceux hors jeu), ensuite on calcule tous les coups possibles pour chacun de ses pions. Pour finir, on attribue une note à chaque coup et on en garde le meilleur.
Nous avons choisi de modéliser la case en haut à gauche comme étant la case [1,1].

Pour la suite, nous pourrons prendre dans l'exemple :

```
EXTAB = [[  0  ,  m  ,  0  ,  0   ,  0  ],
         [  0  ,[r,h],  0  ,  0   ,  0  ],
         [  0  ,[e,d],  m  ,  m   ,  0  ],
         [  0  ,[e,b],  0  ,[e,b],  0  ],
         [  0  ,[r,h],  0  ,  0   ,[r,b]]]
```

Définition des prédicats de base :

case(TAB, LC, C)/3
TAB : le plateau de jeu
LC : coordonnée de la case
C : contenu de la case aux coordonnées LC dans TAB

Renvoie YES, si le contenu de la case de coordonnée LC par rapport au plateau TAB correspond a C.
Exemple :
case(EXTAB, [3,3], m). YES
case(EXTAB, [2,2], C). C = [r,h]

case_valide(LC)/1
LC : coordonnée de la case
Renvoie YES, si les coordonnées LC représente une case valide dans le plateau.
Exemple :
case_valide([2,2]). YES
case_valide([6,1]). NO

case_suivante(LC, ORIEN, ?NLC)/3
LC : coordonnée de la case d'origine
ORIEN : orientation dans laquelle calculer la case suivante
NLC : coordonnée de la nouvelle case
Calcule les coordonnées de la case suivante NLC, à partir de la case [L, C], dans le sens ORIEN.
Exemple :
case_suivante([2,2], h, NLC). NLC = [1,2]
case_suivante([2,2], d, NLC). NLC = [2,3]

bordure(LC)/1
LC : coordonnée de la case
Renvoie YES si la case de coordonnée LC est située en bordure du plateau.
Exemple :
bordure([1,4]). YES
bordure([2,2]). NO

orien_oppose(ORIEN, OORIEN)/2
ORIEN : une orientaion
OORIEN : l'orientation opposée de ORIEN
Renvoie YES si l'orientation ORIEN, est l'opposée de l'orientation OORIEN.
Exemple :
orien_oppose(h, b). YES
orien_oppose(h, OORIEN). OORIEN = b

depl_orien(LCD, LCA, ORIEN)/3
LCD : coordonnée de la case de départ
LCA : coordonnée de la case d'arrivée
ORIEN : orientation pour aller de LCD a LCA
Renvoie YES si pour rejoindre la case LCA depuis la case LCD nécessite un déplacement dans le sens ORIEN.
Exemple :
depl_orien([1,1],[2,5],b). YES
depl_orien([1,1],[2,5],ORIEN). ORIEN = b

dernier_pion(TAB, LC, ORIEN, ?NLC)/4
TAB : le plateau de jeu
LC : coordonnée de la case d'origine
ORIEN : orientation de recherche
NLC : coordonnée de la case contenant le dernier pion

Renvoie les coordonnées de la derniere case LNC ayant un pion en parcourant les cases dans le sens ORIEN, à partir de la case LC sur le plateau TAB.
Exemple :
dernier_pion(EXTAB, [4,2], h, NLC). NLC = [1,2]
dernier_pion(EXTAB, [4,2], b, NLC). NLC = [5,2]

premiere_montagne(TAB, CASE, ORIEN, ?MCASE)/4
TAB : le plateau de jeu
CASE : coordonnée de la case d'origine
ORIEN : orientation de recherche
MCASE : coordonnée de la case contenant la première montagne
Renvoie la case de la première montagne MCASE dans la liste de pion du plateau TAB à partir de la case CASE dans le sens ORIEN.
Exemple :
premiere_montagne(EXTAB, [3,2], d, MCASE). MCASE = [3,3]

derniere_montagne(TAB, CASE, ORIEN, ?MCASE)/4
TAB : le plateau de jeu
CASE : coordonnée de la case d'origine
ORIEN : orientation de recherche
MCASE : coordonnée de la case contenant la dernière montagne
Renvoie la case de la première montagne MCASE dans la liste de pion du plateau TAB à partir de la case CASE dans le sens ORIEN.
Exemple :
derniere_montagne(EXTAB, [3,2], d, MCASE). MCASE = [3,4]

Définition des prédicats pour récuperer la liste des pions :

Le prédicat qui permet de récupérer la liste des pions est le prédicat liste_pion/3.

liste_pion(TAB, JOUEUR, ?LPION)/3
Récupère la liste des pions dans LPION du joueur JOUEUR avec leur position sur le plateau TAB.
Appelle pour cela les prédicats :
liste_pion_jeu/4 pour récupérer la liste des pions dans le jeu en parcourant le plateau depuis la case [1,1]
liste_pion_horsjeu/3 pour compléter la liste des pions dans le jeu avec un pion hors jeu si le nombre de pion en jeu est inférieur à 5.
Un pion est récupéré sous la forme [JOUEUR, CASE, ORIEN] :
- JOUEUR : "e" pour éléphant, "r" pour rhino
- CASE : coordonnées de la case
- ORIEN : orientation du pion (h=haut, b=bas, g=gauche, d=droite, null=pas d'orientation)
Exemple :
liste_pion(EXTAB, r, LPION). LPION = [[r,[null,null],null], [r,[2,2],h], [r,[5,2],h], [r,[5,5],b]]

liste_pion_jeu(TAB, CASE, JOUEUR, ?LPION)/4
Ajoute un pion dans LPION, si le pion contenu dans la case CASE du plateau TAB est un pion du joueur JOUEUR. Puis continue de parcourir le tableau jusqu'à la fin.

liste_pion_horsjeu(LSPION, JOUEUR, LPION)/4

Renvoie la liste des pions LSPION dans LPION avec un pion du JOUEUR qui n'est pas sur le plateau si le nombre de pion du joueur JOUEUR dans LSPION est inferieur à 5.

Définition des prédicats pour générer les coups possibles :

liste_coups_possible_case_pions(TAB, LC, LPION, LCOUP, LNCOUP)/5
Renvoie dans LNCOUP, la liste des coups déjà calculée LCOUP, complétée par la liste des coups possibles sur le plateau TAB pour chaque pion de LPION qui amène le pion dans la case LC.
Ce prédicat appelle ainsi pour chaque pion le prédicat liste_coups_possible_case_pion/5 qui calcule pour chaque pion la liste des coups possibles par rapport à la case LC.
Ce dernier dispose alors ensuite de prédicats pour chacun des coups possibles :
coup_entree/5 : pour récupérer la liste des coups d'entrée qui amène le pion dans LC.
coup_deplace/5 : pour récupérer la liste des coups de déplacement qui amène le pion dans LC.
coup_change/4 : pour récupérer la liste des coups de changement d'orientation qui amène le pion dans LC.
coup_sortie/4 : pour récupérer la liste des coups de sortie qui sort le pion dans LC.
coup_entreepoussee/5 : pour récupérer la liste des coups d'entrée poussée qui amène le pion dans LC.
coup_poussee/5 : pour récupérer la liste des coups de poussée qui amène le pion dans LC.
Les types de coups sont validés par le prédicat valide_coup_poussee/4 qui fait appel à force/4.

force(TAB, LC, ORIEN, F)/4
Renvoie la force F positive cumulée des pions, calculée à partir du plateau TAB, en partant de la case [L,C] suivant l'orientation ORIEN. Si F = 0, alors la poussée n'est pas possible dans le sens ORIEN.
Exemple :
force(EXTAB, [3,3], d, F). F = 0 (pousser les 2 montagnes n'est pas possible).
force(EXTAB, [1,2], h, F). F = 1 (pousser la montagne est possible).

Définition des prédicats pour l'heuristique :

L'heuristique du projet mis en place, consiste à attribuer une note pour chacun des coups générés. meilleur_coup/3 parcours la liste des coups grâce à parcours_lcoup/6 qui pour chaque coup parcouru lui attribue une note avec note_coup/3, et le compare avec l'actuel meilleur coup mémorisé. Si le nouveau coup à une meilleur note, alors il est gardé.
La notation d'un coup fonctionne ainsi (en l'état actuel des choses, mais risque d'évoluer pour le tournoi):
0 : le coup est un coup perdant
1 : le coup est un coup gagnant
2 : le coup fait pousser une montagne
3 : le coup met le pion à côté d'une montagne
90 : coup entrée
91 : poussée
92 : déplace
93 : change
94 : entréepoussée
95 : sortie
Ainsi hormis pour un coup perdant, plus la valeur est petite meilleur est le coup.

La fonction note_coup/3 teste tous les cas de figure en commencant par chercher si le coup est perdant. Si ce n'est pas le cas, on s'arrête dès qu'on entre dans l'un des cas de figure. Ainsi on cherche toujours d'abord la meilleure note pour un coup. Les prédicats qui servent à cela sont :

fin_partie/3 : pour voir si la partie va se finir
fin_partie_gagnante/5 : pour voir si la fin de partie va faire gagner ou perdre
deplace_montagne/3 : pour voir si on va déplacer une montagne
rapproche_montagne/3 : pour voir si on va se rapprocher d'une montagne
cas_defaut/3 : pour les autres cas de figure

Définition des prédicats finaux :

coup_suivant(TAB, JOUEUR, COUP) /3
Calcule le meilleur coup suivant à partir d'un état TAB pour le joueur JOUEUR (e ou r) et le met dans COUP.
Génère la liste des coups possibles pour tous les pions du joueur, et parcours la liste des coups avec le prédicat meilleur_coup/3 qui renvoie le meilleur coup parmi cet liste selon l'heuristique mis en place.