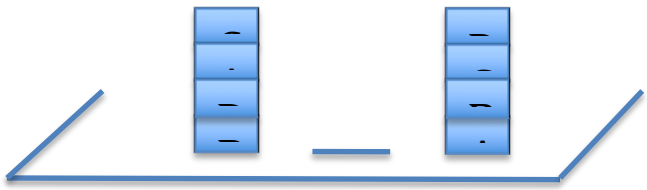
Sujet TP
Exemple de planification (exemple jouet) multi-agents
Le monde des blocs

Description du Problème

Soient 4 blocs A, B, C, D posés sur une table dans une situation initiale (exemple : de bas en haut B,D,A,C) et que l'on souhaite les faire évoluer vers une situation finale (exemple de bas en haut A,B,C,D). On souhaite écrire un programme automatique qui permet de réaliser le passage de la situation initiale à la situation finale, sachant qu'il n'existe sur la table que 3 positions pour le dépôt des blocs.



Dans une modélisation multi-agents, on considère que chaque bloc est un agent et que les agents à l'aide de perceptions locales et d'actions locales, et d'interaction/communication entre eux, réussissent à élaborer un plan de passage de la situation initiale à la situation finale.

Nous proposons de considérer des agents réactifs, plus précisément ce que l'on appelle des agents tropiques (eco-agent) dont la caractéristique est d'être guidés dans leur action par la réalisation d'une fonction de satisfaction ou l'augmentation de la valeur de cette fonction. Dans la programmation multi-agents, le but global du système est traduit en terme de buts locaux à atteindre par chacun des agents. Les perceptions sont locales et permettent d'avoir des informations sur le monde, permettant la prise de décision. Les actions sont locales et sont assez élémentaires comme se déplacer, pousser, etc.

1ere partie (modélisation ad hoc)

Programmer dans le langage de votre choix, le système multi-agents permettant de résoudre ce problème de planification par éco-résolution. Pour cela on considère que chaque agent a comme but, l'emplacement où il doit être dans la situation finale et qu'il peut exécuter des actions : Move (X) et Push() selon les 2 règles de comportements suivants

- *Move(X)* : permet de se déplacer vers l'emplacement X (un des 2 emplacements possibles sur la table), dans le cas où il n'est pas satisfait de sa position et qu'il est libre (aucun bloc au dessus de lui). Le choix de X, se fait de manière probabiliste en suivant une stratégie donnée, dont le choix aléatoire.
- *Push()* : qui consiste à exercer une poussée sur le bloc se trouvant au dessus de lui, s'il n'est pas satisfait et qu'il doit se libérer pour se déplacer.

Dans cette première partie, on ne se préoccupe pas de la coordination des actions des agents et de la non redondance des actions individuelles (possibilité de bouclage).

Avant toute programmation, il convient de définir clairement le modèle multi-agents à implémenter :

- Agent : Architecture interne (Agent réflexe simple ou avec modèle, Agent à but, agent à utilité, etc.), les perceptions, les actions, les règles de comportements.
- Environnement : que représenter ? sa dynamique éventuelle ?
- Interactions : Agent-Agent ? Agent-Environnement ? Communication via un langage ?
- Organisation : organisation a priori ? émergente ? Pas d'organisation ?

2 ème partie (Modélisation par EPS)

- Rependre la programmation en utilisant le framewok : Eco Problem Solving -EPS (vu en cours) où les agents sont définis par des automate à états finis, enchainant 3 comportements simples : satisfaction, agression, fuite. (cf. Cours pour plus de détails)

Avant programmation, poser clairement le modèle en identifiant les différentes composantes Agent, Environnement, Interaction, Organisation.