

## Cours Planification d'actions

### Séance 2 – Travaux pratiques

**Exercice 1 :** récupérer le planificateur d'actions CPT version 2 (exécutable « cpt.exe » compilé sous Windows), le fichier « domain-blocksaips.pddl » (le domaine de planification pour le monde des cubes) et le fichier « blocksaips01.pddl » (un problème de planification dans ce domaine).

Que signifient les quatre opérateurs du fichier de domaine ?

Quelle est la différence entre l'opérateur « put-down » et l'opérateur « stack » ? Pourquoi faut-il dissocier ces deux cas ?

Que veut dire le fluent « (holding ?x) » ? A quoi sert-il ? ( Si ce fluent n'était pas là, comment faudrait-il changer l'écriture des opérateurs pour représenter le monde des cubes ? )

**Exercice 2 :** lancer le planificateur sur ce problème avec ce domaine.

Quelle est la longueur du plan-solution ? En combien de temps est-il trouvé ? Combien y a-t-il eu d'itération ? Combien de temps dure chaque action du plan-solution ?

Quels plans-solutions plus longs peut-on imaginer pour résoudre ce problème de cubes ? Pourquoi ne sont-ils pas fournis par ce planificateur ?

**Exercice 3 :** écrire le problème de planification suivant (un fichier PDDL) pour ce domaine des cubes :

- Etat initial : une tour de cubes B / C / A / D / table et la pince est vide.
- Buts : une tour de cubes D / C / A / B.

Quelle est la longueur du plan-solution ? En combien de temps est-il trouvé ? Combien y a-t-il eu d'itérations ?

**Exercice 4 :** écrire le problème de planification suivant (un fichier PDDL) pour ce domaine des cubes :

- Etat initial : une tour de cubes C / G / E / I / J / A / B / table , une deuxième tour de cubes F / D / H / table et la pince est vide.
- Buts : une tour de cubes C / B / D / F / I / A / E / H / G / J.

Quelle est la longueur du plan-solution ? En combien de temps est-il trouvé ? Combien y a-t-il eu d'itérations ?

(Si le timer de CPT se déclenche sur votre machine, augmenter à 60 secondes le temps alloué avec l'option « -t » )

**Exercice 5 : planification de chemin.** On dispose d'un graphe acyclique orienté de nœuds, et on souhaite trouver un chemin pour aller d'un nœud à un autre. Ecrire un opérateur d'un domaine PDDL qui, si l'agent est sur un nœud **from** et qu'il y a un arc pour aller du nœud **from** au nœud **to**, transporte l'agent de ce nœud **from** vers ce nœud **to**. Ecrire un problème PDDL sur un petit graphe, et tester pour calculer un chemin de quelques nœuds.

A votre avis, cette méthode de planification de chemin est-elle efficace ? Pourquoi ?

**Exercice 6 : le singe et les bananes.** Un singe est dans une salle d'expérience avec des bananes accrochées au plafond hors d'atteinte. Dans cette salle il y a une caisse, sur laquelle le singe peut monter. Il peut aussi la pousser. Le singe et la caisse ont tous deux la hauteur **Bas**, mais si le singe monte sur la caisse, sa hauteur sera **Haut**, la même que les bananes. Au début, le singe est en **A**, la caisse en **B** et les bananes en **C**. Les scientifiques cherchent à déterminer si un singe est capable de faire le raisonnement nécessaire pour pouvoir attraper les bananes.

- Ecrire l'état initial et les buts de ce problème de planification « singe-bananes01 ».
- Ecrire les six opérateurs, formant le domaine de planification « singe » : **Aller** à un endroit à partir d'un autre, **Pousser** un objet d'un endroit à un autre, **Monter** sur un objet, **Descendre** d'un objet, **Attraper** un objet (si le singe et l'objet sont au même endroit et à la même hauteur, le singe peut attraper l'objet) et **Lâcher** un objet.

Quelle est la plan-solution ? En combien de temps est-il trouvé ? Combien y a-t-il eu d'itérations ?

Votre écriture de l'opérateur **Pousser** est probablement incomplète, parce que si l'objet à pousser est trop lourd, l'opérateur **Pousser** n'aura aucun effet. Est-ce un exemple du problème de la qualification ou de la ramification ?

**Exercice 7 : les tours de Hanoi.** Ce jeu (voir figure 1) consiste à déplacer des disques de diamètres différents d'une tour, ou pic, de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », et ceci en un minimum de coups, tout en respectant les règles suivantes :

1. on ne peut déplacer plus d'un disque à la fois ;
2. on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

On suppose que cette dernière règle est également respectée dans la configuration de départ.

On demande :

1. Ecrire en PDDL l'état initial pour une tour à 1, 2, 3, 4 et 5 disques, initialement sur le pic1 à déplacer sur le pic3 avec le pic2 comme intermédiaire (les pics sont numérotés de gauche à droite).
2. Ecrire en PDDL les 4 opérateurs pour le domaine des tours de Hanoi : enlever avec la pince le seul disque d'un pic ; mettre avec la pince le premier disque sur un pic vide ; mettre le disque que tient la pince sur un disque plus grand et libre ; enlever avec la pince un disque libre du disque du dessous. NB : on pourra modéliser un pic par un objet implicitement de grande surface et sur lequel on peut mettre un disque, mais qui n'est pas lui-même un disque « disk » mais un pic « peg ».
3. Générer un plan d'actions avec CPT pour résoudre les tours de Hanoi pour 1 à 4 disques (au besoin, ajuster le timer de CPT avec l'option « -t »). Quelle est la longueur de chaque plan-solution ? En combien de temps est-il trouvé ? Combien y a-t-il eu d'itérations ? Que se passe-t-il pour résoudre les tours de Hanoi pour 5 disques ?
4. Que fait la fonction suivante, écrite en pseudo code :

```

PROCEDURE Hanoi(NDisques, PicDepart, PicIntermediaire, PicArrivee)
  SI NDisques different de 0 ALORS
    Hanoi(NDisques – 1, PicDepart, PicArrivee, PicIntermediaire) ;
    Afficher(“Déplacer le disque numéro “ + NDisques + « du pic » +
      PicDepart + « au pic » + PicArrivee + « \n »);
    Hanoi(NDisques – 1, PicIntermediaire, PicDepart, PicArrivee);
  FINSI
FINPROCEDURE

```

Quelle est la complexité algorithmique de cette fonction en nombre de mouvements ? Cela correspond-il à ce qui est trouvé dans la question 3 ? Pourquoi ?

Quelle est la différence entre la résolution du problème des tours de Hanoi avec cette fonction et avec un planificateur d’actions comme CPT ?

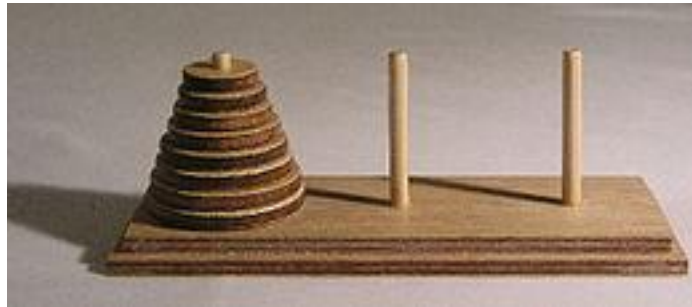


Figure 1 : exemple d’une tour de Hanoi avec 8 disques.