

Attention

- Dans toutes les questions suivantes, la description des actions obéit aux conventions suivantes : en partie gauche de la flèche, les préconditions ; en partie droite, les ajouts (préfixés par +) et les retraits (préfixés par -).
- Dans tous les algorithmes qui suivent, les actions devront être considérées dans l'ordre de leur écriture (de gauche à droite et de haut en bas) ; les préconditions, ajouts et retraits seront également considérés dans cet ordre d'écriture.

Nous allons résoudre un problème de planification simple, dans le cadre de la logique propositionnelle, en utilisant deux des algorithmes du cours. Le problème à résoudre est le suivant :

Etat initial = {a,b,c}

But = {x,y}

Actions :

A : b \rightarrow +d,-a

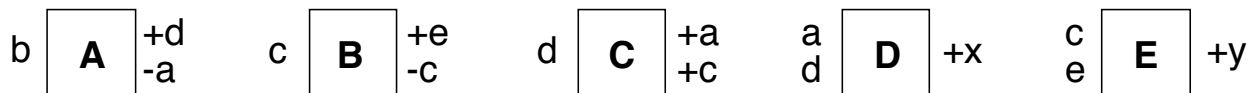
B : c \rightarrow +e,-c

C : d \rightarrow +a,+c

D : a,d \rightarrow +x

E : c,e \rightarrow +y

Représentation graphique des actions :



1. Le planificateur GRAPHPLAN

1. (15 mn.) Résoudre le problème précédent en utilisant l'**algorithme de GRAPHPLAN**. Vous développerez soigneusement le graphe de planification en distinguant les différents niveaux et les mutex entre actions et entre fluents pour chacun de ces niveaux. Vous mettrez en évidence dans ce graphe le plan-solution trouvé en utilisant l'heuristique « Noop-first » et vous l'expliciterez ci-dessous. Vous donnerez également les plans séquentiels qui lui sont associés.
2. (3 mn.) Donnez la définition de l'heuristique d'extraction « level-based ». En l'utilisant sur le graphe de planification précédent à la place de l'heuristique « Noop-first » obtient-on un plan-solution différent ? Si oui, lequel ?
3. (2 mn.) Donner un exemple de mutex qui ne disparaît pas pendant la construction du graphe de planification. Donner un exemple de mutex qui disparaît pendant la construction du graphe de planification. Expliquez cette différence de comportement.
4. (10 mn.) En supposant que l'action D définie précédemment soit maintenant **remplacée** dans le problème de la question 1 par l'action **D' : a,d \rightarrow +x,-y**, décrire clairement le comportement du planificateur GRAPHPLAN à partir du niveau 3 du graphe de planification.
5. (15 mn.) Résoudre maintenant ce problème en utilisant maintenant la relation d'autorisation et le planificateur LCGP.

2. Recherche dans les espaces de plans partiels

(15 mn.) Résoudre le problème précédant en utilisant un algorithme de **recherche dans les espaces de plans partiels en profondeur d'abord**. Les opérations de modification de plans considérées seront uniquement celles du critère de vérité de PWEAK. L'action de début et fin seront respectivement nommées **Init** et **Fin**. Les actions seront considérées **dans l'ordre alphabétique**. Les préconditions, ajouts et retraits seront considérés **dans l'ordre de leur écriture dans l'énoncé**. La liste des buts sera gérée comme une **pile**. Pour le choix d'un établisseur, on préférera un établisseur qui est déjà présent dans le plan partiel à l'insertion d'un nouvel établisseur. Pour la résolution de conflits, on préférera la promotion à la démotion (ou rétrogradation). Vous développerez l'arbre de recherche en partant du plan partiel initial jusqu'à obtenir **un premier plan-solution** que vous explicitez ci-dessous. Vous donnerez également les plans-solutions séquentiels qui correspondent au plan trouvé. Comparer ce résultat avec celui de la question 1.1.

3. Codage du graphe de planification en CSP

(15 mn.) Expliquez clairement, et à l'aide de l'exemple ci-dessous, le principe du codage d'un graphe de planification en CSP. Donnez ensuite le graphe de planification correspondant à cet exemple.

Variables = {f1, f2, f3, f4, f5, f6, f7, f8}

Domaines :

$D(f1) = \{A1\}$

$D(f2) = \{A2, A3, \perp\}$

$D(f3) = \{A4, A5, A6, \perp\}$

$D(f4) = \{A7, \perp\}$

$D(f5) = \{A8, \perp\}$

$D(f6) = \{A9, \perp\}$

$D(f7) = \{A10, \perp\}$

$D(f8) = \{\text{Vrai}, \perp\}$

Contraintes FM :

$\neg ((f4 \neq \perp) \wedge (f7 \neq \perp))$

$\neg ((f5 \neq \perp) \wedge (f7 \neq \perp))$

$\neg ((f6 \neq \perp) \wedge (f7 \neq \perp))$

Contraintes AM :

$(f2=A2) \rightarrow (f3 \neq A6)$

$(f2=A3) \rightarrow (f3 \neq A4)$

$(f2=A3) \rightarrow (f3 \neq A5)$

$(f2=A3) \rightarrow (f3 \neq A6)$

$(f4=A7) \rightarrow (f7 \neq A10)$

$(f5=A8) \rightarrow (f7 \neq A10)$

$(f6=A9) \rightarrow (f7 \neq A10)$

Contraintes A :

$(f1=A1) \rightarrow (f2 \neq \perp) \wedge (f3 \neq \perp)$

$(f2=A2) \rightarrow (f4 \neq \perp)$

$(f2=A3) \rightarrow (f5 \neq \perp)$

$(f3=A4) \rightarrow (f5 \neq \perp)$

$(f3=A5) \rightarrow (f4 \neq \perp) \wedge (f6 \neq \perp)$

$(f3=A6) \rightarrow (f7 \neq \perp)$

$(f4=A7) \rightarrow (f8 \neq \perp)$

$(f5=A8) \rightarrow (f8 \neq \perp)$

$(f6=A9) \rightarrow (f8 \neq \perp)$

$(f7=A10) \rightarrow (f8 \neq \perp)$

4. Heuristiques pour la recherche heuristique dans les espaces d'états

(15 mn.) Pour le problème de planification ci-dessous, construisez le graphe de planification du problème relaxé (par suppression des retraits des actions) puis calculez la valeur de chacune des 3 heuristiques suivantes : Sum, Max et FF (avec choix « Noop d'abord, sinon actions par ordre alphabétique »). Comparer et commentez ces résultats.

Etat initial = {a,b,c,d}

But = {x,y}

