Landmarks - Bibiliograhie

DCSD/CD

Simon Vernhes

Onera

17 *novembre* 2011

Sommaire

1 Définitions
- Landmark
- Ordres
- Complexité

2 Algorithmes

Définitions

1 Définitions
- Landmark
- Ordres
- Complexité

2 Algorithmes

Landmark

Definition (Landmark)

Soit (A, I, G) une tâche de planification. Un fluent L est un landmark si $\forall P = \langle a_1, \ \dots \ , a_n \rangle \in A^*, G \subset Result(I, P)$ $(\exists i \in \{1, \ \dots \ , n\}) \ \ L \in Result(I, \langle a_1, \dots, a_i \rangle))$

C'est-à-dire

L appartient à au moins un état de tous les plans permettant d'aller de l'état initial I à l'état but G.

Necessary Order

Definition ($L \rightarrow_n L'$)

Soient (A, I, G) une tâche de planification et deux landmarks L et L'. On dit qu'il existe un necessary order entre L et L' ($L \rightarrow_n L'$) si et seulement si
$L' \notin I$ et $\forall P = \langle a_1, \ \ldots \ , a_n \rangle \in A^*, L' \in Result(I, P)$
$L \in Result(I, \langle a_1, \ \ldots \ , a_{n-1} \rangle)$

C'est-à-dire

Pour toutes les séquences d'actions qui amènent à un état contenant L', l'avant dernier état contient nécessairement L.

ONERA
THE FRENCH AEROSPACE LAB

Greedy Necessary

Definition ($L \rightarrow_{gn} L'$)

Soient (A, I, G) une tâche de planification et deux landmarks L et L'. On dit qu'il existe un greedy necessary order entre L et L' ($L \rightarrow_{gn} L'$) si et seulement si
$L' \notin I$ et $\forall P = \langle a_1, \ldots, a_n \rangle \in A^*, L' \in Result(I, P)$ et
$(\forall i \in \{1, \ldots, n-1\}) L' \notin Result(I, \langle a_1, \ldots, a_i \rangle)$
$L \in Result(I, \langle a_1, \ldots, a_{n-1} \rangle)$

C'est-à-dire

Pour toutes les séquences d'actions qui amènent pour la première fois à un état contenant L', l'avant dernier état contient nécessairement L.

Reasonable order

Definition ($S_{(L',\neg L)}$)

$S_{(L',\neg L)}$ est l'ensemble de tout les états où L' vient d'être ajouté mais où L n'est jamais encore apparu.

Definition (Aftermath)

L' est un effet/une suite (in the aftermath) de L si depuis tous les $s \in S_{(L',\neg L)}$, L devient vrai dans tous les plans solutions $P \in A^*$ tel que $G \in Result(P, s)$.

Definition ($L \rightarrow_r L'$)

On dit que $L \rightarrow_r L'$ si

$$\begin{cases} L' \text{ est un effet de } L \\ (\forall s \in S_{(L',\neg L)})(\forall P \in A^* \text{ achevant } L)P \text{ supprime nécessairement } L' \end{cases}$$

Reasonable order

Obedient Reasonable Orders

Example

Si on a $L' \rightarrow_n L''$, $L \rightarrow_r L''$ et que les seules actions permettant de produire L détruisent L'.

TODO

Complexité

Décidabilité

- Landmark – PSPACE-complete
- $\rightarrow_n$ – PSPACE-complete
- $\rightarrow_{gn}$ – PSPACE-complete
- $\rightarrow_r$ – PSPACE-complete

Algorithmes

1 Définitions

2 Algorithmes

Landmark

En utilisant un graphe de planification relaxé (RPG), on construit un graphe de génération de landmark. On part des buts, et on remonte.

Algorithm 1: Landmark Generation Graph

input : (A, I, G) a planning task
output: LGG = (N, A) où N est l'ensemble des noeuds, et A les arcs de l'arbre

$LGG \leftarrow (G, \emptyset)$;
$C \leftarrow G$;
while $C \neq \emptyset$ **do**
    $C' \leftarrow \emptyset$;
    for $(L' \in C) level(L') \neq \emptyset$ **do**
        $A \leftarrow \{(a | \forall a \in A) L' \in del(a)$ et $level(a) = level(L) - 1i\}$;