

## Version C du planificateur TLPlan

Voici une courte documentation expliquant comment planifier un problème du domaine du monde des blocs (*blocks world*) avec la version programmée en langage C du planificateur TLPlan.

### 1. Éléments non-fonctionnels

Voici premièrement les choses qui ne fonctionnent pas avec le planificateur.

1. Deux stratégies de recherche, dont celle par défaut, causent des erreurs de segmentation dans TLPlan. Ces stratégies non-fonctionnelles sont *depth-first* et *depth-first-priority*. Toutes les autres stratégies de recherche disponibles dans TLPlan fonctionnent sans problème.
2. Deux fonctions utilitaires du monde des blocs qui génèrent aléatoirement des états initiaux et des buts (*random-blocks-world* et *random-blocks-goal*, respectivement) causent des erreurs de segmentation quand le code est optimisé lors de la compilation avec l'option `-O3`.

Malgré cela, le planificateur TLPlan peut être compilé et exécuté sous Windows sans devoir recourir à Cygwin. Le compilateur `gcc` de GNU a été utilisé (un `Makefile` ainsi qu'un fichier projet pour l'environnement de développement Eclipse seront fournis avec le code source), mais les autres compilateurs fonctionnant sous Windows devraient aussi pouvoir compiler TLPlan sans problème.

### 2. Étapes de la planification d'un problème

Voici maintenant comment planifier un problème dans TLPlan. Nous prendrons comme exemple le domaine du monde des blocs.

Il faut d'abord lancer TLPlan, par exemple à partir de la ligne de commande. L'exécutable doit être placé dans le même répertoire que la librairie libtl.dll sous Windows, ou libtl.so sous Linux/Unix.

Une fois le logiciel lancé, on doit charger le domaine du problème avec la commande suivante :

```
(load-file "LinearBlocksWorld.tlp")
```

Alternativement, on peut donner le nom du fichier à charger en argument lors du lancement de TLPlan à partir de la ligne de commande :

```
./tlplan LinearBlocksWorld.tlp
```

On peut aussi charger le fichier en définissant et en chargeant le domaine, à l'aide de ces deux fonctions de TLPlan :

```
(def-domain  
  Blocks                ; Nom du domaine  
  "Description du monde des blocs" ; Commentaire  
  "LinearBlocksWorld.tlp") ; Fichier à charger  
  
(load-domain Blocks)
```

Dans tous les cas, les fichiers LinearBlocksWorld.tlp et la librairie dynamique libblocksworld.dll doivent se trouver dans le même répertoire que l'exécutable.

On peut aussi charger un autre fichier, qui contient des définitions de buts et d'états initiaux du monde des blocks. On peut utiliser la commande `(load-file "LinearBlocksProblems.tlp")` pour y parvenir.

Ensuite, il faut définir un état initial et un but à atteindre. On doit d'abord définir le but en utilisant la commande `(set-goal ...)`. Cette commande s'attend à recevoir une série de prédicats et de fonctions, par exemple `(on a b)`, `(clear a)`, etc. Le fichier LinearBlocksProblems.tlp contient plusieurs buts que l'on peut utiliser directement de la façon suivante :

```
(set-goal (goal2))
```

On peut aussi utiliser une fonction utilitaire du monde des blocs qui génère aléatoirement un but :

```
(set-goal
  (random-blocks-goal
    1      ; Nombre utilisé pour générer un nombre aléatoire
    5))   ; Nombre de blocs différents
```

Une fois le but défini, on peut définir l'état initial de façon analogue à la définition du but :

```
(set-initial-world (state2))
; Définit dans LinearBlocksProblems.tlp
```

ou encore :

```
(set-initial-world
  (random-blocks-world
    3      ; Nombre utilisé pour générer un nombre aléatoire
    5))   ; Nombre de blocs différents
```

Finalement, la planification peut s'effectuer avec la commande `(plan)`. On peut choisir la stratégie de recherche grâce à la commande `(set-search-strategy ...)` avant de démarrer la planification. Tel que dit précédemment, `depth-first` et `depth-first-priority` causent des erreurs dans TLPlan. Il faut donc utiliser une des stratégies suivantes pour effectuer la planification :

```
(set-search-strategy depth-first-no-backtracking)
(set-search-strategy depth-best-first)
(set-search-strategy breadth-first-priority)
(set-search-strategy breadth-first)
(set-search-strategy best-first)
```

Le fichier `LinearBlocksWorld.tlp` sélectionne déjà la stratégie `depth-first-no-backtracking`; on peut donc utiliser la commande `(plan)` tout de suite après avoir défini l'état initial et le but à atteindre.

Le plan trouvé, s'il existe, sera écrit dans le fichier `tlplan.log`, se trouvant dans le même répertoire que l'exécutable. Il existe une commande

(print-plan-list ...) mais elle ne semble pas fonctionner correctement pour l'instant. On peut ensuite quitter TLPlan avec la commande (exit).

En résumé :

1. Charger le domaine du problème :

```
(load-file "LinearBlocksWorld.tlp")
```

2. Définir le but à atteindre :

```
(set-goal  
(on a b)  
(on b c)  
(ontable c))
```

3. Définir l'état initial :

```
(set-initial-world  
(ontable a)  
(on c b)  
(ontable b)  
(clear a)  
(clear c)  
(handempty))
```

4. Démarrer la planification :

```
(plan)
```

Avec ces instructions, on peut effectuer la planification d'un problème du monde des blocs. La planification d'un problème d'un autre domaine ne devrait avoir comme seule différence la liste des prédicats et des fonctions lors de la définition de l'état initial et du but.