

Segunda práctica de IA I, Curso 2007-08

Planificación

La segunda práctica de IA I consiste en la implementación en Java de un **planificador lineal con pila de objetivos**. El dominio de aplicación será una pequeña extensión del “mundo de los bloques”, definido de la forma siguiente:

Predicados:

- **Sobre-mesa**(x): x está situado sobre la mesa.
- **Sobre**(x,y): x está situado sobre y .
- **Libre**(x): x no tiene ningún objeto encima (puede estar cogido por el brazo del robot).
- **Brazo-libre**: el brazo del robot no tiene ningún objeto cogido.
- **Cogido**(x): el objeto x está cogido por el brazo del robot.
- **Num-cols-usadas**(n): se utilizan n columnas de bloques.

Operadores:

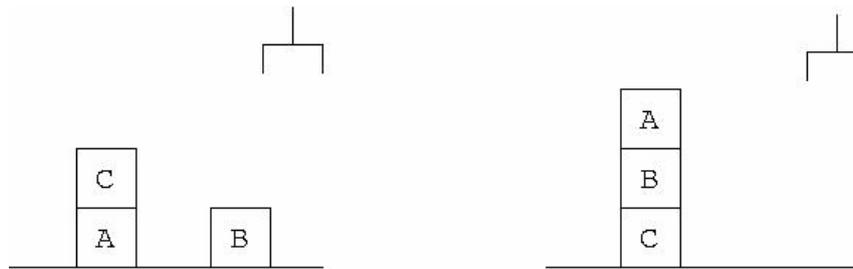
- **Coger**(x): sirve para que el robot coja bloques de la mesa.
 - *Precondiciones*: **Num-cols-usadas**(n), **Sobre-mesa**(x), **Brazo-libre**, **Libre**(x).
 - *Eliminar*: **Sobre-mesa**(x), **Brazo-libre**, **Num-cols-usadas**(n).
 - *Añadir*: **Cogido**(x), **Num-cols-usadas**($n-1$).
- **Dejar**(x): sirve para que el robot suelte bloques en la mesa
 - *Precondiciones*: **Cogido**(x), **Num-cols-usadas**(n), $n < 3$.
 - *Eliminar*: **Cogido**(x), **Num-cols-usadas**(n).
 - *Añadir*: **Sobre-mesa**(x), **Num-cols-usadas**($n+1$), **Brazo-libre**.
- **Desapilar**(x,y): sirve para que el robot pueda coger x de encima de y
 - *Precondiciones*: **Sobre**(x,y), **Libre**(x), **Brazo-libre**.
 - *Eliminar*: **Sobre**(x,y), **Brazo-libre**.
 - *Añadir*: **Cogido**(x), **Libre**(y).
- **Apilar**(x,y): sirve para que el robot pueda colocar x sobre y
 - *Precondiciones*: **Cogido**(x), **Libre**(y).
 - *Eliminar*: **Cogido**(x), **Libre**(y).
 - *Añadir*: **Sobre**(x,y), **Brazo-libre**.

Como se puede ver, se ha introducido una modificación en el dominio clásico del mundo de los bloques: en la mesa de nuestro dominio solo caben un máximo de 3 objetos. Esto se controla con el predicado *Num-cols-usadas*, que se actualiza cada vez que se utiliza el operador *Coger* o un *Dejar*. Notad que, en este dominio, en todos los casos es factible construir un plan que lleve desde un cierto estado inicial a uno final.

Consideraremos las dos situaciones siguientes:



Problema 1



Initial State

Goal State

Problema 2

Se pide:

- Implementar en Java un planificador lineal con pila de objetivos que resuelva el problema, indicando la representación interna que se ha utilizado para manejar precondiciones, comprobar la aplicabilidad de los operadores, instanciarlos, etc. Se valorará el diseño de heurísticas (específicas para este problema) que ayuden a la planificación durante el proceso de apilado de precondiciones, aplicación de operadores e instanciación.
- Realizar un extenso juego de pruebas con, al menos, las dos configuraciones propuestas, comentando la solución encontrada. Se deberá estudiar la influencia del orden de apilado de las precondiciones, el orden de evaluación de los operadores y el orden de instanciación.

Entregar el informe por escrito a David Sánchez y enviar por e-mail (a david.sanchez@urv.cat) el código de la práctica.

Condiciones de entrega:

- La práctica se hará de forma individual.
- Se hará una entrevista con el profesor de prácticas.
- Fechas de entrega:
 - Hasta el 21 de diciembre (nota máxima 10, 1ª conv.)
 - Hasta el 22 de enero (nota máxima 5, 2ª conv.)