

Inteligencia Artificial

Planeamiento: Strips

Diego Marcovecchio (LU: 83815) Tomás Touceda (LU: 84024)

24 de Noviembre de 2010

Índice general

1. Introducción	2
1.1. Descripción	2
1.2. Modularización	2
2. Estado del mundo	3
2.1. Relaciones entre bloques	3
2.2. Acciones	3
3. Estrategia del planeador	4
3.1. Decisiones de diseño	4
3.2. Algoritmo general	4
4. Implementación	5
4.1. Descripción de los predicados	5
4.2. Código	5

Capítulo 1

Introducción

1.1. Descripción

Este proyecto consiste en la implementación de un planificador STRIPS para un Mundo de Bloques similar al utilizado para los trabajos prácticos durante el cuatrimestre. El mundo posee un estado inicial fijo en el que todos los bloques se encuentran apoyados sobre la mesa. El objetivo del planificador es: dado un conjunto de metas, encontrar una secuencia de acciones que permita alcanzar todas ellas en un estado final.

1.2. Modularización

Dada la sencillez del planificador, a diferencia de los proyectos anteriores, toda la implementación del proyecto se encuentra en el archivo *strips.pl*. Más adelante en el informe se detallará cada uno de los predicados utilizados.

Capítulo 2

Estado del mundo

2.1. Relaciones entre bloques

El estado del mundo se describe mediante un conjunto de predicados que denotan relaciones entre bloques. Dichas relaciones son:

- *sobre*(A, B): indica que el bloque A está inmediatamente arriba del bloque B .
- *libre*(A): indica que el bloque A no tiene ningún bloque encima.
- *enMesa*(A): es verdadera cuando el bloque A está apoyada encima de la mesa (es decir, no hay ningún bloque *debajo* de A).

2.2. Acciones

Las únicas acciones que pueden realizarse en este mundo son las acciones de apilar, y desapilar bloques. Presentamos, a continuación, el formato de dichas acciones en la notación de STRIPS:

- *apilar*(A, B): coloca al bloque A inmediatamente arriba del bloque B

Precondiciones

[libre (A) , libre (B) , enMesa (A)]
--

Add_List

[sobre (A, B)]

Del_List

[libre (B) , enMesa (A)]

- *desapilar*(A, B): coloca el bloque A que está encima del bloque B sobre la mesa. Precondiciones

[sobre (A, B) , libre (A)]

Add_list

[enMesa (A) , libre (B)]

Del.list

[sobre (A, B)]

Capítulo 3

Estrategia del planeador

3.1. Decisiones de diseño

Es importante destacar que el algoritmo trabaja realcanzando las metas que figuran en la `delete_list` de cada acción, en lugar de ordenar las acciones para proteger las metas. Ésta decisión fue tomada debido a que no siempre es posible reordenar las metas para evitar los conflictos. A partir de esta decisión, también podemos concluir que el plan encontrado no necesariamente será minimal (pues para ello puede ser necesario realizar dicho reordenamiento).

3.2. Algoritmo general

El planeador está basado en la estrategia presentada en la sección «The STRIPS Planner» del capítulo 8 del libro *Computational Intelligence: a logical approach* (Poole, Mackworth & Goebel). Se corrigieron algunos errores del pseudocódigo allí presentado, y se pasó a utilizar un parámetro más en el predicado *achieve_all/4* para representar las metas que ya se cumplieron. De esta manera, el algoritmo general del planeador es:

- Obtener la primer meta de la lista de metas pendientes.
- Analizar el conjunto de acciones necesarias para alcanzar dicha meta.
- Observar cuáles de las metas alcanzadas previamente fueron deshechas por el segundo paso, y volver a agregarlas a la lista de metas pendientes.
- Repetir el algoritmo, quitando de la lista de metas pendientes a todas aquellas que hayan sido alcanzadas por la acción realizada.

Capítulo 4

Implementación

4.1. Descripción de los predicados

En esta sección mostraremos todos los predicados utilizados en la implementación, su signatura, y una descripción para cada uno de ellos.

PREDICADORRRRRRRRRRRRRRRRRRRRRRRRRRRRR

4.2. Código

Por último, detallaremos a continuación el código del planeador: