

Inteligencia Artificial



Algoritmos STRIPS

Elaborado por:

- Fausto León Amador Mairena
- Jason David Saavedra Córdoba

Carrera:

- Ing. Cibernética Electrónica

Email:

- fausto1mayo@gmail.com
- sjasondavid@yahoo.es

Representación de Planificación

Lógica De Predicados – Técnica STRIPS

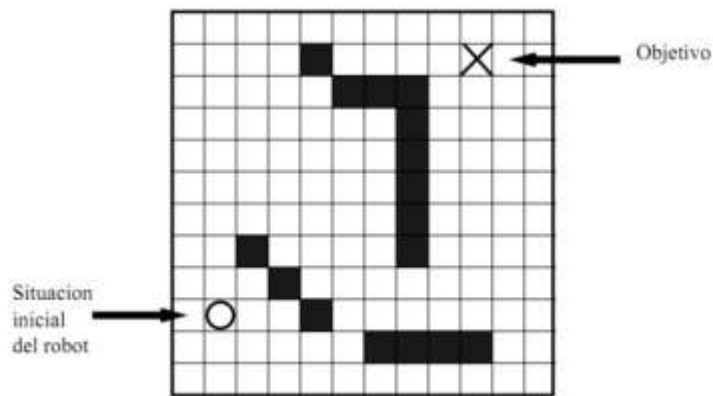
Operador STRIPS: representa una acción mediante 3 componentes:

precondiciones (PC): lista de átomos que tienen que ser ciertos en el estado para que la acción pueda ser aplicada

lista añadir (A): lista de átomos que pasan a ser ciertos una vez aplicada la acción

lista borrar (B): lista de átomos que dejan de ser ciertos una vez aplicada la acción

1. Un robot ha de desplazarse por una rejilla, desde una posición inicial a una final.
 - a. 8 movimientos posibles: N, S, E, O, NO, NE, SO, SE
 - b. En algunas de las rejillas existen obstáculos no franqueables



Lenguaje: números que indican coordenadas horizontales y verticales

Predicados: ROBOT_EN(.,.) y LIBRE(.,.)

Estado inicial (casillas sin obstáculos y posición del robot): LIBRE(1,1),..., LIBRE(6,2), LIBRE(11,2),..., LIBRE(12,12),

ROBOT_EN(2,3).

Objetivo: ROBOT_EN(10,11)

Modelar las acciones (operadores) a través de la técnica STRIPS.

- MOVER_SE(X,Y)
- MOVER_SO(X,Y)
- MOVER_NE(X,Y)
- MOVER_NO(X,Y)
- MOVER_EN(X,Y)
- MOVER_ES(X,Y)
- MOVER_ON(X,Y)
- MOVER_OS(X,Y)

Resolución:

MOVER_SE(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X+1,Y-1)
Añadido	ROBOT_EN(X+1,Y-1)
Borrado	ROBOT_EN(X,Y)

MOVER_SO(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X-1,Y-1)
Añadido	ROBOT_EN(x-1, y-1)
Borrado	ROBOT_EN(X,Y)

MOVER_NE(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X+1, Y+1)
Añadido	ROBOT_EN(X+1, Y+1)
Borrado	ROBOT_EN(X,Y)

MOVER_NO(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X-1, Y+1)
Añadido	ROBOT_EN(X-1, Y+1)
Borrado	ROBOT_EN(X,Y)

MOVER_EN(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X+1, Y+1)
Añadido	ROBOT_EN(X+1, Y+1)
Borrado	ROBOT_EN(X,Y)

MOVER_ES(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X+1, Y-1)
Añadido	ROBOT_EN(X+1, Y-1)
Borrado	ROBOT_EN(X,Y)

MOVER_ON(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X-1, Y+1)
Añadido	ROBOT_EN(X-1, Y+1)
Borrado	ROBOT_EN(X,Y)

MOVER_OS(X,Y)

Precondición	ROBOT_EN(X,Y), LIBRE(X-1, Y-1)
Añadido	ROBOT_EN(X-1, Y-1)
Borrado	ROBOT_EN(X,Y)

2. Un robot tiene que limpiar un cuarto de una cocina, en el cual hay objetos

1 cocina gas d/ 4 quemadores: cocinaGas

1 Mesa: mesa

1 Heladera: heladera

El Suelo: suelo

Donde el robot para limpiar tendrá algunas consideraciones, el objetivo del robot será dejar todo limpio, algunas condiciones adicionales como cubrir con papel aluminio los quemadores.

Descripción de los predicados del dominio:

sucio(x): indica que x está sucio, donde: x pertenece {cocinaGas, suelo, heladera, mesa}

limpio(x): indica que x está limpio, donde: x pertenece {cocinaGas, suelo, heladera, mesa}

cubiertos(z): indica que z están cubiertos, donde z pertenece {quemadores}

sinCubrir (z): indica que z están sin cubrir, donde z pertenece { quemadores }

Modelar los operadores con la técnica STRIPS

- ✓ Limpiar_cocina(cocinaGas, suelo)
- ✓ Limpiar_suelo(suelo)
- ✓ Limpiar_heladera(heladera ,suelo, mesa)
- ✓ Limpiar_mesa (mesa)
- ✓ CubrirQuemadores

Resolución:

Limpiar_cocina (Cocina gas, suelo):

Precondición: sucio (cocina gas) \square limpio (suelo)

Añadido: limpio (cocina gas) \square sucio (suelo)

Borrado: sucio (cocina gas) \square limpio (suelo)

Limpiar_suelo (suelo):

Precondición: sucio (suelo)

Añadido: limpio (suelo)

Borrado: sucio (suelo)

Limpiar_heladera (heladera, suelo, mesa)

Precondición: sucio (heladera) \square limpio (suelo) \square limpia (mesa)

Añadido: limpio (heladera) \square sucio (suelo) \square sucio (mesa)

Borrado: sucio (heladera) \square limpio (suelo) \square limpio (mesa)

Limpiar_mesa (mesa)

Precondición: sucio (mesa)

Añadido: limpio (mesa)

Borrado: sucio (mesa)

Cubrir_quemadores

Precondición: sin cubrir (quemadores) \square limpia (cocina)

Añadido: cubrir (quemadores)

Borrado: sin cubrir (quemadores)

3. Una empresa de tecnología industrial desea automatizar el proceso de decisión empleado para dirigir el movimiento de un ascensor en grandes edificios. Para ello, ha decidido emplear técnicas de planificación automática de tareas. En los problemas que pretenden resolver, se considera la presencia de varias personas distribuidas por un mismo edificio que, en un momento dado, solicitan la atención del ascensor. Por lo tanto, se asume que, de cada persona, se conoce el piso en el que se encuentra, así como el piso al que desea ir.

Se pide:

Empleando la lógica de predicados, indicar qué predicados sirven para describir cada estado.

Pon un ejemplo de un estado inicial y un estado final con tres personas.

Representar los operadores en lenguajes STRIPS.

Resolución:

Descripción de los predicados del dominio

Ocupado(x): indica que **x** está ocupado donde **x** pertenece (ascensor).

Desocupado (x): indica que **x** está desocupado donde **x** pertenece (ascensor).

Espera (y, z): indica que **y** tiene que esperar el ascensor donde **y** pertenece (usuario) y **z** pertenece a (piso 1, 2, 3, 4, 5).

Servido (y, z): indica que **y** no tiene que esperar el ascensor donde **y** pertenece (usuario) y **z** pertenece a (piso 1, 2, 3, 4, 5).

Operadores (strips)

Subir (ascensor)

Bajar (ascensor)

Servido (usuario1, usuario2, usuario3)

Espera (usuario1, usuario2, usuario3)

Ascensor (piso1, piso2, piso3, piso4, piso5)

Estado inicial:

Esperar (usuario1) ☐ Esperar (usuario2) ☐ Esperar (usuario3) ☐ ascensor en (piso5)

Estado final:

Servido (usuario1) ☐ Servido (usuario2) ☐ Servido (usuario3) ☐ ascensor en (piso1)

Subir (ascensor)

Precondición: Ocupado (ascensor) ☐ Servido (usuario)

Añadido: desocupado (ascensor) ☐ espera (usuario)

Borrado: ocupado (ascensor) ☐ servido (usuario)

Bajar (ascensor)

Precondición: desocupado (ascensor) ☐ espera (usuario)

Añadido: servido (usuario) ☐ ocupado (ascensor)

Borrado: Desocupado (ascensor) ☐ espera (usuario)

Servido (usuario1, usuario2, usuario3)

Precondición: espera (usuario) ☐ desocupado (ascensor)

Añadido: servido (usuario) ☐ ocupado (ascensor)
Borrado: espera (usuario) ☐ desocupado (ascensor)

Espera (usuario1, usuario2, usuario3)

Precondición: servido (usuario) ☐ ocupado (ascensor)
Añadido: espera (usuario) ☐ desocupado (ascensor)
Borrado: servido (usuario) ☐ ocupado (ascensor)

Ascensor (piso1, piso2, piso3, piso4, piso5)

Precondición: desocupado (ascensor) ☐ espera (usuario)
Añadido: ocupado (ascensor) ☐ servido (usuario)
Borrado: desocupado (ascensor) ☐ espera (usuario)

4. Considérese el siguiente conjunto de predicados que describen el mundo en un problema de planificación de acciones de un camión T que transporta paquetes entre ciudades:
- paquete(x): el objeto x es un paquete.
 - ciudad(x): el objeto x es una ciudad.
 - pistas(c1,c2): las ciudades c1 y c2 están conectadas por pistas.
 - en(x,c): el objeto x (el camión o un paquete) está en la ciudad c.
 - dentro_camion(x): el paquete x está cargado en el camión.
 - descargado(): el camión está descargado.

Las acciones que se pueden realizar son las siguientes:

carga(p,c): el camión (que debe estar descargado) carga el paquete p en la ciudad c. Una vez cargado, el paquete ya no se considera que esté en la ciudad c.

descarga(p,c): el camión descarga el paquete p en la ciudad c.

ir(c1,c2): el camión se desplaza por pista desde la ciudad c1 a la ciudad c2.

Supongamos que deseamos encontrar la secuencia de acciones que a partir de un estado inicial en el que un paquete P1 está en Leon, un paquete P2 está en Managua, y el camión T está en Matagalpa, dejar finalmente el paquete P1 en Matagalpa, el paquete P2 en Leon y el camión descargado. Supondremos que existe una pista entre Leon y Managua y otra entre Managua y Matagalpa. Representar el problema en el formalismo STRIPS. Es decir, describir el estado inicial, el objetivo y las acciones.

Resolución:

P1->PAQUETE 1	C1->LEON	LEON->MANAGUA
P2-> PAQUETE 1	C2->MANAGUA	MANAGUA->MATAGALPA
	C3->MATAGALPA	

COMIENZO	FINAL
P1-> LEON	P1-> MATAGALPA
P2-> MANAGUA	P2-> LEON
T->DESCARGADO	T->DESCARGADO

Estado inicial:

EN(P1,C1)	EN(P2,C2)	EN(T,C3)	DESCARGADO()
-----------	-----------	----------	--------------

IR(C3,C2)	
Precondición	DESCARGADO() <input type="checkbox"/> CUIDAD(C3) <input type="checkbox"/> PISTA(C3,C2)
Añadido	EN(T,C2)
Borrado	CUIDAD(C2)

CARGA(P2,C2)	
Precondición	PAQUETE(P2) <input type="checkbox"/> CUIDAD(C2) <input type="checkbox"/> DESCARGADO()
Añadido	DENTRO_CAMION(P2)
Borrado	DESCARGADO()

IR(C2,C1)	
Precondición	CARGADO() <input type="checkbox"/> CUIDAD(C2) <input type="checkbox"/> PISTA(C2,C1)
Añadido	EN(T,C1)
Borrado	CUIDAD(C1)

DESCARGA(P2,C1)	
Precondición	PAQUETE(P2) <input type="checkbox"/> CUIDAD(C1)
Añadido	EN(P2,C1)
Borrado	

CARGA(P1,C1)	
Precondición	PAQUETE(P1) <input type="checkbox"/> CUIDAD(C1) <input type="checkbox"/> DESCARGADO()
Añadido	DENTRO_CAMION(P1)
Borrado	DESCARGADO()

IR(C1,C2)	
Precondición	CARGADO() <input type="checkbox"/> CUIDAD(C1) <input type="checkbox"/> PISTA(C1,C2)
Añadido	EN(T,C2)
Borrado	CUIDAD(C1)

IR(C2,C3)	
Precondición	CARGADO() <input type="checkbox"/> CUIDAD(C2) <input type="checkbox"/> PISTA(C2,C3)
Añadido	EN(T,C3)
Borrado	CUIDAD(C2)

DESCARGA(P1,C3)	
Precondición	PAQUETE(P1) <input type="checkbox"/> CUIDAD(C3)
Añadido	EN(P1,C3)
Borrado	