

RELACIÓN DE EJERCICIOS DEL TEMA 7

Ejercicio 1 (las torres de Hanoi):

Las torres de Hanoi son un juego matemático. El juego tiene tres palos, y cierta cantidad de discos de distintos tamaños que pueden ensartarse en cualquier palo. El juego empieza con los discos apilados en orden ascendente sobre uno de los palos, con el más pequeño en lo alto, formando una figura cónica. El objetivo es mover toda la pila a otro palo, obedeciendo estas reglas:

- Sólo se puede mover un disco cada vez.
- Cada movimiento consiste en coger el disco de arriba de uno de los palos y colocarlo en otro palo, quedando encima de los que pudiera haber anteriormente.
- No se puede colocar un disco encima de otro más pequeño.

Se pide:

a) Escribe el dominio de planificación de las torres de Hanoi en PDDL. Emplea tres predicados: el predicado unario *Clear* y los predicados binarios *On* y *Smaller*. Basta con una sola acción *Move*.

b) Especifica el siguiente problema de tres discos en PDDL: *D3*, *D2* y *D1* (listados desde el más grande hasta el más pequeño).

Ejercicio 2 (robot de reparto):

Tenemos un robot de reparto de paquetes que puede mover paquetes entre la Escuela de Informática (*Computer Science*), la de Telecomunicaciones (*Telecommunications*) y la Facultad de Medicina (*Medicine*). Supondremos que el robot puede transportar una cantidad ilimitada de carga. Cuando viaja entre la Facultad de Medicina y los otros dos centros, debe salir al exterior. En cambio, puede viajar bajo techo entre ambas escuelas. Si el robot sale al exterior mientras llueve, se mojará a menos que lleve un paraguas. Puede conseguir un paraguas en la Escuela de Informática.

Se pide:

a) Escribe el dominio de planificación en PDDL.

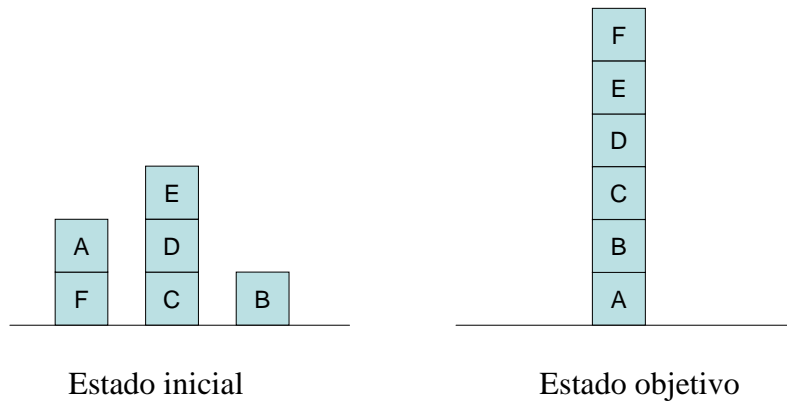
b) Especifica el siguiente problema en PDDL. Hay cuatro paquetes: *P1* (está en Telecomunicaciones y debe ser entregado en Informática), *P2* (está en Informática y debe entregarse en Telecomunicaciones), *P3* (está en Medicina, y debe enviarse a Telecomunicaciones), y *P4* (está en Informática y hay que enviarlo a Medicina). El robot está en Telecomunicaciones, y está lloviendo. Debemos entregar todos los paquetes manteniendo seco el robot.

Ejercicio 3 (mundo de bloques):

Se le asigna a un robot la tarea de colocar varios bloques formando una torre. Se pide:

a) Escribe el dominio de planificación en PDDL.

b) Especifica el siguiente problema en PDDL.



Ejercicio 4 (dominó solitario):

Un dominó es una pieza rectangular con una línea que divide su cara superior en dos mitades cuadradas. Cada mitad está marcada con cierto número de puntos, o bien está vacía. La tarea consiste en jugar un solitario con una cierta bolsa de dominós. El juego comienza colocando una pieza cualquiera. Dicha pieza es la primera de la línea de juego, una serie de dominós en la que los dominós adyacentes se tocan con valores iguales. El jugador extiende la línea de juego cada vez con un dominó colocado en cualquiera de los dos extremos. El juego termina si todos los dominós disponibles están colocados sobre la mesa, es decir, la línea de juego contiene todos los dominós. Nótese que la bolsa de dominós puede contener algunos repetidos. Se pide:

- a) Escribe el dominio de planificación en PDDL.
- b) Especifica el siguiente problema en PDDL. La bolsa de dominós contiene: 5-0, 5-0, 4-4, 4-6, 6-1, 0-3, 3-4, y 1-0.

Ejercicio 5 (planificación de itinerarios):

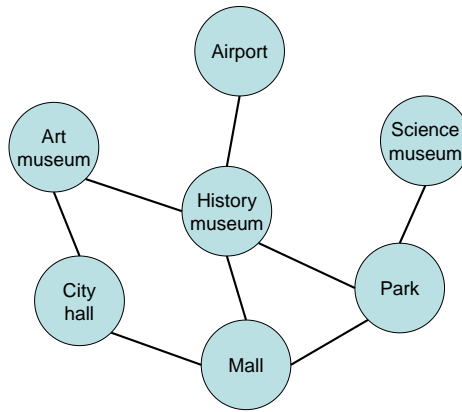
Son las 3PM, y Speedy acaba de llegar al aeropuerto. Quiere visitar los tres museos principales de la ciudad antes de que acabe el día: Arte (*Art*), Historia (*History*) y Ciencias (*Science*). Debido a los atascos de tráfico, las únicas conexiones disponibles son las que se muestran en el grafo de abajo. Cada movimiento desde un lugar a otro adyacente necesita una hora, y los horarios de visita de los museos son los siguientes:

- Museo de Arte: 4PM, 5PM o 6PM.
- Museo de Historia: 4PM, 5PM, 6PM o 7PM.
- Museo de Ciencias: 4PM, 5PM o 6PM.

Esto quiere decir que, por ejemplo, Speedy puede ir del aeropuerto al museo de Historia, llegando allí a las 4PM, que es uno de los momentos en que se puede visitar. A continuación podría ir al centro comercial (*Mall*), adonde llegaría a las 5PM.

Se pide:

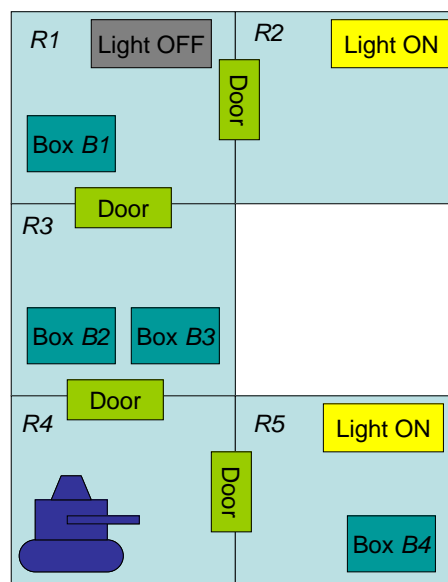
- a) Escribe el dominio de planificación en PDDL.
- b) Especifica el problema de Speedy en PDDL.



Ejercicio 6 (el mundo de Shakey):

Shakey es un robot que está diseñado para moverse entre un conjunto de habitaciones conectadas por puertas. Algunas de las habitaciones tienen interruptores de luz, de modo que la luz de la habitación puede encenderse (ON) o apagarse (OFF). Además, hay algunas cajas repartidas por este mundo. Shakey puede empujar una caja de una habitación a otra si hay una puerta que las conecte. Nótese que sólo puede empujar una caja cada vez, y que no puede empujar si está colocado encima de una caja. A fin de accionar el interruptor de luz de una habitación, Shakey debe subirse en una caja que se encuentre en esa habitación. De otra manera no podría alcanzar el interruptor. Se pide:

- Escribe el dominio de planificación en PDDL.
- Especifica en PDDL el problema de mover todas las cajas a la habitación *R1* y apagar todas las luces, para el siguiente mundo:



N.B.: Shakey fue un robot real que fue desarrollado desde 1966 hasta 1972. En 1970, la revista *Life* se refirió a él como la “primera persona electrónica” (*the first electronic person*).

POSIBLES SOLUCIONES:

Ejercicio 1 (las torres de Hanoi):

a) El dominio de planificación se especifica como sigue:

Predicados:

$Clear(x)$ quiere decir que no hay ningún objeto sobre x , donde x puede ser un palo o un disco.

$On(x,y)$ quiere decir que x está sobre y . Nótese que x debe ser un disco, mientras que y puede ser un palo o un disco.

$Smaller(x,y)$ quiere decir que x es más pequeño que y , es decir, x puede colocarse encima de y . Nótese que x debe ser un disco, mientras que y puede ser un palo o un disco.

Acciones:

Action($Move(disc,orig,dest)$,

PRECOND: $Clear(disc) \wedge On(disc,orig) \wedge Clear(dest) \wedge Smaller(disc,dest)$

EFFECT: $\neg Clear(dest) \wedge \neg On(disc,orig) \wedge Clear(orig) \wedge On(disc,dest)$)

b) El problema específico para tres discos $D3$, $D2$ y $D1$ se especifica como sigue:

Estado inicial:

$Clear(D1) \wedge Clear(B) \wedge Clear(C) \wedge$
 $On(D1,D2) \wedge On(D2,D3) \wedge On(D3,A) \wedge$
 $Smaller(D1,D2) \wedge Smaller(D1,D3) \wedge Smaller(D2,D3) \wedge$
 $Smaller(D1,A) \wedge Smaller(D1,B) \wedge Smaller(D1,C) \wedge$
 $Smaller(D2,A) \wedge Smaller(D2,B) \wedge Smaller(D2,C) \wedge$
 $Smaller(D3,A) \wedge Smaller(D3,B) \wedge Smaller(D3,C)$

Estado objetivo:

$Clear(D1) \wedge Clear(A) \wedge Clear(B) \wedge$
 $On(D1,D2) \wedge On(D2,D3) \wedge On(D3,C)$

Ejercicio 2 (robot de reparto):

a) El dominio de planificación se especifica como sigue:

Predicados:

$At(x,y)$ quiere decir que el objeto x está en la ubicación y . Nótese que y debe ser uno de los tres centros, mientras que x puede ser el robot, un paquete o el paraguas.

$On(x,y)$ quiere decir que x está siendo acarreado por y . Nótese que y solamente puede ser el robot, mientras que x can be either a package or the umbrella.

$DryPath(x,y)$ quiere decir que el camino desde x hasta y está seco.

$Wet(x)$ quiere decir que x está mojado.

$Robot(x)$ quiere decir que x es un robot.

$Cargo(x)$ quiere decir que x es carga, es decir, un paquete o el paraguas.

Umbrella(x) quiere decir que *x* es un paraguas.

Acciones:

Action(Load(robot,cargo,location),
 PRECOND: $At(robot,location) \wedge At(cargo,location) \wedge Robot(robot) \wedge$
 Cargo(cargo)
 EFFECT: $\neg At(cargo,location) \wedge On(cargo,robot))$

Action(UnLoad(robot,cargo,location),
 PRECOND: $On(cargo,robot) \wedge At(robot,location) \wedge Robot(robot) \wedge$
 Cargo(cargo)
 EFFECT: $At(cargo,location) \wedge \neg On(cargo,robot))$

Action(MoveDry(robot,umbrella,orig,dest),
 PRECOND: $At(robot,orig) \wedge Robot(robot) \wedge Umbrella(umbrella) \wedge [$
 DryPath(orig,dest) \vee On(umbrella,robot)]
 EFFECT: $\neg At(robot,orig) \wedge At(robot,dest))$

Action(MoveWet(robot,umbrella,orig,dest),
 PRECOND: $At(robot,orig) \wedge Robot(robot) \wedge Umbrella(umbrella) \wedge$
 $\neg DryPath(orig,dest) \wedge \neg On(umbrella,robot)$
 EFFECT: $\neg At(robot,orig) \wedge At(robot,dest) \wedge Wet(robot))$

b) El problema específico que hay que resolver es el siguiente:

Estado inicial:

$At(P1,Telecommunications) \wedge At(P2,ComputerScience) \wedge$
 $At(P3,Medicine) \wedge At(P4,ComputerScience) \wedge$
 $At(TheUmbrella,ComputerScience) \wedge$
 $At(TheRobot,Telecommunications) \wedge$
 $Robot(TheRobot) \wedge Umbrella(TheUmbrella) \wedge$
 $Cargo(P1) \wedge Cargo(P2) \wedge Cargo(P3) \wedge Cargo(P4) \wedge$
 $Cargo(TheUmbrella) \wedge$
 $DryPath(Telecommunications,ComputerScience) \wedge$
 $DryPath(ComputerScience,Telecommunications)$

Estado objetivo:

$At(P1,ComputerScience) \wedge At(P2,Telecommunications) \wedge$
 $At(P3,Telecommunications) \wedge At(P4,Medicine) \wedge$
 $\neg Wet(TheRobot)$

Ejercicio 3 (mundo de bloques):

a) El dominio de planificación se especifica como sigue:

Predicados:

$On(x,y)$ quiere decir que el objeto x está directamente encima del objeto y .
Nótese que y puede ser la mesa o un bloque, mientras que x debe ser un bloque.

$Clear(x)$ quiere decir que no hay nada encima del bloque x .

$Block(x)$ quiere decir que x es un bloque.

$Table(x)$ quiere decir que x es una mesa.

Acciones:

$Action(PutOn(block,orig,dest),$
PRECOND: $Clear(block) \wedge Clear(dest) \wedge Block(dest) \wedge On(block,orig)$
EFFECT: $\neg On(block,orig) \wedge On(block,dest) \wedge Clear(orig) \wedge \neg Clear(dest))$

$Action(PutTable(block,orig,table),$
PRECOND: $Clear(block) \wedge Block(orig) \wedge On(block,orig) \wedge Table(table)$
EFFECT: $\neg On(block,orig) \wedge On(block,table) \wedge Clear(orig))$

b) El problema específico que debemos resolver es el siguiente:

Estado inicial:

$Block(A) \wedge Block(B) \wedge Block(C) \wedge Block(D) \wedge Block(E) \wedge Block(F) \wedge$
 $Table(T) \wedge On(F,T) \wedge On(A,F) \wedge$
 $On(C,T) \wedge On(D,C) \wedge On(E,D) \wedge$
 $On(B,T) \wedge$
 $Clear(A) \wedge Clear(E) \wedge Clear(B)$

Estado objetivo:

$On(F,E) \wedge On(E,D) \wedge On(D,C) \wedge$
 $On(C,B) \wedge On(B,A) \wedge On(D,C) \wedge$
 $Clear(F)$

Ejercicio 4 (dominó solitario):

a) El dominio de planificación se especifica como sigue:

Predicados:

$Domino(x,y,z)$ quiere decir que hay un dominó x que tiene y puntos en la mitad izquierda y z puntos en la mitad derecha.

$Placed(x,y)$ quiere decir que el dominó x está colocado sobre la mesa y .

$Empty(x)$ quiere decir que la mesa x está vacía, es decir, no se ha colocado aún ningún dominó sobre ella.

$Leftmost(x,y)$ quiere decir que y es el dominó colocado más a la izquierda sobre la mesa x .

$Rightmost(x,y)$ quiere decir que y es el dominó colocado más a la derecha sobre la mesa x .

Acciones:

$Action(PlaceFirst(table,domino,leftside,rightside),$

PRECOND: $Empty(table) \wedge Domino(domino, leftside, rightside)$
 EFFECT: $\neg Empty(table) \wedge Placed(domino) \wedge Leftmost(table, domino) \wedge Rightmost(table, domino)$

Action(*Reverse*(*domino*, *leftside*, *rightside*),
 PRECOND: $\neg Placed(domino) \wedge Domino(domino, leftside, rightside)$
 EFFECT: $\neg Domino(domino, leftside, rightside) \wedge Domino(domino, rightside, leftside)$

Action(*PlaceLeft*(*table*, *leftmost*, *domino*, *leftside*, *rightside*, *otherside*),
 PRECOND: $Leftmost(table, leftmost) \wedge Domino(domino, leftside, rightside) \wedge \neg Placed(domino, table) \wedge Domino(leftmost, rightside, otherside)$
 EFFECT: $\neg Leftmost(table, leftmost) \wedge Placed(domino, table) \wedge Leftmost(table, domino)$

Action(*PlaceRight*(*table*, *rightmost*, *domino*, *leftside*, *rightside*, *otherside*),
 PRECOND: $Rightmost(table, rightmost) \wedge Domino(domino, leftside, rightside) \wedge \neg Placed(domino, table) \wedge Domino(rightmost, otherside, leftside)$
 EFFECT: $\neg Rightmost(table, rightmost) \wedge Placed(domino, table) \wedge Rightmost(table, domino)$

b) El problema específico que hay que resolver es el siguiente:

Estado inicial:

$Domino(A, 5, 0) \wedge Domino(B, 5, 0) \wedge Domino(C, 4, 4) \wedge Domino(D, 4, 6) \wedge Domino(E, 6, 1) \wedge Domino(F, 0, 3) \wedge Domino(G, 3, 4) \wedge Domino(H, 1, 0) \wedge Empty(Table)$

Estado objetivo:

$Placed(A) \wedge Placed(B) \wedge Placed(C) \wedge Placed(D) \wedge Placed(E) \wedge Placed(F) \wedge Placed(G) \wedge Placed(H)$

Ejercicio 5 (planificación de itinerarios):

a) El dominio de planificación se especifica como sigue:

Predicados:

Before(*x*, *y*) quiere decir que el tiempo *x* es el inmediatamente anterior del tiempo *y*.

Visited(*x*) quiere decir que el museo *x* ya ha sido visitado.

Open(*x*, *y*) quiere decir que el museo *x* está abierto en el tiempo *y*.

Now(*x*) quiere decir que el tiempo actual es *x*.

Location(*x*) quiere decir que la ubicación actual del robot es *x*.

Path(*x*, *y*) quiere decir que hay un camino desde la ubicación *x* hasta la ubicación *y*.

Acciones:

Action(Move(now,then,orig,dest),
 PRECOND: $Now(now) \wedge Before(now,then) \wedge Location(orig) \wedge Path(orig,dest) \wedge$
 $\neg Open(dest,then)$
 EFFECT: $\neg Now(now) \wedge Now(then) \wedge Location(dest)$)

Action(Visit(now,then,orig,dest),
 PRECOND: $Now(now) \wedge Before(now,then) \wedge Location(orig) \wedge Path(orig,dest) \wedge$
 $Open(dest,then)$
 EFFECT: $\neg Now(now) \wedge Now(then) \wedge Location(dest) \wedge Visited(dest)$)

b) El problema específico que debemos resolver es el siguiente:

Estado inicial:

$Now(3PM) \wedge Before(3PM,4PM) \wedge Before(4PM,5PM) \wedge$
 $Before(5PM,6PM) \wedge Before(6PM,7PM) \wedge$
 $Location(Airport) \wedge$
 $Path(Airport,History) \wedge Path(History,Airport) \wedge$
 $Path(Art,History) \wedge Path(History,Art) \wedge$
 $Path(Mall,History) \wedge Path(History,Mall) \wedge$
 $Path(Park,History) \wedge Path(History,Park) \wedge$
 $Path(Art,City) \wedge Path(City,Art) \wedge$
 $Path(Mall,City) \wedge Path(City,Mall) \wedge$
 $Path(Mall,Park) \wedge Path(Park,Mall) \wedge$
 $Path(Park,Science) \wedge Path(Science,Park) \wedge$
 $Open(Art,4PM) \wedge Open(Art,5PM) \wedge Open(Art,6PM) \wedge$
 $Open(History,4PM) \wedge Open(History,5PM) \wedge$
 $Open(History,6PM) \wedge Open(History,7PM) \wedge$
 $Open(Science,4PM) \wedge Open(Science,5PM) \wedge Open(Science,6PM)$

Estado objetivo:

$Visited(History) \wedge Visited(Art) \wedge Visited(Science)$

Ejercicio 6 (el mundo de Shakey):

a) El dominio de planificación se especifica como sigue:

Predicados:

$At(x,y)$ quiere decir que el objeto x está en la ubicación y . Nótese que y debe ser una habitación, mientras que x puede ser una caja o el robot.

$Door(x,y)$ quiere decir que hay una puerta que conecta la habitación x con la habitación y .

$High(x)$ quiere decir que el robot x está encima de una caja.

$On(x,y)$ quiere decir que x está encima de y . Nótese que y ha de ser una caja, mientras que x ha de ser el robot.

$Switch(x)$ quiere decir que hay un interruptor de luz en la habitación x .

$LightOn(x)$ quiere decir que la luz de la habitación x está encendida.

$Box(x)$ quiere decir que x es una caja.

$Robot(x)$ quiere decir que x es un robot.

Acciones:

$Action(Move(robot, orig, dest),$
PRECOND: $At(robot, orig) \wedge Robot(robot) \wedge \neg High(robot) \wedge Door(orig, dest)$
EFFECT: $\neg At(robot, orig) \wedge At(robot, dest)$)

$Action(Push(robot, box, orig, dest),$
PRECOND: $At(robot, orig) \wedge At(box, orig) \wedge Robot(robot) \wedge Box(box) \wedge \neg High(robot) \wedge Door(orig, dest)$
EFFECT: $\neg At(robot, orig) \wedge \neg At(box, orig) \wedge At(robot, dest) \wedge At(box, dest)$)

$Action(ClimbUp(robot, box, room),$
PRECOND: $At(robot, room) \wedge At(box, room) \wedge Robot(robot) \wedge Box(box) \wedge \neg High(robot)$
EFFECT: $High(robot) \wedge On(robot, box)$)

$Action(ClimbDown(robot, box, room),$
PRECOND: $At(robot, room) \wedge At(box, room) \wedge Robot(robot) \wedge Box(box) \wedge On(robot, box)$
EFFECT: $\neg High(robot) \wedge \neg On(robot, box)$)

$Action(SwitchOn(robot, box, room),$
PRECOND: $At(robot, room) \wedge At(box, room) \wedge Robot(robot) \wedge Box(box) \wedge On(robot, box) \wedge Switch(room) \wedge \neg LightOn(room)$
EFFECT: $LightOn(room)$)

$Action(SwitchOff(robot, box, room),$
PRECOND: $At(robot, room) \wedge At(box, room) \wedge Robot(robot) \wedge Box(box) \wedge On(robot, box) \wedge Switch(room) \wedge LightOn(room)$
EFFECT: $\neg LightOn(room)$)

b) El problema específico que hay que resolver es el siguiente:

Estado inicial:

$Robot(Shakey) \wedge Box(B1) \wedge Box(B2) \wedge Box(B3) \wedge Box(B4)$
 $At(Shakey, R4) \wedge At(B1, R1) \wedge$
 $At(B2, R3) \wedge At(B3, R3) \wedge At(B4, R5) \wedge$
 $Door(R1, R2) \wedge Door(R2, R1) \wedge$
 $Door(R1, R3) \wedge Door(R3, R1) \wedge$
 $Door(R3, R4) \wedge Door(R4, R3) \wedge$
 $Door(R4, R5) \wedge Door(R5, R4) \wedge$
 $Switch(R1) \wedge Switch(R2) \wedge Switch(R5) \wedge$
 $LightOn(R2) \wedge LightOn(R5)$

Estado objetivo:

$At(B1, R1) \wedge At(B2, R1) \wedge At(B3, R1) \wedge At(B4, R1) \wedge$
 $\neg LightOn(R1) \wedge \neg LightOn(R2) \wedge \neg LightOn(R5)$

