Prácticas

Sistemas Inteligentes I

Sesión 7. Planificación

José A. Montenegro Montes
monte@lcc.uma.es

Resumen

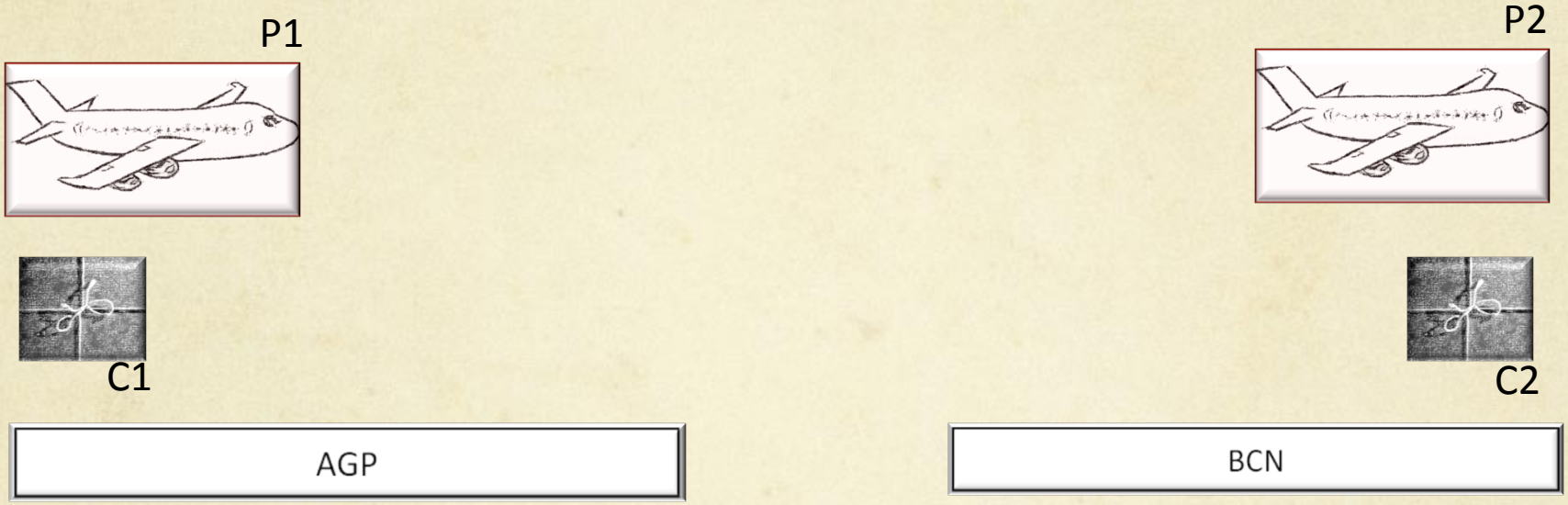
- Problema 10.1.1 Air cargo transport
- Problema Mono y Plátanos

Ejemplo

10.1.1 Example: Air Cargo transport

Página 369 AIMA

10.1.1 Air cargo transport



INIT

Plane(P1)∧ Plane(P2)∧
Cargo(C1)∧ Cargo(C2)∧
Airport(AGP)∧ Airport(BCN)∧
At(P1, AGP) ∧ At(C1, AGP) ∧
At (P2, BCN) ∧ At(C2, BCN)

10.1.1 Air cargo transport

GOAL $At(C1, BCN) \wedge At(C2, AGP)$

10.1.1 Air cargo transport

- Action(Load(c,p,a),
  PRECOND: Cargo(c), Plane(p), Airport(a), At(c,a), At(p,a)
  EFFECT: ¬At(c,a), In(c,p))

- Action(UnLoad(c,p,a),
  PRECOND: Cargo(c), Plane(p), Airport(a), In(c,p), At(p,a),
  EFFECT: At(c,a), ¬In(c,p))

- Action(Fly(p,from,to),
  PRECOND: Plane(p), Airport(from) , Airport(to), At(p,from)
  EFFECT: ¬At(p,from), At(p,to))

JavaGP Air cargo transport

ProblemAvion.txt

start (
plane(p1),
plane(p2),
cargo(c1),
cargo(c2),
airport(agp),
airport(bcn),
at(p1, agp),
at(c1, agp),
at(p2, bcn),
at(c2, bcn))

goal(
at(c1, bcn),
at(c2, agp))

JavaGP Air cargo transport

DomainAvion.txt

operator load(C,P,A)
pre: cargo(C), plane(P), airport(A), at(C,A), at(P,A)
post: ~at(C,A), in(C,P)

operator unload(C,P,A)
pre: cargo(C), plane(P), airport(A),in(C,P), at(P,A)
post: at(C,A), ~in(C,P)

operator fly(P,From,To)
pre: plane(P), airport(From), airport(To), at(P,From)
post: ~at(P,From), at(P,To)

JavaGP Air cargo transport

Ejecución

java -jar javagp.jar –d domainAvion.txt -p problemAvion.txt

Result_trace

INFO: Expanding graph
INFO: Goals not possible with 1 steps
INFO: Expanding graph
INFO: Extracting solution
INFO: Plan not found with 2 steps
INFO: Expanding graph
INFO: Extracting solution
INFO: Plan found with 3 steps
INFO: Plan found

load(c1,p1,agp)
load(c2,p2,bcn)
fly(p2,bcn,agp)
fly(p1,agp,bcn)
unload(c2,p2,agp)
unload(c1,p1,bcn)

INFO: Planning took 0s

Práctica

Planes

Problema Mono y Plátanos

El problema del mono y los plátanos (*the monkey-and-bananas problem*) es aquel al que se enfrenta un mono (*monkey*) en un laboratorio donde hay varios plátanos colgando del techo, fuera de su alcance.

Hay una caja (*box*) disponible que permitirá el mono alcanzar los plátanos si se sube en ella.

Al principio el mono está en A, los plátanos en B, y la caja en C.

El mono y la caja tienen altura (*height*) baja (*low*), pero si el mono se sube a la caja tendrá altura elevada (*high*), la misma que tienen los plátanos.

Problema Mono y Plátanos

Las **acciones** disponibles para el mono son:

1. ir (*go*) de un lugar a otro,
2. empujar (*push*) un objeto de un lugar a otro,
3. subirse (*climbUp*) o
4. bajarse (*climbDown*) de un objeto, y
5. agarrar (*grasp*) o
6. soltar (*unGrasp*) un objeto.

El **resultado** de *grasp* es que el mono obtiene el objeto si el mono y el objeto están en el mismo lugar y a la misma altura.

Problema Mono y Plátanos

Debes hacer lo siguiente:

a) Escribir la descripción del estado inicial y final.

b) Escribir los seis esquemas de acción.

Nota: Ten en cuenta que si el objeto es demasiado pesado, su posición seguirá siendo la misma cuando se aplique el esquema *push*. Incluye esta consideración en tu esquema de acción para tener en cuenta los objetos pesados.

c) Crea los ficheros *mb_domain.txt* y *mb_problem.txt* para el dominio y el problema, respectivamente. A continuación suminístraselos a JavaGP (una implementación del algoritmo GRAPHPLAN) y comprueba que se encuentra un plan correcto. Guarda la salida del programa en el archivo **result_trace.txt**

Problema Mono y Plátanos

d) OPCIONAL: Supón que el mono quiere engañar a los científicos, que han ido a tomarse un café, cogiendo los plátanos, pero dejando la caja en su lugar original. Escribe esto como un objetivo genérico (es decir, sin suponer que la caja está necesariamente en C) en el lenguaje del cálculo de situaciones. ¿Puede resolver este objetivo un sistema de planificación clásico?

Si la respuesta es no, cambia el objetivo general a un objetivo más específico que pueda ser resuelto por JavaGP.

Crea el archivo **mb_problem2.txt** con el problema. Ejecute el problema en JavaGP con **mb_domain.txt** y **mb_problem2.txt**. Almacena la salida en **result_trace2.txt**.

Problema Mono y Plátanos

Copiar el contenido de los archivos al campus Virtual
mb_domain.txt, mb_problem.txt y **result_trace.txt**

Si realizas la parte opcional copia también los archivos
mb_problem2.txt y **result_trace2.txt**.