

Inteligencia Artificial : Segundo Parcial 29/10/2012

Ejercicio 1

STRIPS

C = cofre

L = llave

	Abrir(C,L)	Cerrar(C,L)	Poner(L,C)	Sacar(L,C)	Tragar(L)
Precondiciones	Abre(C,L) Cerrado(C) enPiso(L)	Abre(C,L) Abierto(C) enPiso(L)	Abierto(C) enPiso(L)	Abierto(C) dentroDe(L,C)	enPiso(L)
Add-list	Abierto(C)	Cerrado(C)	dentroDe(L,C)	enPiso(L)	-
Delete-lis	Cerrado(C)	Abierto(C)	enPiso(L)	dentroDe(L,C)	enPiso(L)

- a) Calculo de situación: dar axioma de posibilidad para acción **abrir**.
- b) Calculo de situación: dar axioma de estado sucesor para relaciones **enPiso** y **dentroDe**.
- c) Describir y relacionar los siguientes conceptos:
 - Problema del marco.
 - Problema representacional del marco.
 - Axioma del marco (SC).
 - Axioma del estado sucesor (SC).

Ejercicio 2

Inicial = { cerrado(c1), abierto(c2), dentroDe(l2, c2), enPiso(l1), abre(l1, c1), abre(l2, c2) }

- a) Realizar traza de aplicando algoritmo de planificación STRIPS para la meta:
M = { dentroDe(l2, c1), cerrado(c1), dentroDe(l1, c2) }
- b) Se alcanza la meta M en el inciso anterior? Que sucede si las metas están en otro orden?
M = { dentroDe(l2, c1), dentroDe(l1, c2) , cerrado(c1) }
Explicar brevemente sin realizar traza.
- c) Dar 2 soluciones alternativas para b. Describir sin realizar trazas.

Ejercicio 3

$T = \langle W, D \rangle$

$W = \{ x, t \rightarrow \neg u \}$

$D = \{ x:z/z, \ x:\neg z/v, \ t \vee u:p/p, \ z:t/t, \ z:w,u/u, \ u:q/q \}$

- Dar todas las extensiones de T.
- Dar algunos ejemplos de conclusiones cautas y osadas en T.
- Hallar un D' tal que $T1 = \langle W, D \cup D' \rangle$ no tenga extensiones.

Ejercicio 4

a) Definir formalmente:

- Escenario
- Explicación
- Explicación minimal
- Extensión

b)

Las leyes de la robotica de asimov (al menos las 2 primeras):

1 – No poder perjudicar a ningún humano....

2 – Acatar orden, salvo que este en conflicto con 1...

F	H
Perj_h(A) ^evitar_acc_perj(A) -> ¬hacer(A) Preserva_h(A) ^hacer_acc_pres(A) -> hacer(A) H_ordena(A) ^acatar_orden(A) -> hacer(A) Perj_h(dispon_sosp) Preserva_h(dispon_sosp) H_ordena(dispon_sosp)	Evitar_acc_perj(A) Hacer_acc_pres(A) Acatar_orden(A)

- 1) Dar explicaciones minimales para hacer(dispon_sosp) y ¬hacer(dispon_sosp).
- 2) Dar las extensiones.
- 3) Agregar regla de cancelación en F de manera de modelar la prioridad de la 1ra regla por sobre la 2da regla.