

Examen Final de Aprendizaje

(23 de enero de 2002)

Duración: 3 horas

Problemas

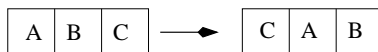
- (3 puntos) Tenemos un vector con n posiciones donde en cada una de ellas tenemos una letra, disponemos de un operador que nos permite intercambiar dos posiciones consecutivas del vector definido de la siguiente manera:

Operador	swap(x,y,vx,vy)
Prec.	next(x,y), val(x,vx), val(y,vy)
Supr.	val(x,vx), val(y,vy)
Añadir	val(x,vy), val(y,vx)

Los predicados con los que se puede describir una escena son:

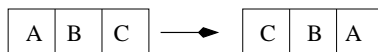
next(x,y): La posición y es la siguiente a la x
val(x,vx): la posición x tiene el valor vx

- Genera el plan que lleva del vector de la izquierda al de la derecha utilizando el operador **swap** (basta con indicar que pasos hacen falta).



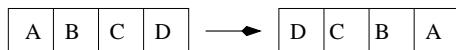
Aprende el operador **back** generalizando el plan generado usando la técnica de EBL utilizada por STRIPS. Indica la tabla triangular inicial justificando que predicados son los relevantes, la tabla generalizada y la tabla restringida con las unificaciones de variables que haces y por que.

- Genera el plan que lleva del vector de la izquierda al de la derecha utilizando el operador **swap** (basta con indicar que pasos hacen falta).



Aprende el operador **mirror** generalizando el plan generado usando la técnica de EBL utilizada por STRIPS. Indica la tabla triangular inicial justificando que predicados son los relevantes, la tabla generalizada y la tabla restringida con las unificaciones de variables que haces y por que.

- Dados los planes de la tabla, que resuelven el problema de la figura, ¿Es alguno más ventajoso uno que el otro? Compáralos respecto al número de pasos primitivos y el número de precondiciones que hay que comprobar, y comenta los resultados.



Plan 1	Plan 2	Plan 3	Plan 4
swap(3,4,C,D)	back(2,3,4,B,C,D)	mirror(1,2,3,A,B,C)	swap(3,4,C,D)
swap(2,3,B,D)	swap(1,2,A,D)	swap(3,4,A,D)	back(1,2,3,A,B,D)
swap(1,2,A,D)	back(2,3,4,A,B,C)	mirror(1,2,3,C,B,D)	mirror(2,3,4,A,B,C)
swap(3,4,B,C)	swap(3,4,A,B)	swap(2,3,B,C)	
swap(2,3,A,C)			
swap(3,4,A,B)			

2. (2 puntos) Es desitja dissenyar un mecanisme de detecció de falles per a l'intermitent d'un vehicle, de l'estil $\leftarrow \bullet \Rightarrow$, on anomenem L l'intermitent esquerre \leftarrow , R l'intermitent dret \Rightarrow , i S el senyal rodó \bullet . Es vol que el mecanisme s'activi (sortida 1) quan alguna cosa vagi malament. En cas contrari, es vol sortida 0. L'intermitent funciona be quan L està encés i R , S apagats, quan R està encés i L , S apagats, o quan L , R està encesos i S també (llums de perill). Es demana:
- Plantejar la tasca com una funció lògica de 3 entrades.
 - Mostrar que no és separable linealment.
 - Ebossar una solució amb perceptrons.

Cuestiones

- (1 1/3 puntos) Sigui una xarxa de Hopfield de $N = 4$ neurones. S'emmagatzemen els patrons $p_1 = (1, 1, 1, -1)$ i $p_2 = (-1, 1, 1, 1)$ (i res més). Es demana:
 - Calcular la matriu de pesos W .
 - Per $z = (z_1, z_2, z_3, z_4)$, sigui $f(z) = \text{sgn}[Wz]$, on la funció signe d'un vector $\text{sgn}[\cdot]$ s'aplica component a component. Donar una expressió per $f(z)$.
 - Demostrar que $f(p_1) = p_1$ i $f(p_2) = p_2$. Com es pot interpretar això en termes d'atractors?
 - Demostrar que $f(f(z)) = z$. Com es pot interpretar això en termes d'atractors?
 - Per quins altres z es compleix $f(z) = z$? És a dir, quins altres atractors n'hi ha? Com es pot interpretar això en termes d'atractors?
- (3 2/3 puntos) Contesta **brevemente** a estas cuestiones:
 - ¿Por que el aprendizaje que realiza el algoritmo del espacio de versiones no depende del orden de presentación de ejemplos?
 - ¿Como es posible que el error estimado de un árbol de decisión disminuya tras el proceso de pruning, si estamos eliminando ejemplos del árbol?
 - ¿Por que el algoritmo de EBL de STRIPS puede obtener planes mas generales de lo esperado que el plan concreto utilizado para la generalización? ¿Podria calificarse a esta particularidad de aprendizaje inductivo?
 - Explica que es una regla de asociación y comenta en que principio se basa el algoritmo de calculo de conjuntos frecuentes usado para la generación de reglas de asociación.
 - Supongamos que tenemos un lenguaje de descripción de conceptos que admite una única disyunción (como en el problema de espacio de versiones que resolvisteis) y tenemos el dominio de los coches con los siguientes atributos:

Precio: caro, medio, barato.

Motor: diesel, gasolina.

Tamaño: grande, mediano, pequeño.

Suponiendo que los disyuntandos no estan ordenados (igual que la otra vez), cual sería la modificación que se obtendria en este espacio de versiones, si le presentaramos cada uno los contraejemplos indicados (cada contraejemplo se ha de aplicar al espacio de versiones inicial, no son tres contraejemplos sucesivos)

$G = \{((? \text{ gasolina } ?) \text{ (caro } ? \text{ grande)})\}$

$S = \{(((? \text{ gasolina } \text{ pequeño}) \text{ (caro } \text{ diesel } \text{ grande)}))\}$

- ((caro gasolina mediano) (caro gasolina grande))
- ((barato gasolina grande) (caro diesel pequeño))
- ((barato gasolina pequeño) (caro diesel grande))

Las notas saldrán el día 29 de enero

La revisión será hasta el día 31 de enero via e-mail