

Apellidos:

Nombre:

IMPORTANTE: Contestar los ejercicios de manera clara y concisa, usando para ello el espacio en blanco que aparece a continuación de cada uno de ellos (excepto en los ejercicios 5 y 6, en los que se permite usar hojas adicionales).

Ejercicio 1: [1.2 puntos]

Describir (mediante pseudocódigo o mediante un programa Prolog) la implementación de un sistema de deducción en el que el conocimiento está expresado en forma de reglas y existen objetivos retardables ¿Con qué finalidad se usan los objetivos retardables?

Ejercicio 2: [0.9 puntos]

Definir una gramática de cláusulas definidas para reconocer el lenguaje

$$\{a^n b^{2n} : n \geq 1\} \cup \{a^{2n} b^n : n \geq 1\}$$

Utilizar el símbolo **s** como símbolo inicial de esta gramática, incluyendo además un argumento en el que se recoja la longitud de la cadena de símbolos reconocida. Ejemplos:

?- phrase(s(X), [a, a, b, b, b]).

No

?- phrase(s(X), [a, a, b, b, b, b]).

X = 6

Yes

?- phrase(s(X), [a, a, b]).

X = 3

Yes

¿Es posible definir una gramática independiente del contexto que reconozca el mismo lenguaje?

Ejercicio 3: [0.9 puntos]

Responder a las siguientes cuestiones relacionadas con el aprendizaje de conceptos:

- ¿Qué es un concepto?
- ¿Qué es un conjunto de entrenamiento?
- ¿Qué es un espacio de hipótesis? ¿Cuál es el orden de generalidad entre hipótesis?
- ¿Cuál es el objetivo de un algoritmo de aprendizaje?
- Dado un espacio de hipótesis H y un conjunto de entrenamiento D ¿Qué es el espacio de versiones respecto de H y D ? ¿Qué es la cota general y la cota específica?
- Enunciar el teorema de representación del espacio de versiones.

Ejercicio 4: [1.2 puntos]

Considérese el siguiente problema de programación lógica inductiva:

Ejemplos positivos: $q(a,d)$ $q(a,c)$ $q(c,b)$ $q(b,d)$ $q(d,b)$

Ejemplos negativos: $q(a,b)$ $q(c,d)$ $q(d,d)$

Conocimiento base: $h(a)$ $h(b)$ $m(c)$ $m(d)$ $r(a,b)$ $r(c,d)$

Supongamos que el algoritmo FOIL aplicado a este problema se encuentra en el bucle interno, con la regla $q(X,Y) :- h(X).$, parcialmente construida. ¿Incluirá esta regla en el conjunto de reglas a devolver o continuará añadiendo condiciones? Si es así: ¿cuáles son los posibles literales candidatos a ser añadidos y cuál se elegirá? Una vez completada esta regla ¿será la única regla que devuelva el algoritmo como salida? Justificar todas las respuestas.

Apellidos:

Nombre:

Ejercicio 5: [1.4 puntos]

La siguiente tabla muestra ejemplos de situaciones en las que comprar o no un ordenador, en función de su precio (alto, medio o bajo), su procesador (AMD o Intel), si tiene tarjeta ethernet y si el monitor es TFT (se supone que el resto de características es común).

Ej.	Precio	Ethernet	Procesador	TFT	Comprar
E_1	alto	si	amd	si	no
E_2	alto	si	amd	no	no
E_3	alto	si	intel	si	no
E_4	alto	si	intel	no	no
E_5	alto	no	amd	no	no
E_6	alto	no	intel	no	no
E_7	medio	si	amd	si	si
E_8	medio	si	amd	no	si
E_9	medio	no	amd	si	si
E_{10}	medio	no	intel	si	no
E_{11}	medio	no	intel	no	no
E_{12}	bajo	si	amd	si	no
E_{13}	bajo	si	intel	no	si
E_{14}	bajo	no	amd	si	no
E_{15}	bajo	no	amd	no	si
E_{16}	bajo	no	intel	si	no

Aplicar (detallando cada uno de los pasos realizados) el algoritmo ID3 para encontrar, a partir de este conjunto de entrenamiento, un árbol que nos permita decidir sobre la compra de un ordenador. Según el árbol aprendido ¿deberíamos comprar un ordenador con monitor TFT si el precio es bajo? ¿hay algún atributo irrelevante?

Nota: Puede que sean necesarias las siguientes entropías: $Ent[2+, 3-] = 0.971$, $Ent[3+, 5-] = 0.954$, $Ent[2+, 6-] = 0.811$, $Ent[4+, 5-] = 0.991$ y $Ent[1+, 6-] = 0.591$.

Ejercicio 6: [1.4 puntos] Considerar el siguiente conjunto de reglas CLIPS:

```
(defrule regla-1
  ?h1 <- (lista1 $?a ?x $?b)
  ?h2 <- (lista2 $?c ?x $?d)
  =>
  (retract ?h1 ?h2)
  (assert (lista1 ?a ?b))
  (assert (lista2 ?c ?d)))
```

```
(defrule regla-2
  ?h1 <- (lista1 $?a ?x $?b)
  ?h2 <- (lista2 $?)
  (not (lista2 $?c ?x $?d))
  =>
  (retract ?h1 ?h2)
  (printout t "Fallo" crlf))
```

```
(defrule regla-3
  ?h2 <- (lista2 $?a ?x $?b)
  ?h1 <- (lista1 $?)
  (not (lista1 $?c ?x $?d))
  =>
  (retract ?h1 ?h2)
  (printout t "Fallo" crlf))
```

```
(defrule regla-4
  ?h1 <- (lista1)
  ?h2 <- (lista2)
  =>
  (retract ?h1 ?h2)
  (printout t "Exito" crlf))
```

Se pide explicar brevemente el significado del programa CLIPS anterior, y escribir las dos tablas de seguimiento considerando los dos siguientes conjuntos de hechos iniciales, respectivamente:

- a) (deffacts ejemplo-1
 (lista1 a b c)
 (lista2 c b a))
- b) (deffacts ejemplo-2
 (lista1 a a b)
 (lista2 a b))

En las tablas de seguimiento se tienen que incluir TODAS las activaciones y desactivaciones de las reglas, indicando para cada una de ellas los hechos que las activan y el valor que toma (al menos) la variable `?x`. En caso de tener varias activaciones al mismo tiempo, se deja libertad al alumno para que elija la que se dispare (ya que el resultado final es independiente de estas elecciones).