

Apellidos:

Nombre:

Ejercicio 1: [4 puntos]

Contestar las siguientes cuestiones de manera clara y concisa, usando para ello el espacio en blanco que aparece a continuación de cada una de ellas:

- (a) Definir en Prolog un metaintérprete `prueba_xor`, modificando el metaintérprete básico dado en teoría, de manera que en lugar de conjunciones (conectiva `&`) se consideren disyunciones exclusivas (conectiva `xor`), donde `A xor B` se verifica si se verifica `A` y no se verifica `B` o viceversa. Por ejemplo, supóngase dada la siguiente base de conocimiento:

```
a <- verdad.
```

```
x <- verdad.
```

```
c <- a xor b.
```

```
d <- a xor x.
```

Lo que sigue son algunos ejemplos del comportamiento de `prueba_xor`:

```
?- prueba_xor(c).
```

```
Yes
```

```
5 ?- prueba_xor(d).
```

```
No
```

¿Implementa este metaintérprete un razonamiento monótono? Justificar la respuesta con un ejemplo.

(b) Decir si las siguientes afirmaciones son verdaderas o falsas, justificando la respuesta:

- FIND-S siempre encuentra una hipótesis consistente con los ejemplos.
- ID3 siempre encuentra una hipótesis consistente con los ejemplos.
- ID3 siempre encuentra el árbol de decisión de menor número de nodos, consistente con los ejemplos.
- ID3 tiene un sesgo inductivo en el lenguaje.
- Un algoritmo de aprendizaje insesgado es inútil.

(c) Considérese el siguiente problema de programación lógica inductiva:

Ejemplos positivos: $p(a,1)$ $p(b,2)$ $p(3,c)$

Ejemplos negativos: $p(1,1)$ $p(a,b)$

Conocimiento base: $q(a)$ $q(b)$ $q(c)$ $r(1)$ $r(2)$ $r(3)$

¿Es posible que el algoritmo FOIL, aplicado a este problema, devuelva como resultado la siguiente (y única) regla: $p(X,Y) :- q(X),r(Y)$? ¿Y si en el problema quitamos $p(3,c)$ como ejemplo positivo y añadimos $p(c,3)$ como ejemplo negativo? Justificar las respuestas.

(d) Escribir el pseudocódigo del algoritmo de aprendizaje de reglas por recubrimiento.

Apellidos:

Nombre:

Ejercicio 2 [1 punto]

Una entidad bancaria concede un préstamo a un cliente determinado en función de una serie de parámetros: su edad (puede ser joven, mediano o mayor), sus ingresos (altos, medios, bajos), un informe sobre su actividad financiera (que puede ser positivo o negativo) y finalmente en función de que tenga o no a su cargo otro préstamo. La siguiente tabla presenta una serie de ejemplos en los que se especifica la concesión o no del préstamo en función de estos parámetros:

Ej.	Edad	Ingresos	Informe	Otro	Conceder
E_1	joven	altos	negativo	no	no
E_2	joven	altos	negativo	si	no
E_3	mediano	altos	negativo	no	si
E_4	mayor	medios	negativo	no	si
E_5	mayor	bajos	positivo	no	si
E_6	mayor	bajos	positivo	si	no
E_7	mediano	bajos	positivo	si	si
E_8	joven	medios	negativo	no	no
E_9	joven	bajos	positivo	si	si
E_{10}	mayor	medios	positivo	no	si
E_{11}	joven	medios	positivo	si	si
E_{12}	mediano	medios	negativo	si	si
E_{13}	mediano	alto	positivo	no	si
E_{14}	mayor	mediano	negativo	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 concesión de préstamos.

Apellidos:

Nombre:

Ejercicio 3 [1 punto]

Se considera la siguiente base de conocimiento en CLIPS:

```
(defrule r1
  ?h1 <- (secuencia $?i ?x&~r $?m r $?f)
  =>
  (retract ?h1)
  (assert (secuencia $?i r $?m ?x $?f)))

(defrule r2
  ?h1 <- (secuencia $?i m $?m ?x&~m $?f)
  =>
  (retract ?h1)
  (assert (secuencia $?i ?x $?m m $?f)))

(deffacts ejemplo
  (secuencia r m a m r a))
```

Se pide explicar brevemente el significado del programa CLIPS anterior, y escribir una tabla de seguimiento de su ejecución, indicando los hechos que quedan finalmente en memoria. En la tabla de seguimiento se tienen que incluir TODAS las activaciones y desactivaciones de las reglas r1 y r2, indicando para cada una de ellas (al menos) el valor que toman las variables \$?i, ?x, \$?m y \$?f.