APELLIDOS:	PL:
NOMBRE:	DNI:

ESCUELA DE INGENIERÍA INGORMÁTICA

SISTEMAS INTELIGENTES

Examen Final de Teoría. Lunes 23 de Mayo de 2016.

1.- [3 puntos] Aprendizaje

a) [1 punto] Considera el siguiente conjunto de ejemplos utilizado para aprender a identificar si una seta es venenosa o no

Ejemplo	*EsDura*	*EsOlorosa*	*TieneManchas*	*EsRegular*	*EsVenenosa*
A	0	0	0	0	0
B	0	0	1	0	0
C	1	1	0	1	0
D	1	0	0	1	1
E	0	1	1	0	1
F	0	0	1	1	1
G	0	0	0	1	1
H	1	1	0	0	1

Considerando sólo los ejemplos de la A a la H, selecciona la raíz del árbol de decisión cuando se utiliza como métrica el índice de Gini.

H(esVenenosa/EsRegular=0)=1 (porque hay igual numero de ejemplos de cada clase)
*H(esVenenosa/EsRegular=1)=-1/4*log(1/4)-3/4 log(3/4)=0.8133 (Estos mismos cálculos están en las trasparencia 44)*
H(esVenenosa/EsRegular)=1.8133

*H(esVenenosa/TieneManchas=0)= -2/5*log(2/5)-3/5*log(3/5)=0.970*
*H(esVenenosa/TieneManchas=1)= -1/3*log(1/3)-2/3*log(2/3)=0.9183*

*H(esVenenosa/EsOlorosa=0)= -2/5*log(2/5)-3/5*log(3/5)*
*H(esVenenosa/EsOlorosa=1)= -1/3*log(1/3)-2/3*log(2/3)*

*H(esVenenosa/EsDura=0)= -2/5*log(2/5)-3/5*log(3/5)*
*H(esVenenosa/EsDura=1)= -1/3*log(1/3)-2/3*log(2/3)*

H(esVenenosa/EsOlorosa)= H(esVenenosa/EsDura)= H(esVenenosa/EsOlorosa)= 1.8883

Como H(esVenenosa/EsRegular) < H(esVenenosa/EsOlorosa)= H(esVenenosa/EsDura)= H(esVenenosa/EsOlorosa), el atributo que mayor ganancia de información tiene es EsRegular

b) [0,5 puntos] Dado un conjunto de 1500 ejemplos distribuidos en dos clases de modo que 1300 son de la clase “APRUEBA” y “200” de la clase “SUSPENDE”, proporciona RAZONADAMENTE una cota inferior del porcentaje de error que debería tener cualquier clasificador.

La cota inferior del porcentaje de error es 200/1500, que correspondería con asignar todos los ejemplos como de la clase mayoritaria.

c) [0,75 punto] Dada la siguiente matriz de confusión, calcula la precisión y el recall del clasificador utilizado para obtener dicha matriz.

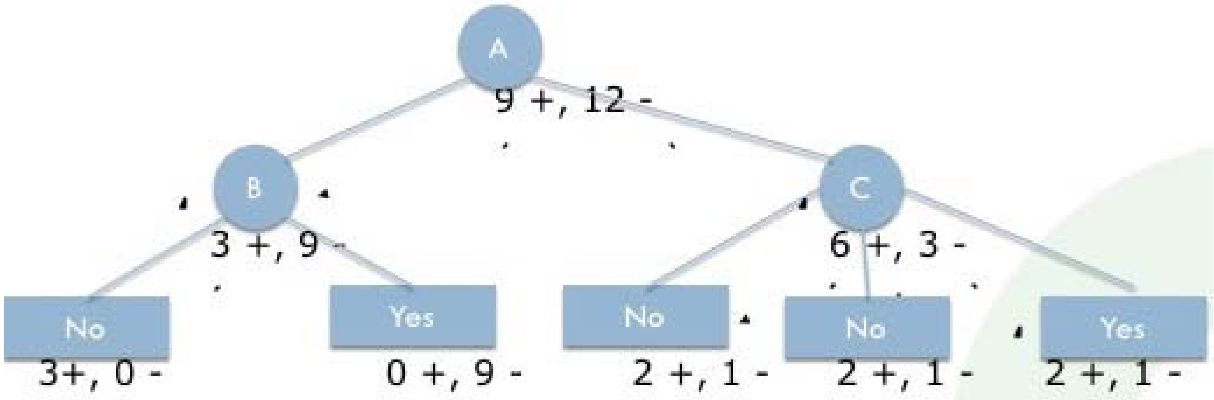
$$
\begin{array}{cc}
 & \begin{array}{ccc} A-predicho & B-predicho & C-predicho \end{array} \\
\begin{array}{c} A-real \\ B-real \\ C-real \end{array} & \begin{pmatrix} 23 & 70 & 1 \\ 4 & 5 & 0 \\ 6 & 2 & 100 \end{pmatrix}
\end{array}
$$

	A	*B*	*C*	*Total*
Precisión=TP/TP+FP	*70%*	*6%*	*99%*	*58%*
Recall=TP/TP+FN	*24,4%*	*55,5%*	*92,6%*	*57,54%*

La clase B tiene 9 ejemplos de test de acuerdo a la matriz de confusión anterior

0.15 por calcular el número de ejemplos de la clase B
0.2 por precisiones
0.2 por recalles
0.2 por P y R del método

d) [0,75 puntos] Dado el siguiente árbol de decisión donde se especifica en cada nodo cuantos ejemplos hay de la clase "+" y cuantos de la clase "-", si aplicáramos un mecanismo de poda, ¿podríamos eliminar alguna rama del árbol obteniendo al menos el mismo porcentaje de acierto? ¿Por qué?



Sí, las tres ramas que salen del nodo C son prescindibles, manteniendo el mismo error, ya que el error cometido al etiquetar el nodo C como de la clase "+" es 3/9=1/3, exactamente el mismo que se comete en las tres hojas que salen de C

---

2 .- a) **[1 punto]** ¿Qué es un heurístico consistente o monótono y qué consecuencias tiene el uso de uno de estos heurísticos en el comportamiento del algoritmo A*?

RESP: Un heurístico h es consistente si para todo par de nodos n1 y n2 del espacio de búsqueda se cumple que $h(n1) \leq h(n2) + k(n1,n2)$, siendo k(n1,n2) el coste del camino mínimo entre n1 y n2. Y es monótono si cumple $h(n1) \leq h(n2) + c(n1,n2)$, siendo c(n1,n2) el coste de la regla que lleva de n1 a n2. Las dos propiedades son equivalentes y tienen interesantes consecuencias para el comportamiento del algoritmo A*:
i.- Si el heurístico h es consistente y A* elige un nodo n para expandir, entonces g(n) = g*(n). Esto significa que en ese momento ya se conoce el camino óptimo desde el inicial a n y en consecuencia no habrá que rectificar este camino posteriormente. Lo que a su vez tiene como consecuencia que no hay que rectificar ni reexpandir nodos que ya fueron expandidos.
ii.- Las condiciones necesaria y suficiente de expansión de nodos se pueden expresar de forma simple como $g^*(n)+h(n) \leq C^*$ y $g^*(n)+h(n) < C^*$, respectivamente.
iii.- La condición de dominancia entre heurísticos se puede expresar como: si h1 y h2 son dos heurísticos monótonos tales que $h1(n) \leq h2(n)$ para todo nodo n, si un nodo n’ es expandido por h2 y no por h1, entonces: h1(n’) = h2(n’) = C*-g*(n’), lo que en general cumplirán solamente unos pocos nodos del espacio de búsqueda. Esto significa que es muy poco probable que haya nodos (al menos muchos) expandidos por h2 que no lo sean por h1.
iv.- La secuencia de los valores de f de los nodos expandidos es no decreciente.

b) Consideremos los conocidos heurísticos h1 (número de fichas descolocadas) y h2 (suma de distancias ortogonales) para el problema del 8-puzzle y los siguientes estados:

inicial:

1	2	3
7	8	5
4		6

n1:

1	2	3
7		5
4	8	6

n2:

1	2	3
7	8	5
	4	6

n3:

1	2	3
7	8	5
4	6	

objetivo:

1	2	3
8		4
7	6	5

b1) **[0,25 puntos]** Calcula los valores de la siguiente tabla:

RESP: Dado que n1, n2 y n3 son los sucesores del inicial, el valor de g* para los tres nodos es 1. Los valores de h1 y h2 se calculan simplemente contando las fichas descolocadas en cada estado y la suma de sus distancias ortogonales a la posición en el objetivo, respectivamente:

	n1	n2	n3
g*	1	1	1
h1	5	5	4
h2	8	6	6
h*	7	6	8

Para calcular los valores de h* de los nodos n2 y n3 hay que resolver primero las cuestión b4.

Sabiendo que para el problema representado por los estados inicial y objetivo anteriores el valor de C* es 7 y sin necesidad de ejecutar el algoritmo A*, responder de forma breve y razonada a las siguientes cuestiones:
b2) **[1 punto]** ¿Qué se puede decir con respecto a la expansión de los nodos n1, n2 y n3, con cada uno de los heurísticos h1 y h2?
RESP: Dado que los dos heurísticos son consistentes, las condiciones necesaria y suficiente de expansión se expresan como $g^*(n)+h(n) \leq C^*$ y $g^*(n)+h(n) < C^*$, respectivamente. Dado que C* = h*(inicial) = 7, de los cálculos de la tabla siguiente:

	n1	n2	n3
g*+h1	6	6	5
g*+h2	9	7	7

Se deduce que los tres nodos se expandirán con toda seguridad con h1, que el nodo n1 no se expandirá con h2, y que los nodos n2 y n3 podrán expandirse o no con h2.

b3) **[0,5 puntos]** ¿Podemos asegurar que alguno de los nodos n1, n2 y n3 no forma parte de la solución óptima?

RESP: Si un nodo está en un camino óptimo desde el inicial hasta el objetivo, para cualquier h admisible, debe cumplirse que $g^*(n)+h(n) \leq C^*$, ya que $g^*(n)+h^*(n)=C^*$ y $h(n) \leq C^*$. Dado que para el nodo n2 $g^*(n2)+h2(n2) = 1 + 8 = 9 > C^* = 7$, es claro que n2 no está en un camino óptimo desde el inicial al objetivo.
Los nodos n2 y n3 podrían estar en un camino óptimo. De hecho uno de ellos tiene que estar necesariamente porque n1, n2 y n3 son los tres sucesores del inicial y el camino óptimo tiene que pasar por alguno de ellos.

b4) **[0,25 puntos]** Observa la estructura del nodo n3 y compárala con la estructura del nodo objetivo, ¿se te ocurre algún argumento para justificar que $h^*(n3) > h2(n3)$?

RESP: Esta pregunta es un poco más sutil que las anteriores y requiere un razonamiento que tiene que ver con la estructura del problema. Fijémonos en la posición del 4 en el nodo n3. El heurístico h2 supone que tendremos que moverlo tres veces para colocarlo en su posición en el objetivo. Pero esto es imposible sin mover la ficha 6 o sin mover la ficha 7 más de una vez. Si movemos el 6 habría que hacerlo por lo menos 2 veces para dejarla otra vez en su posición. Y si movemos el 7, habría que moverla al menos tres veces para llevarla a su posición en el objetivo. Luego podemos decir que h2 está subestimando el valor de h* en al menos dos movimientos para el nodo n3, luego $h^*(n3) \geq h2(n3)+2= 8$.
El razonamiento anterior nos permite además asegurar que n3 no está en un camino óptimo del inicial al objetivo y en consecuencia el que sí lo estará es n2, por lo tanto $h^*(n2)=6$.
Una solución posible de n3 consiste en volver al inicial y luego ir a través de n2. Esta solución tiene un coste de $1 + 7 = 8$, luego $h^*(n3)$ no puede ser mayor que 8 y es por lo tanto igual a 8. Con esto se completa la tabla de la pregunta b1.

3.- [2 puntos] Representación del conocimiento

a. **[1 punto]** Dado el siguiente código CLIPS, completa las instrucciones que se piden y completa los espacios.

```
(deffacts listas
(lista platanos leche tomates manzanas)
(lista azucar huevos leche tomates patatas)
(lista azucar leche tomates patatas peras manzanas)
(lista peras leche tomates)
)
;Qué hace este código?
Define la BH iniciales[0.1]
;Definimos la regla lista-compra
(defrule lista-compra
(lista ? leche tomates $?) =>)
; ¿Si ejecutamos el comando(run), que sucede?
Nada pues no se han cargado los hechos iniciales. [0.2]
; Comando para cargar los hechos iniciales:
[0.1] (reset)
; Una vez cargados los hechos iniciales, si tecleamos el comando
[0.1] (facts), CLIPS nos muestra
f-0 (initial-fact)
f-1 (lista platanos leche tomates manzanas)
f-2 (lista azucar huevos leche tomates patatas)
f-3 (lista azucar leche tomates patatas peras manzanas)
f-4 (lista peras leche tomates)
For a total of 5 facts.
;Indica las activaciones de la regla lista-compra guardadas en
[0.1] la agenda y que se produjeron al cargar los hechos iniciales.
[0.2]
lista-compra:f-4
lista-compra:f-3
lista-compra:f-1
Teniendo en cuenta que CLIPS emplea por defecto la estrategia de
resolución de conflictos [0.1] depth, la instancia de la regla que
se dispararía al ejecutar el comando (run 1) es la activada por
el hecho f-4 [0.1]
```

b. **[1 punto]** Dado un SE que identifica las relaciones familiares elementales, en base a unos datos de entrada expresados con las relaciones progenitor_de y sexo en un entorno CLIPS. Se pide implementar las reglas que te permitan identificar las relaciones familiares abuelo y nieta. Si lo necesitas puedes implementar reglas que identifiquen alguna relación intermedia que te facilite la tarea.

```
(deffacts familia
(progenitor_de antonio luisa)
(progenitor_de antonio jaime)
(progenitor_de  luisa juan)
(progenitor_de  luisa ana)
(progenitor_de  luisa patricia)
(progenitor_de  jaime luis)
(progenitor_de  luis marta)
(progenitor_de  luis isabel)
)
;La relación
;(progenitor_de antonio luisa)
;debes interpretarla así:
;antonio progenitor_de luisa
```

```
(deffacts sexo
 (sexo  antonio masculino)
 (sexo  luisa femenino)
 (sexo  jaime masculino)
 (sexo  juan masculino)
 (sexo  ana femenino)
 (sexo  patricia femenino)
 (sexo luis masculino)
 (sexo marta femenino)
 (sexo  isabel femenino)
)
```

RESP:

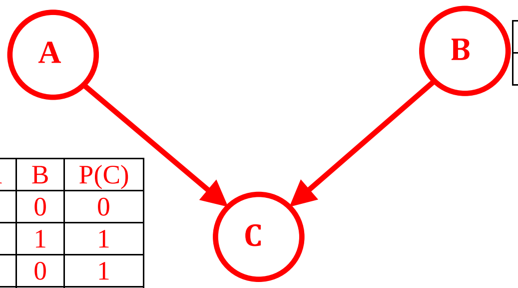
```
; Relación abuelo ----------------
(defrule abuelo
     (progenitor_de ?uno ?dos)
     (progenitor_de ?dos ?tres)
     (sexo ?uno masculino)
=>
      (assert (abuelo ?uno ?tres))
      (printout t ?uno " es el abuelo de " ?tres crlf)
)

; Relación nieta ----------------
(defrule nieta
      (progenitor_de ?uno ?dos)
      (progenitor_de ?dos ?tres)
      (sexo ?tres femenino)
=>
      (assert (nieta ?tres ?uno))
      (printout t ?tres " es la nieta de " ?uno crlf)
)
```

4.- [2 puntos] Razonamiento con incertidumbre

a) **[0,5 puntos]** Considera el siguiente problema: Tenemos tres variables booleanas A, B y C que pueden valer 0 o 1, y se cumple la siguiente relación entre las variables: C=XOR(A, B) (es decir, que C=(A+B) mod 2). A priori, se eligen aleatoriamente los valores para A y para B. Dibuja una red bayesiana que represente este problema, con sus correspondientes tablas de probabilidad condicionada.

Solución:



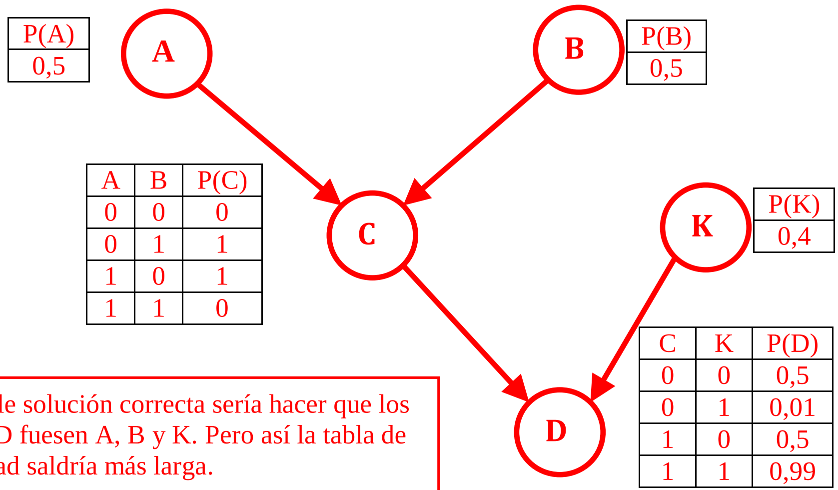
P(A)
0,5

P(B)
0,5

A	B	P(C)
0	0	0
0	1	1
1	0	1
1	1	0

b) **[0,9 puntos]** Ampliación del apartado anterior: estamos enseñándole informática a un mono y queremos comprobar si se sabe la función XOR. Para ello, le damos al mono los valores de A y de B y el mono responde cuál piensa que es el resultado de XOR(A, B). La respuesta del mono es la variable D. Lo ideal sería que D fuese igual que C. Pero puede ser diferente, o bien porque el mono no se sepa la función XOR, o bien porque se la sabe pero se ha equivocado al pulsar el botón. La variable K=1 indica que el mono conoce la función XOR, y si no la conoce es K=0. Si el mono conoce la función XOR, responderá correctamente el 99% de las veces. Por el contrario, si no la conoce, el mono responderá de forma aleatoria. Supón que, a priori, la probabilidad de que el mono conozca la función XOR es de 0,4. Amplia la red bayesiana del apartado anterior (con sus correspondientes tablas de probabilidad condicionada) para representar este problema. Si para alguna tabla necesitas alguna probabilidad no especificada en este enunciado, puedes proponer cualquier valor que consideres razonable.

Solución:



P(A)
0,5

P(B)
0,5

A	B	P(C)
0	0	0
0	1	1
1	0	1
1	1	0

P(K)
0,4

C	K	P(D)
0	0	0,5
0	1	0,01
1	0	0,5
1	1	0,99

Otra posible solución correcta sería hacer que los padres de D fuesen A, B y K. Pero así la tabla de probabilidad saldría más larga.

c) **[0,6 puntos]** En una determinada red bayesiana compuesta por 5 variables A, B, C, D y E, queremos calcular la probabilidad P(A, ¬B | C) mediante el método del muestreo estocástico estándar. Para ello, hemos realizado 8 muestras aleatorias, cuyos valores resumimos en la siguiente tabla. Indica cuál sería el resultado final de dicha probabilidad, utilizando únicamente las muestras realizadas.

A	**B**	**C**	**D**	**E**
V	V	F	F	V
V	F	V	F	V
V	F	F	F	F
V	V	F	V	V
F	V	F	F	V
V	V	V	F	F
F	F	F	V	F
F	V	V	F	V

Solución:

Sabemos que P(A, ¬B | C) = P(A, ¬B, C) / P(C). En el muestreo estocástico estándar, simplemente se trataría de contar cuántas muestras cumplen A y ¬B y C (hay 1 que cumple eso), y dividir ese número por el número de muestras que cumplen C (hay 3 que lo cumplen). Por tanto, el resultado final sería 1/3.