APELLIDOS:	PL:
NOMBRE:	DNI:

ESCUELA DE INGENIERÍA INGORMÁTICA

SISTEMAS INTELIGENTES

Examen Final de Teoría. Lunes 19 de Diciembre de 2016.

1.- [2,75 puntos] Consideremos el siguiente heurístico, h_5, para el problema del 8-puzzle que estima el coste desde un estado n al objetivo de la siguiente forma: por cada ficha que no está en la fila que le corresponde en el objetivo suma 1 y por cada ficha que no está en la columna que le corresponde en el objetivo suma 1

Se pide contestar con precisión y de forma razonada a las siguientes cuestiones:

a)[0,25] ¿Cuál es el valor de los heurísticos h*, h_1 (número de fichas descolocadas), h_2 (suma de distancias ortogonales) y h_5 para los estados n_1, n_2 y n_3 siguientes?

n_1=

1	2	3
7	5	
6	4	8

n_2 =

1	2	3
7	5	8
6	4	

n_3=

1	2	3
7	8	
6	5	4

n_4=

1	2	3
	8	4
7	6	5

objetivo =

1	2	3
8		4
7	6	5

Indicar los valores en la siguiente tabla:

	n_1	n_2	n_3	n_4
h*	9	8	5	1
h_2	9	8	5	1
h_5	8	7	5	1
h_1	5	5	5	1

b)[0,25] ¿Está bien definido el heurístico h_5?

RESP: Sí, está bien definido ya que en el objetivo todas las fichas están en la fila y columna que les corresponde y por lo tanto h_5(objetivo)=0, y para cualquier otro estado n, $h_5(n) \geq 0$, en realidad es mayor que 0.

c)[0,5] ¿Es monótono el heurístico h_5?

RESP: Un heurístico es monótono si para todo par de nodos n y n', tales que n' es alcanzable desde n con un movimiento simple, se cumple que: $h(n) \leq h(n') + c(n,n')$. En este caso $c(n,n')=1$, y por lo tanto debe cumplirse $h(n) \leq h(n') + 1$. Al pasar de n a n' se mueve una ficha y ocurrirá siempre una de estas situaciones: si se mueve en horizontal y en n la ficha no estaba en la columna que le corresponde en el objetivo, en n' puede estar o no en esa columna; si estaba en la columna que le corresponde en el objetivo en n, entonces en n' no lo estará, es decir que si se mueve en horizontal como no se cambia de fila, tendremos que $h_5(n') = h_5(n)$, o bien $h_5(n') = h_5(n)+1$, o bien $h_5(n') = h_5(n)-1$. El mismo razonamiento se puede hacer si se mueve en vertical. En consecuencia, siempre se cumplirá que $h_5(n) \leq h_5(n') + 1$, luego el heurístico es monótono.

d)[0,25] ¿Es admisible el heurístico h_5?

RESP: Sí, ya que todo heurístico monótono es admisible.

e)[0,75] ¿Qué se puede decir sobre la comparación de h_5 con los heurísticos h_1 y h_2 con respecto al coste computacional y al grado de información?

RESP: El coste computacional de los tres heurísticos es el mismo. En los tres casos hay que recorrer las posiciones del estado actual y hacer una comprobación que se hace en tiempo constante.
En cuanto al grado de información, es claro que para todo nodo n se cumple: $h_1(n) \leq h_5(n) \leq h_2(n)$. Con lo que cabe esperar que en general h5 expanda menos nodos que h1 y más que h2. Como los tres heurísticos son monótonos, los nodos que pueden ser expandidos por uno más informado y no por otro menos informado son realmente pocos. Por ejemplo, los que pueden ser expandidos por h2 y no por h5 deben cumplir $h_2(n) = h_5(n) = C^*-g^*(n)$. Esta condición depende de cada problema concreto.

f)[0,75] Suponiendo que el primero de los estados del apartado a) es el estado inicial, ¿Cuál es el valor de C*?, y, de acuerdo con las propiedades de los heurísticos, ¿qué se puede decir sobre la expansión de los otros dos estados, n_2 y n_4, con los heurísticos h_2 y h_5 si se aplica el algoritmo A* para resolver este problema? ¿Es posible que alguno de estos nodos sea expandido con h_2 y no con h_5?

RESP: $C^* = h^*(n1) = 9$. Como los heurísticos h_2 y h_5 son monótonos las condiciones necesaria y suficiente de expansión se expresan así: $g^*(n) + h(n) \leq C$ y $g^*(n) + h(n) < C$, respectivamente. Por otra parte, n2 es un sucesor del inicial (n_2) y n_3 es sucesor de n_2 (distinto de n_1). Luego:

$g^*(n_2) + h_2(n_2) = 1 + 8 = 9 = C^*$ (cumple la necesaria pero no la suficiente, luego n_2 puede expandirse o no con h_2)

$g^*(n_4) + h_2(n_4) = 8 + 1 = 9 = C^*$ (cumple la necesaria pero no suficiente luego n_4 puede expandirse o no con h_2)

$g^*(n_2) + h_5(n_2) = 1 + 7 = 8 < C^*$ (cumple la suficiente luego n_3 se expandirá con seguridad con h_5)

$g^*(n_4) + h_5(n_4) = 8 + 1 = 9 = C^*$ (cumple la necesaria pero no suficiente luego n_4 puede expandirse o no con h_5).

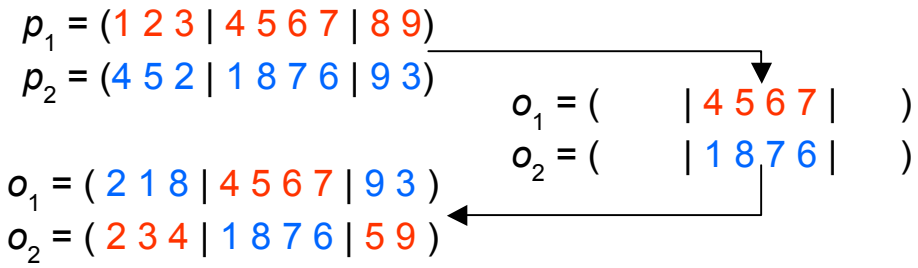
Además, en el caso del nodo n_4, se cumple que $h_2(n_4) = h_5(n_4) = C^* - g^*(n_4)$, luego de acuerdo con la condición de dominancia amplia que se enuncia como:

- si dos heurísticos son monótonos y uno es mayor o igual que el otro para todo nodo n, tal y como ocurre con h_2 y h_5, entonces todo nodo expandido por el que es mayor o igual también lo será por el otro, salvo quizás algún nodo para el que los dos heurísticos deben tomar el mismo valor y que además este valor sea igual a la diferencia de C* y g* para ese nodo.

El nodo n_4 podría ser expandido por h_2 y no por h_5.

2.- [0,5 puntos] Explica un posible operador de cruce para un algoritmo genético cuyos individuos están codificados mediante permutaciones (por ejemplo para el problema del viajante de comercio).

Solución: Un posible ejemplo es el cruce denominado OX, que extrae una subsecuencia aleatoria de un padre y la coloca en el hijo en la misma posición, y los restantes elementos se insertan en las restantes posiciones, manteniendo el orden relativo que tienen en el otro padre. Para generar un segundo hijo los padres invierten su rol. Por ejemplo, a partir de dos padres p_1 y p_2 podríamos obtener los siguientes hijos o_1 y o_2:



3.- [2,75 puntos] Contesta las siguientes cuestiones relacionadas con Sistemas de Representación de Conocimiento:

a) [0,4 puntos] En un Sistema Experto basado en Lógica de Predicados, se dispone del siguiente conocimiento ya formalizado en Forma Clausal:

$$\{ \sim p(a),\ p(x) \vee q(f(x)) \vee \sim r(x,y),\ r(x,f(a)) \}$$

Se desea preguntar al sistema experto si existe algún par de elementos x, y tal que se cumpla q(x) y r(y, x). Es decir, ¿ $\exists x\ \exists y\ (q(x) \wedge r(y,x))$?

Muestra el conjunto de cláusulas en notación Kowalski que serían necesarias para aplicar el proceso de resolución utilizando el **<u>método de Green</u>**. Si aplicamos resolución al conjunto de cláusulas anterior, ¿qué deberíamos obtener para saber si la pregunta es afirmativa? *NOTA: No es necesario aplicar resolución, únicamente resolver las cuestiones indicadas*.

Solución:

Dado que ya tenemos la base de conocimiento en forma clausal, se puede pasar directamente a notación Kowalski:

$\leftarrow p(a)$

$p(x), q(f(x)) \leftarrow r(x,y)$

$r(x, f(a)) \leftarrow$

Para aplicar resolución, además de las cláusulas anteriores, necesitamos la negación de la pregunta en Forma Clausal:

Paso a FNS: $\neg\left(\exists x\ \exists y\ \left(q(x)\ \wedge r(y,x)\right)\right) \Leftrightarrow \forall x\ \forall y\ \left(\neg q(x) \vee \neg r(y,x)\right)$

FC: $\neg q(x) \vee \neg r(y,x)$

De aquí obtendríamos la siguiente cláusula en notación Kowalski:

$\leftarrow q(x), r(y,x)$

Ahora bien, el método de Green se basa en introducir un predicado artificial en la conclusión de forma que la respuesta a nuestra pregunta, en caso de ser afirmativa, quede almacenada en dicho predicado. Es el método que utiliza por defecto aima-java. En este caso, dado que la pregunta involucra dos variables, podemos hacer un predicado artificial R(x,y), con lo que **la solución de este ejercicio quedaría como sigue**:

$\leftarrow p(a)$

$p(x), q(f(x)) \leftarrow r(x,y)$

$r(x, f(a)) \leftarrow$

$\mathbf{R(x,y)} \leftarrow q(x) \vee r(y,x)$

Al utilizar el método de Green, nuestro objetivo al aplicar resolución ya no es alcanzar la cláusula vacía, sino la siguiente cláusula: $R(x,y) \leftarrow$, dónde x e y contendrán la respuesta a nuestra pregunta.

b) [0,85 puntos] En un determinado país se dispone de un sistema de lotería en el cuál se celebra un sorteo cada mes en el que se extraen 5 bolas. Se dispone de un pequeño SE que almacena los resultados de los últimos sorteos de la siguiente forma:

```
(deftemplate sorteo
        (slot mes (type SYMBOL))
        (slot anyo (type INTEGER) (range 2000 ?VARIABLE))
        (multislot combinacion)
)

(deffacts historico
  (numeroSuerte 30)
  (sorteo (mes Septiembre) (anyo 2016) (combinacion 44 2 30 47 29))
  (sorteo (mes Mayo) (anyo 2016) (combinacion 40 17 24 32 8))
  (sorteo (mes Enero) (anyo 2016) (combinacion 30 26 3 33 12))
  (sorteo (mes Diciembre) (anyo 2015) (combinacion 26 14 46 10 30))
  (sorteo (mes Noviembre) (anyo 2015) (combinacion 31 7 2 49 23))
)
```

Resuelve las siguientes cuestiones:

a. [0,15] Identifica qué tipo de hechos aparecen en el código anterior.

Solución: Los hechos tipo "sorteo" son hechos **no ordenados**, mientras que "numeroSuerte" es un hecho **ordenado**.

b. [0,35] Dada la siguiente regla:

```
(defrule ganadoras
        (sorteo (mes ?m) (anyo ?a) (combinacion ?n $?))
        (numeroSuerte ?n)
        =>
        (printout t ?m " de " ?a))
```

Indica qué hace la regla y qué hechos la activan.

Solución: La regla muestra el mes y año en los que el primer número de la combinación ganadora coincide con "numeroSuerte. La regla es activada únicamente por el hecho correspondiente al sorteo de Enero de 2016, junto con el hecho "numeroSuerte".

c. [0,35] ¿y si en vez de (combinación ?n $?) aparece (combinación $? ?n $?)?

Solución: En este caso la regla muestra el mes y año en los que la combinación ganadora contiene el número "numeroSuerte" en cualquier posición. La regla es activada por los hechos correspondientes a los sorteos de Septiembre y Enero de 2016, y Diciembre de 2015, junto con el hecho "numeroSuerte".

c) [1,25 puntos] Se pretende diseñar un SE en el que dados todos los ingredientes que tengo en mi casa, me diga qué puedo preparar para cenar esta noche. Para ello se dispone de los siguientes hechos iniciales:

```
(deffacts armario
        (atun gramos 750)
        (cebolla unidades 4)
        (harina gramos 1000)
        (levadura gramos 100)
)
```

```
(deffacts nevera
        (huevo unidades 12)
        (pimientoVerde unidades 1.0)
        (pimientoRojo unidades 1.0)
        (tomate gramos 500)
)
```

```
(deffacts otros
        (sal)
        (agua)
)
```

En el caso del conjunto de hechos "otros", se interpreta que se dispone de una cantidad infinita de esos ingredientes.
Diseñar un sistema de reglas en CLIPS que permita averiguar si puedo hacer una empanada. Para ello han de generarse dos productos intermedios y luego juntarlos para hacer la empanada. La receta es la siguiente:

- Para hacer una empanada necesitamos generar primero otros dos productos: relleno de empanada y masa
- Para el relleno se necesitan los siguientes ingredientes: 600 gramos de atún, 200 gramos de tomate, 3 cebollas, medio pimiento rojo y sal
- Para la masa se necesitan los siguientes ingredientes: 750 gramos de harina, 40 gramos de levadura, sal, agua y un huevo.

NOTA: Téngase en cuenta que si utilizo por ejemplo 750 gramos de harina para hacer la empanada, sólo me quedarán 250 para preparar otro plato.

Solución:

```
; Regla para el relleno de la empanada: Debemos asegurarnos de que disponemos de las cantidades necesarias de cada ingrediente y en caso de ser así, crear el relleno y actualizar la cantidad de ingredientes restantes:
(defrule rellenoEmpanada
        ?at <- (atun gramos ?ga&:(>= ?ga 600))
        ?to <- (tomate gramos ?gt&:(>= ?gt 200))
        ?ce <- (cebolla unidades ?uc&:(>= ?uc 3))
        ?pr <- (pimientoRojo unidades ?up&:(>= ?up 0.5))
        (sal)
        (not (relleno))          ; Opcional
        =>
        (retract ?at)
        (assert (atun gramos (- ?ga 600)))
        (retract ?to)
        (assert (tomate gramos (- ?gt 200)))
        (retract ?ce)
        (assert (cebollas unidades (- ?uc 3)))
        (retract ?pr)
        (assert (pimientoRojo unidades (- ?up 0.5)))
        (assert (relleno) )
)

; Igual que el anterior
(defrule masaEmpanada
        ?ha <- (harina gramos ?gh&:(>= ?gh 750))
        ?le <- (levadura gramos ?gl&:(>= ?gl 40))
        ?hu <- (huevo unidades ?uh&:(>= ?uh 1))
        (agua)
        (sal)
        (not (masa))             ; Opcional
        =>
        (retract ?ha)
        (assert (harina gramos (- ?gh 750)))
        (retract ?le)
```

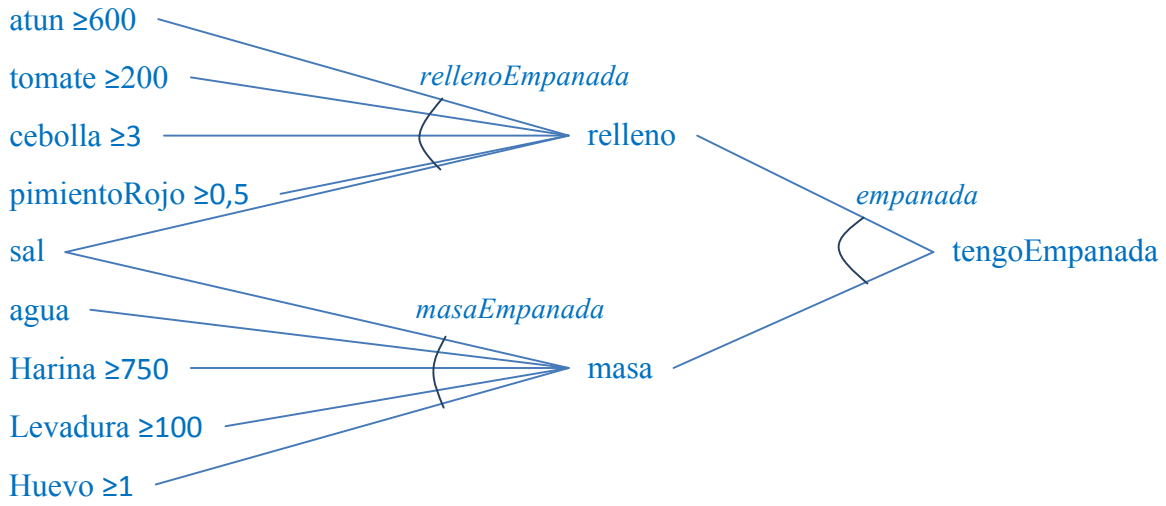
```
            (assert (levadura gramos (- ?gl 40)))
            (retract ?hu)
            (assert (huevo unidades (- ?uh 1)))
            (assert (masa) )
    )

    ; Regla para crear la empanada
    (defrule empanada
            ?r <- (relleno)
            ?m <- (masa)
            =>
            (retract ?r)            ; Opcional
            (retract ?m)            ; Opcional
            (assert (tengoEmpanada))
    )
```

NOTA: La instrucción modify no se puede utilizar con hechos no ordenados. Es por eso que tenemos que eliminar el hecho y generarlo de nuevo.

d) [0,25 puntos] Dibuja el grafo del problema anterior

Solución:



4.- [3 puntos] a) [1.25 punto] Dado un conjunto de entrenamiento con 150 ejemplos y tres variables, la variable Color que puede tomar los valores Rojo, Azul o Amarillo, la variable Tamaño que toma los valores Grande o Pequeño y la variable a predecir, COMPRA; que toma los valores SÍ o NO. Suponiendo que :

- Hay 50 ejemplos donde Color=Rojo, 50 para los que Color=Azul y 50 para los que Color=Amarillo.
- Hay 20 ejemplos para los que Tamaño=Grande y 130 para los que Tamaño=Pequeño.
- Hay 100 ejemplos de la clase COMPRA= SI y 50 de la clase COMPRA=NO
- IG(Color)=IG(Tamaño)

Contesta razonadamente:
¿Qué variable selecciona como raíz el método C4.5?

Dado que IG(Color)=IG(Tamaño) tendrá mayor GR aquella variable con menor Split Info. Por tanto lo único que hay que hacer es calcular la Split Info de las variables Color y Tamaño.

En el caso de la variable COLOR la Split Info es 1,58 y en el caso de Tamaño 0,56, siendo esta variable la que obtiene un mayor GR y por tanto será la variable seleccionada por C4.5

¿Es el 70% un porcentaje de acierto aceptable para el método C4.5?

Es aceptable puesto que el porcentaje de acierto del azar se sitúa en torno al 66,7%.

b) [0.5 puntos] Considera las variables Altura y Tipo, donde Tipo es la variable a predecir y Altura es una variable continua utilizada para predecir el Tipo. Dados los siguientes 6 ejemplos {Ej1, Ej2, …., Ej6}

	Ej1	Ej2	Ej3	Ej4	Ej5	Ej6
Altura	20	25	35	50	80	150
Tipo	Árbol	Árbol	Arbusto	Arbusto	Árbol	Árbol

¿Cuántos Split Points puedes obtener para la variable Altura? Calcula dichos Split Points.

Hay un número indeterminado de Split Points de 2 tipos, cualquier valor entre 25 y 35 y cualquier valor entre 50 y 80.

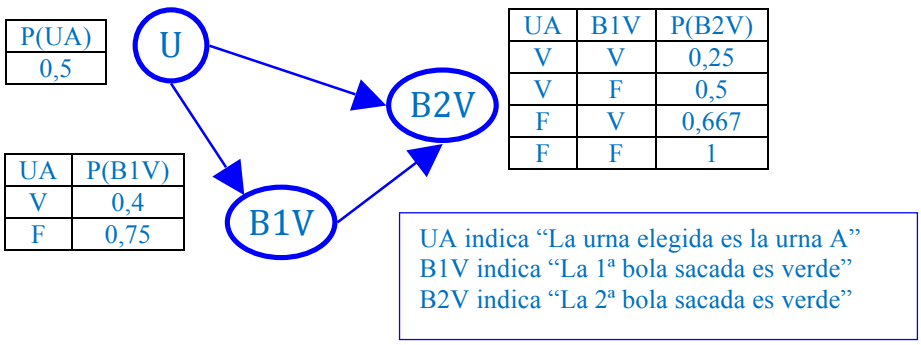
c) [0.75] Proporciona una matriz de confusión para el problema de predecir si la inversión de una determinada empresa debe ser "alta", "baja" o "media" de modo que el Recall (Cobertura) de la clase "alta" sea del 75% y la el número de Falsos Positivos (FP) de la clase "baja" sea 50. NOTA: En la matriz de confusión no puede haber 0's.

No existe una única solución para este ejercicio, de hecho hay infinitas. Sin embargo, todas aquellas matrices de confusión donde no se especificaba si lo predicho (y consecuentemente la situación real) se situaban en filas o en columnas, no se han valorado. Tampoco aquellas matrices de confusión con al menos un número no natural.

5.- [1 punto] Tenemos dos urnas: en la urna A hay dos bolas verdes y tres bolas azules. En la urna B hay tres bolas verdes y una bola azul. Alguien elige una urna al azar, saca una bola al azar y resulta que es verde. Luego saca otra bola (sin devolver a la urna la primera bola que sacó) y resulta que también es verde. ¿Cuál es la probabilidad de que sea la urna A?

Notas: Para resolver el ejercicio, construye la red bayesiana asociada al problema y calcula la inferencia sobre ella. Hacer bien la red bayesiana cuenta 0,5 puntos y hacer bien los cálculos correspondientes cuenta 0,5 puntos. Si no dispones de calculadora puedes dejar los cálculos indicados (por ejemplo 0,25 + 0,5 / 0,25 + 0,8).

Solución:



$$P(UA|B1V,B2V) = \frac{P(UA,B1V,B2V)}{P(B1V,B2V)} = \frac{P(UA,B1V,B2V)}{P(UA,B1V,B2V) + P(\sim UA,B1V,B2V)} =$$

$$= \frac{0{,}05}{0{,}05 + 0{,}25} = \mathbf{0{,}167}$$

$$P(UA, B1V, B2V) = P(UA) * P(B1V|UA) * P(B2V|UA, B1V) = 0{,}5 * 0{,}4 * 0{,}25 = 0{,}05$$

$$P(\sim UA, B1V, B2V) = P(\sim UA) * P(B1V|\sim UA) * P(B2V|\sim UA, B1V) = 0{,}5 * 0{,}75 * 0{,}667 = 0{,}25$$