APELLIDOS:	PL:
NOMBRE:	DNI:

ESCUELA DE INGENIERÍA INGORMÁTICA

SISTEMAS INTELIGENTES

Convocatoria Extraordinaria. Lunes 27 de mayo de 2019.

I.- Búsqueda (3,5 puntos)

Sobre el problema del 8-puzzle y el algoritmo A.*
Consideremos un heurístico h(n) tal que para cada estado suma 1 por cada ficha que no está en la columna que le corresponde en el objetivo y suma 1 también por cada ficha que no está en su fila en el objetivo.

a) **[1,25 puntos]** Indicar de forma razonada si h está bien definido y cuáles son sus propiedades.
RESP: Está bien definido ya que si todas las fichas de n están colocadas también están en sus filas y columnas, y por lo tanto h(n)=0. Si alguna ficha está descolocada, entonces no está en su fila o no está en su columna, luego h(n)>0. Luego h(n)>=0 para todos los estados, y en particular h(objetivo)=0.
Además, h es admisible ya que para llevar una ficha que no está en su fila (columna) a la posición que le corresponde en el objetivo hay que moverla al menos una vez; y si no está ni en su fila ni en su columna habrá que moverla al menos 2 veces.
También es consistente (o monótono que en la práctica es lo mismo), monótono ya que al pasar de n1 a n2 con un movimiento de una ficha, tendremos que:
- h(n2) = h(n1)+1, si se hace un movimiento horizontal (vertical) y la ficha estaba en su columna (fila) y por lo tanto en n2 la ficha está fuera de su columna (fila)
- h(n2) = h(n1) si la ficha estaba fuera de su fila (columna) y en n2 sigue estándolo
- h(n2) = h(n1)-1 si la ficha estaba fuera de su fila (columna) y en n2 pasa a estar en su fila (columna).

b) **[0,75 puntos]** Comparar el heurístico h con los clásicos h1 y h2 en términos de calidad de la solución y nodos expandidos.
RESP: Como los tres heurísticos son admisibles, todos encontrarán la solución óptima. Además, se cumple que, para todo n, h1(n)<=h(n)<=h2(n). En el primer caso es claro ya que cada ficha que no está en su fila o columna, tampoco está en su posición en el objetivo. En el segundo, si una ficha no está en su fila, o no está en su columna, entonces está al menos a distancia ortogonal 1 de su posición en el objetivo, y si no está ni en su fila ni en su columna, entonces está al menos a distancia ortogonal 2.
Como los tres heurísticos son monótonos, por el hecho de tener h1(n)<=h(n)<=h2(n), se cumple que h domina ampliamente a h1, y h2 domina ampliamente a h. En consecuencia, es muy poco probable que un nodo expandido por h1 sea expandido por h. Solo podría darse el caso en nodos que cumplan h1(n)=h(n)=C*-g*(n). Obviamente, la cantidad de nodos que cumplen estas condiciones son muy pocos en términos relativos. Análogamente, si consideramos h2 y h.
De modo que lo que cabe esperar, para problemas no triviales, es que h expanda menos nodos que h1 pero más que h2 para llegar a una solución.

c) **[0,5 puntos]** Indicar los valores de h, h1 y h2 para el siguiente estado y sus sucesores.

$$\begin{bmatrix} 7 & 3 & 1 \\ 6 & & 4 \\ 2 & 8 & 5 \end{bmatrix}$$

RESP:

$$\begin{bmatrix} 7 & 3 & 1 \\ 6 & & 4 \\ 2 & 8 & 5 \end{bmatrix} \begin{matrix} h1 = 6 \\ h = 9 \\ h2 = 12 \end{matrix}$$

$$\begin{bmatrix} 7 & & 1 \\ 6 & 3 & 4 \\ 2 & 8 & 5 \end{bmatrix} \begin{matrix} h1 = 6 \\ h = 10 \\ h2 = 13 \end{matrix} \begin{bmatrix} 7 & 3 & 1 \\ & 6 & 4 \\ 2 & 8 & 5 \end{bmatrix} \begin{matrix} h1 = 6 \\ h = 8 \\ h2 = 11 \end{matrix} \begin{bmatrix} 7 & 3 & 1 \\ 6 & 4 & \\ 2 & 8 & 5 \end{bmatrix} \begin{matrix} h1 = 7 \\ h = 10 \\ h2 = 13 \end{matrix} \begin{bmatrix} 7 & 3 & 1 \\ 6 & 8 & 4 \\ 2 & & 5 \end{bmatrix} \begin{matrix} h1 = 6 \\ h = 8 \\ h2 = 11 \end{matrix}$$

Sobre algoritmos evolutivos.

Consideremos un algoritmo genético que resuelve el problema del viajante de comercio (TSP) y que utiliza el clásico esquema de codificación basado en permutaciones (interpretando que cada elemento de la permutación representa a una de las ciudades, incluida la ciudad de partida). Consideremos también los dos operadores de cruce siguientes:

OX1. El clásico de cruce en dos puntos que hace lo siguiente: se eligen dos posiciones p1 y p2 (p1 < p2) aleatorias y luego se extrae la subsecuencia entre p1 y p2 del primer padre y se coloca en el hijo en la misma posición. El resto de las ciudades se insertan en las posiciones del hijo que están libres, de izquierda a derecha y manteniendo el orden relativo que tienen en el segundo padre.

OX2. Se recorre el primer padre y cada elemento se copia al hijo en la misma posición con una probabilidad de 0,5. El resto de posiciones del hijo se rellena como en el caso anterior.

a) **[0,5 puntos]** Indicar un posible resultado de los dos operadores anteriores al cruzar los siguientes cromosomas para un problema con 10 ciudades

(9 7 6 4 0 2 3 1 5 8)

(7 4 2 0 8 9 1 5 6 3)

RESP: (OX1) Si las posiciones van de la 0 a la 9 y si, por ejemplo, p1 = 3 y p2 = 5, el hijo se construye en dos pasos: en el primero tendremos que los elemento 4,0 y 2 quedan en el hijo en las mismas posiciones que en el primer papre

(_ _ _4 0 2 _ _ _ _)

Las ciudades 4, 0 y 2 se eliminan del segundo padre

(7 _ _ _ 8 9 1 5 6 3)

Y los elementos que quedan se insertan en el hijo manteniendo el orden relativo que tenían en el segundo padre

(7 8 9 4 0 2 1 5 6 3)

(OX2) En este caso, supongamos que en el recorrido se seleccionan las siguientes posiciones del primer hijo

(9 _ _ 4 _ 2 3 _ _ 8)

Como antes, estos elementos se eliminan del segundo padre

(7 _ _ 0 _ _ 1 5 6 _)

Y se insertan en el hijo manteniendo su orden relativo

(9 7 0 4 1 2 3 5 6 8)

b) **[0,5 puntos]** Indicar de forma razonada cuál sería el operador más adecuado teniendo en cuenta que el algoritmo genético está resolviendo el problema del TSP.

RESP: Dado que el problema es el TSP, es razonable pensar que OX1 va a ser mejor que OX2. El motivo es que OX1 conserva subcadenas de mayor longitud que OX2, y en este caso las subcadenas representan fragmentos de soluciones. En la terminología de los algoritmos genéticos, las subcadenas son bloques básicos. Si dos ciudades consecutivas en un padre, siguen siendo consecutivas en el hijo, entonces el hijo hereda una propiedad relevante: el coste entre esas dos ciudades.

II.- Representación (3 puntos)

2.- Se desea representar el siguiente conocimiento mediante lógica de predicados:

- Se dispone de un edificio con varias habitaciones y cada una de ellas dispone de un botón para encender y apagar la luz.
- Al presionar el botón, la luz de la habitación se enciende, y al volver a pulsar se apaga.

[0,75 puntos] Define los axiomas efecto y axiomas marco que permitan representar este conocimiento mediante lógica de predicados.

Solución:

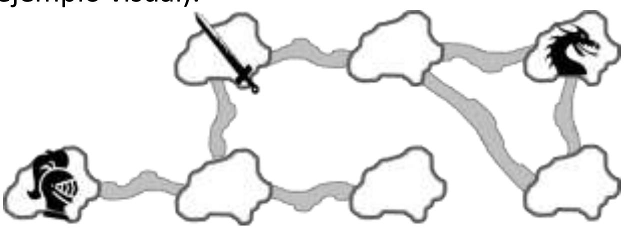
Axiomas efecto:

$$\forall x, s\ Apagada(x, s) \rightarrow Encendida(x, result(boton(x), s)$$
$$\forall x, s\ Encendida(x, s) \rightarrow Apagada(x, result(boton(x), s)$$

Axiomas marco:

$$\forall x, y, s\ Apagada(x, s) \wedge (x \neq y) \rightarrow Apagada(x, result(boton(y), s)$$
$$\forall x, y, s\ Encendida(x, s) \wedge (x \neq y) \rightarrow Encendida(x, result(boton(y), s)$$

3.- **[1 punto]** Se desea implementar un sistema de reglas (un bot) para ganar en un sencillo juego. El jugador controla a un caballero que puede moverse entre distintos poblados de un mapa comunicados por senderos. En uno de estos poblados se encuentra un dragón, por lo que el objetivo del jugador es encontrarlo y matarlo. No obstante, el caballero comienza el juego sin ningún arma, por lo que antes de enfrentarse al dragón deberá encontrar una espada. El dibujo muestra un posible mapa (NOTA: El juego debe funcionar para cualquier mapa, este es solo un ejemplo visual).



Diseñar un sistema de reglas en CLIPS tal que:

- El jugador pueda moverse de un poblado a otro siempre que haya un sendero entre ambos.
- El jugador no vuelva nunca al poblado inmediatamente anterior.
- Si en un poblado vecino hay una espada, el jugador se desplace ahí (en el ejemplo anterior, si el caballero está en el segundo poblado deberá elegir desplazarse hacia el norte, que es donde se encuentra la espada)
- Si el jugador se encuentra en el poblado con la espada, que la recoja (el jugador pasa a tener la espada, y ésta desaparece de donde esté).
- El juego finaliza cuando el jugador llega a la habitación dónde se encuentra el dragón. Si el jugador tiene la espada, se mostrará un mensaje indicando que ha ganado. En caso contrario el mensaje indicará que ha perdido.

Para ello pueden utilizarse opcionalmente las siguientes plantillas de hechos en CLIPS:

```
(deftemplate sendero
	(slot origen) ; Poblado origen
	(slot destino) ; Poblado destino
)
```

```
(deftemplate caballero
	(slot poblado) ; Ubicación actual
	(slot tieneEspada (allowed-symbols no si)) ; Indica si tiene una espada o no
	(slot pobladoPrevio) ; Último poblado visitado por el caballero
)
(deftemplate elemento
	(slot nombre) (allowed_symbols espada dragon)
	(slot poblado)
)
```

Solución:

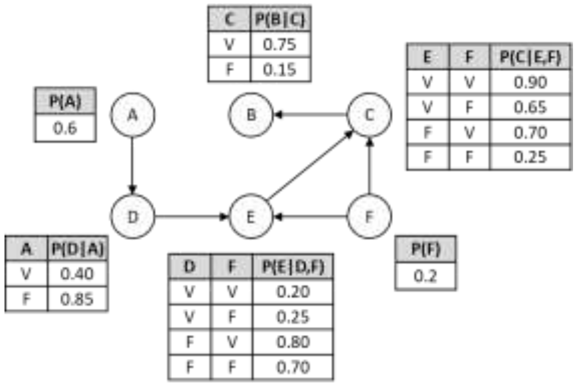
```
(defrule cambiar-poblado-espada
	(declare (salience 1))
	?c <- (caballero (poblado ?p) (tieneEspada ?e) (pobladoPrevio ?u))
	(sendero (origen ?p) (destino ?d & ~?u))
	?e <- (elemento (nombre espada) (poblado ?d))
	=>
	(assert (caballero (poblado ?d) (tieneEspada si) (pobladoPrevio ?p)))
	(retract ?c)
	(retract ?e)
)

(defrule cambiar-poblado
	?c <- (caballero (poblado ?p) (tieneEspada ?e) (pobladoPrevio ?u))
	(sendero (origen ?p) (destino ?d & ~?u))
	=>
	(assert (caballero (poblado ?d) (tieneEspada ?e) (pobladoPrevio ?p)))
	(retract ?c)
)

(defrule ganar
	?c <- (caballero (poblado ?p) (tieneEspada si))
	(elemento (nombre dragon) (poblado ?p))
	=>
	(printout t "Has ganado" crlf)
	(retract ?c)
)

(defrule perder
	?c <- (caballero (poblado ?p) (tieneEspada no))
	(elemento (nombre dragon) (poblado ?p))
	=>
	(printout t "Has perdido" crlf)
	(retract ?c)
)
```

4.- Dada la siguiente red bayesiana, responder razonadamente a las siguientes preguntas:



a) **[0,5 puntos]** Sabiendo que la variable B es falsa, ¿podemos afirmar que las variables A y F son independientes? Justifica la respuesta utilizando el criterio de D-separación.
Solución: No son independientes, ya que el camino A-> D -> E <- F cumple que:

- Hay un nodo cabeza-no cabeza entre A y E: Como D no está observado, no hay bloqueo
- Hay un nodo cabeza-cabeza entre D y F: Como E tiene un descendiente (B) observado, no hay bloqueo.

Como hay al menos un camino no bloqueado, no son independientes.

b) **[0,75 puntos]** Se desea calcular la probabilidad P(E | ¬D, ¬B) mediante el método de muestreo estocástico. Haz una iteración de este método utilizando la siguiente secuencia de números aleatorios: 0.9, 0.45, 0.6, 0.5, 0.15, 0.8.
Solución: Primero hay que ordenar las variables según su orden topológico en la red: A, F, D, E, C, B (también valdría A,D,F,E,C,B)

Variable	Probabilidad	Comparación	Valor asignado
A	$P(A) = 0.6$	$0.9 \nless 0.6$	Falso
F	$P(F) = 0.2$	$0.45 \nless 0.2$	Falso
D	$P(D\|\neg A) = 0.85$	$0.6 < 0.85$	Verdadero
E	$P(E\|D, \neg F) = 0.25$	$0.5 \nless 0.25$	Falso
C	$P(C\|\neg E, \neg F) = 0.25$	$0.15 < 0.25$	Verdadero
B	$P(B\|C) = 0.75$	$0.8 \nless 0.2$	Falso

III.- Aprendizaje (3,5 puntos)

5.- **[0,25 puntos]** Supón un problema de clasificación para predecir si se da una enfermedad que suele aparecer en 5 casos de cada 1000 personas. Indica que medida de evaluación de modelos sería más adecuada en este caso y porqué.
Solución: Dado el alto grado de desbalanceo en los valores de la variable objetivo es necesaria una medida de evaluación que tenga en cuenta este factor. Una opción muy adecuada es kappa ya que valora el acierto de un modelo predictivo sobre la base de las frecuencias de la variable objetivo.

6.- **[0,75 puntos]** Dada la siguiente tabla que muestra un conjunto de datos donde *Play* es la variable a predecir:

Temp	40	48	60	72	80	90
Play	No	No	Yes	Yes	Yes	No

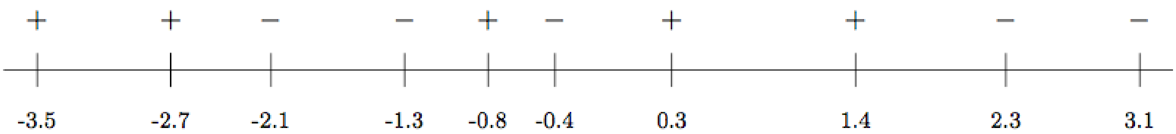
Calcula razonadamente el valor de la ganancia de información (IG) sobre *Play* con *Temp*, que es numérica.

Solución: Como la variable *Temp* es numérica hay que definir los posibles split points para poder calcular IG. En este caso los únicos posibles son entre 48 y 60 (54), y entre 80 y 90 (85), porque es donde se produce un cambio en *Play* estando las instancias ordenadas por *Temp*.

$$IG(Play, Temp > 54) = H(Play) - H(Play|Temp > 54)$$
$$= 1 - \frac{2}{3}H(Play|Temp > 54) - \frac{1}{3}H(Play|Temp \leq 54) = 1 - 0.54 - 0$$
$$= 0.46$$
$$IG(Play, Temp > 85) = H(Play) - H(Play|Temp > 85)$$
$$= 1 - \frac{1}{6}H(Play|Temp > 85) - \frac{5}{6}H(Play|Temp \leq 85) = 1 - 0 - 0.81$$
$$= 0.19$$

Nos quedaríamos con la mejor configuración que es el split point en 54 y su ganancia de información es 0.46.

7.- **[0,75 puntos]** Dado el siguiente conjunto de ejemplos con un atributo real y una variable objetivo que toma dos valores (+,-)



x	-3.5	-2.7	-2.1	-1.3	-0.8	-0.4	0.3	1.4	2.3	3.1
y	+	+	−	−	+	−	+	+	−	−

Utilizando como conjunto de validación los ejemplos 1, 5 y 6 ({-3.5, +}, {-0.8, +} y {-0.4,-}) y como conjunto de entrenamiento el resto, **calcula razonadamente** el porcentaje de error que se obtiene con el paradigma de vecinos más cercanos con K=3 y distancia euclídea.

Solución: Los 3 vecinos de la primera instancia de validación son claramente las instancias 2, 3 y 4, y como dos de ellas tiene clase '-' la predicción sería este valor siendo un fallo.

Los 3 vecinos de la segunda instancia de validación son las instancias 3, 4, y 7. La instancia 6 está más cerca que la 3 pero no pertenece al conjunto de entrenamiento. En este caso la predicción es predicción es de nuevo '-' y también es un fallo.

Los 3 vecinos de la tercera instancia de validación son las instancias 3, 4 y 7. La instancia 5 no se puede considerar por la misma razón que antes, y entre la instancia 3 ({-2.1,-}) y la instancia 8 ({1.4,+}), la 3 está más cerca de la 6 ($|-2.1-(-0.4)| < |1.4-(-0.4)|$). En este último caso hay un acierto.

Por tanto, el porcentaje de error es de 2/3 o 0.66.

8.- **[1 punto]** Describe en pseudo-código el algoritmo de propagación hacia atrás o backpropagation para aprender una red neuronal artificial a partir de un conjunto de entrenamiento con D casos de la forma $\langle \boldsymbol{x}, y \rangle$, una tasa de aprendizaje η, y suponiendo una sola capa de neuronas ocultas con n_h neuronas.

Solución: Dia positivas 22 y 23 de Redes neuronales.

9.- **[0,75 puntos]** Dado el siguiente conjunto de entrenamiento {(1,+),(2,+),(4,-),(5,-),(6,+)} al que se la he aplicado SVM con kernel $K(x,y) = (xy+1)^2$, obteniendo $\alpha_1 = 0$, $\alpha_2 = 2.499$, $\alpha_3 = 0$, $\alpha_4 = 7.331$, $\alpha_5 = 4.832$. Sabiendo que $b = 9$, **calcula razonadamente** la predicción de este modelo para 3.5 y 5.5. NOTA: usa al menos un decimal significativo en los cálculos.

Solución:

$$g(3.5) = 2.499 \cdot 1 \cdot (2 \cdot 3.5 + 1)^2 + 7.331 \cdot (-1) \cdot (5 \cdot 3.5 + 1)^2 + 4.832 \cdot 1$$
$$\cdot (6 \cdot 3.5 + 1)^2 + 9 = 159.936 - 2509.035 + 2338.688 + 9 = -1.411$$

$$g(5.5) = 2.499 \cdot 1 \cdot (2 \cdot 5.5 + 1)^2 + 7.331 \cdot (-1) \cdot (5 \cdot 5.5 + 1)^2 + 4.832 \cdot 1 \cdot (6 \cdot 5.5 + 1)^2 + 9 = 359.856 - 5954.605 + 5585.792 + 9 = 0.043$$

Según estos resultados la entrada 3.5 se clasifica como '-' y la entrada 5,5 como '+'.

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_d \sum_k (y_{kd} - o_{kd})^2$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\|x-y\|^2/(2\sigma^2)}$$

$$H(X) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

$$\Delta w_i = \eta(y_d - \boldsymbol{w}\boldsymbol{x}_d)x_{di}$$

$$b = \frac{1}{|VS|}\sum_{i \in VS} y_i - \boldsymbol{w}\boldsymbol{x}_{\boldsymbol{i}}$$

$$IG(X, Y) = H(X) - H(X|Y)$$

$$g(\cdot) = \begin{cases} 1 \text{ si } \boldsymbol{wx}>0 \\ -1 \text{ en otro caso} \end{cases}$$

$$d = \frac{2}{\|\boldsymbol{w}\|}$$

$$H(X|Y) = \sum_{y \in Y} p(y) H(X|Y = y)$$

$$K(x_i, x_j) = \phi(x_i)\phi(x_j)$$

$$dist(\vec{x}, \vec{y}) = \left(\sum_i |x_i - y_i|^p\right)^{1/p}$$

$$\boldsymbol{w} = \sum_{i \in VS} \alpha_i y_i \boldsymbol{x}_{\boldsymbol{i}}$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{xy} + 1)^d$$

$$E(\boldsymbol{w}) = \frac{1}{2}(y_d - \boldsymbol{w}\boldsymbol{x}_d)^2$$

$$g(x) = \frac{1}{1 - e^{-x}}$$

$$y(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$

$$g(\boldsymbol{x}) = \boldsymbol{wx} + b$$