



APELLIDOS:

PL:

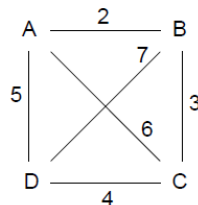
NOMBRE:

DNI:

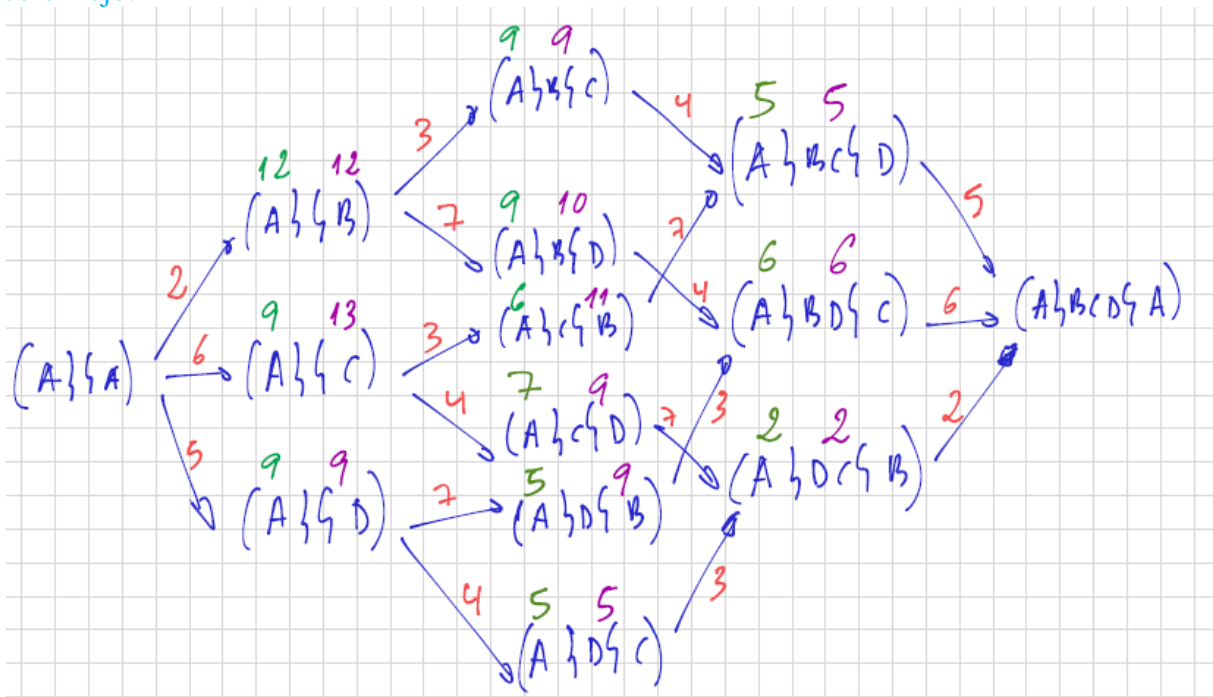
ESCUELA DE INGENIERÍA INGORMÁTICA**SISTEMAS INTELIGENTES****Examen Final de Teoría. Lunes 22 de Mayo de 2017.**

1.- [2,75 puntos] Se trata de utilizar el algoritmo A* para resolver dos variantes del problema del viajante de comercio (TSP). La primera es la variante clásica en la que un viajante debe partir de una ciudad A, pasar por el resto de las ciudades una y solo una vez y volver a la ciudad de partida con un coste mínimo. En este caso se pide:

a) Describir de forma genérica el espacio de búsqueda para un problema con N ciudades, conectadas directamente todas con todas, y dibujar el espacio completo para la siguiente instancia del problema:



RESPUESTA: Un estado se puede representar por tres elementos: la ciudad de partida, el conjunto de ciudades por las que ya pasó el viajante y la ciudad actual. Los sucesores de un estado vienen dados por cada una de las ciudades no visitadas, y el coste para ir a cada uno de ellos desde el actual es el coste desde la ciudad actual a la nueva ciudad. Para el problema anterior, el espacio de búsqueda completo es el siguiente, los costes de los arcos se indican en color rojo:



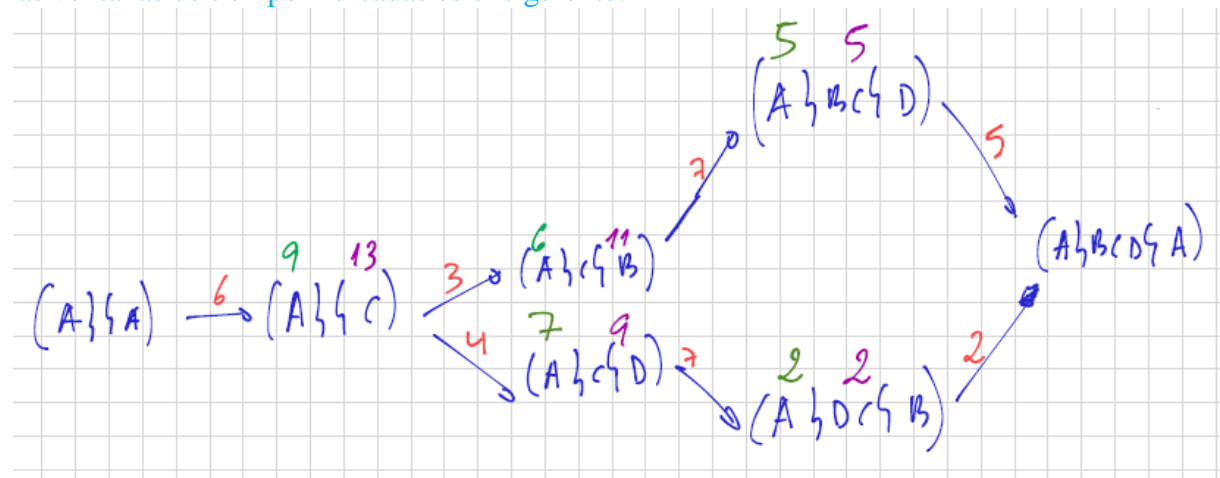
b) Describir un heurístico, lo mejor posible, para el problema. Justificar sus propiedades y dar los valores que toma este heurístico para todos los nodos del espacio de búsqueda del ejemplo anterior.

RESPUESTA: Uno de los mejores heurísticos para este problema es el que consiste en calcular el árbol de expansión mínimo del grafo “residual”, es decir el grafo de conexiones eliminando las ciudades ya visitadas, salvo la inicial y la actual, así como los arcos que conectan a las ciudades eliminadas con otras ciudades. El valor de este heurístico se muestra

en color verde en la figura anterior. En morado están los valores de $h^*(n)$ que como se puede ver son mayores o iguales que $h(n)$ para todo nodo n .

c) Consideremos ahora la versión del TSP con “ventanas de tiempo”. Es como el anterior, con una restricción añadida: el viajante tiene que pasar por cada ciudad X en un tiempo del intervalo $[t_{Xi}, t_{Xf}]$. En este caso se pide lo mismo que para la versión anterior. Como ejemplo concreto consideraremos el problema anterior en el que las ventanas de tiempo para las ciudades B, C, y D son respectivamente $[8,19]$, $[5,7]$ y $[9,17]$.

REPUESTA: En este caso, el espacio de búsqueda se define de una forma análoga. Pero debido a la nueva restricción los estados alcanzables desde uno dado se reducen a los definidos por las ciudades no visitadas que son alcanzables desde la ciudad actual en un instante del intervalo correspondiente. En cuanto al heurístico, en principio sirve el mismo que para el TSP original. Dado que el TSP original es en realidad una relajación del TSP con ventanas de tiempo, cualquier heurístico admisible para el TSP original también lo es para el TSP con ventanas de tiempo. El espacio de búsqueda completo para el problema anterior con las ventanas de tiempo indicadas es el siguiente:



2.- [0,5 puntos] Imagina que quieres resolver el problema de las N-Reinas utilizando un algoritmo genético. Explica qué codificación podrías utilizar para representar los individuos, y de qué forma evaluarías la calidad de cada individuo.

Nota: recuerda que el problema de las N-reinas consiste en colocar N reinas en un tablero de ajedrez de tamaño $N \times N$, de tal forma que no se ataquen entre ellas (una reina ataca a otra reina si está en su misma fila, su misma columna, o en alguna de sus mismas diagonales).

Solución:

Representación: hay varias posibles representaciones, pero una representación eficiente tendría en cuenta que en una solución sólo puede haber una reina por fila y una reina por columna. Entonces, para reducir el espacio de búsqueda podríamos utilizar permutaciones. Consideremos por ejemplo un tablero de 4×4 , entonces la permutación $(1,3,4,2)$ indicaría que en la primera fila hay una reina en la columna 1, en la segunda fila hay una reina en la columna 3, en la tercera fila hay una reina en la columna 4, y en la cuarta fila hay una reina en la columna 2.

Evaluación: una forma sería contar cuántos posibles pares de reinas no se atacan entre sí (en el anterior ejemplo 5 de las 6 posibles parejas de reinas no se atacan entre sí). Otra forma sería simplemente contar el número de reinas que no son atacadas por ninguna otra reina (en el anterior ejemplo 2 de las 4 reinas no son atacadas por ninguna otra).

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

a) [0,5] Explica brevemente la diferencia entre deducción, inducción y abducción.

Solución:

En los sistemas de deducción, se parte de una base de conocimiento (p.e. reglas) y una serie de hechos constatados, y a partir de todo ello se extrae una conclusión

En los sistemas de inducción, se parte de una serie de hechos y la conclusión extraída, y se trata de buscar una base de conocimiento (p.e. reglas) que explique las observaciones, por ejemplo a partir de generalizaciones o extrapolación.

Por último, en los sistemas de abducción partimos de una base de conocimiento (p.e. reglas) y la conclusión extraída, y tratamos de averiguar los hechos iniciales que han desencadenado esa conclusión

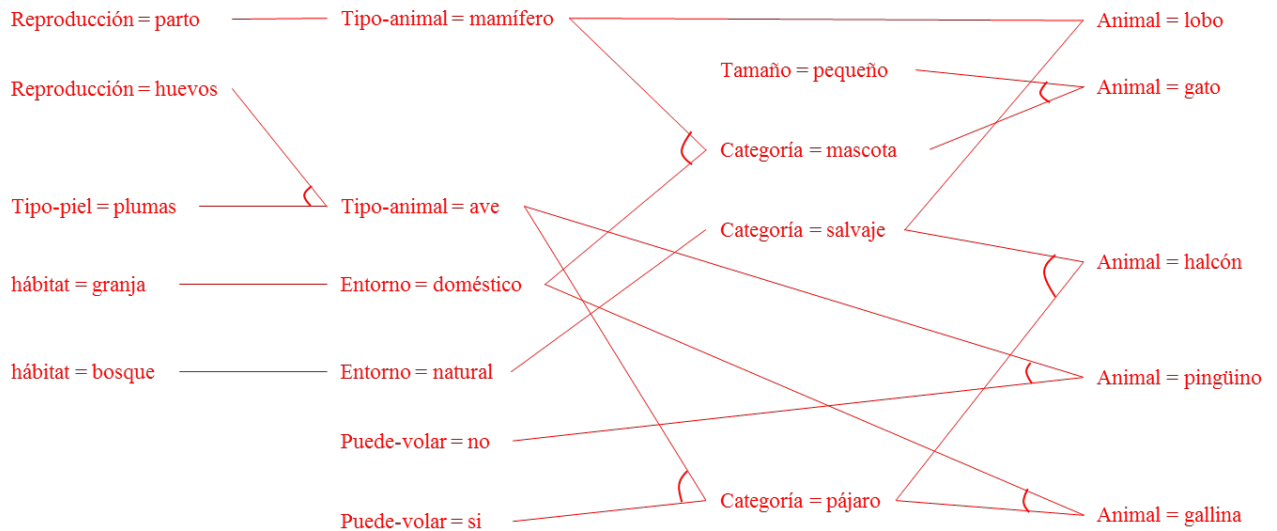
b) [1 punto] Estamos tratando de enseñar a un extraterrestre como distinguir algunos de los animales de la Tierra. Para ello disponemos del siguiente sistema de Sistema de Reglas que funciona mediante encadenamiento hacia adelante y cuya estrategia de resolución de conflictos es en “profundidad”:

Regla 1: IF reproducción = parto THEN tipo-animal = mamífero	Regla 2: IF tipo-piel = plumas AND reproducción = huevos THEN tipo-animal = ave	Regla 3: IF hábitat = granja THEN entorno = domestico
Regla 4: IF hábitat = bosque THEN entorno = natural	Regla 5: IF entorno = natural THEN categoría = salvaje	Regla 6: IF tipo-animal = ave AND puede_volar = si THEN categoría = pájaro
Regla 7: IF entorno = doméstico AND tipo-animal = mamífero THEN categoría = mascota	Regla 8: IF tipo-animal = mamífero AND categoría = salvaje THEN animal = lobo	Regla 9: IF categoría = pájaro AND entorno = domestico THEN animal = gallina
Regla 10: IF tipo-animal = ave AND puede-volar = no THEN animal = pingüino	Regla 11: IF categoría = mascota AND tamaño = pequeño THEN animal = gato	Regla 12: IF categoría = pájaro AND Entorno = salvaje THEN animal = halcón

Responde a las siguientes cuestiones:

a. [0.25] Dibuja el grafo asociado al sistema de reglas.

Solución:



- b. [0.75] Tenemos ante nosotros un animal que se reproduce mediante huevos, tiene plumas, habita en granjas, vuela y es mediano. Indica las reglas que el sistema anterior activará a cada paso, las que se disparan y los nuevos hechos generados hasta saber de qué animal se trata.

Solución:

Paso	Reglas activas	Regla disparada	Nuevos hechos
1	2, 3	2	tipo-animal = ave
2	3,6	6	categoría = pájaro
3	3	3	entorno = doméstico
4	9	9	animal = gallina

- c) [1.25 puntos] Un estudiante que comparte piso decide diseñar un SE en el que dados todos los ingredientes de que dispone, diga qué puede preparar para cenar esta noche. Cada persona en el piso dispone de un armario y una parte de la nevera y solo puede usar lo que hay en ellos. Para indicar qué hay en cada uno y a quién pertenece, se dispone de las siguientes plantillas y hechos:

<pre>(deftemplate armario (slot atun (type FLOAT)) (slot cebolla (type INTEGER)) (slot harina (type FLOAT)) (slot ajo (type FLOAT)) (slot levadura (type FLOAT)) (slot dueño (type SYMBOL))) (deffacts otros (sal) (agua))</pre>	<pre>(deftemplate nevera (slot huevo (type INTEGER)) (slot pimientaV (type FLOAT)) (slot pimientaRojo (type FLOAT)) (slot tomate (type INTEGER)) (slot dueño (type SYMBOL)))</pre>
--	--

- a. [1 punto] Diseñar un sistema de reglas que permita averiguar si puede 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, medio pimiento rojo, 3 cebollas y sal
- Para la masa se necesitan los siguientes ingredientes: 750 gramos de harina, 40 gramos de levadura, sal, agua y un huevo.
- Por último, el sistema debe asegurarse de que los ingredientes utilizados pertenecen a la misma persona. Por otra parte, la sal y el agua son ilimitadas y comunes para todos.

Téngase en cuenta que si utilizo por ejemplo 750 gramos de harina para hacer la empanada, sólo me quedarán 250 para preparar otro plato. Es decir, deberás modificar los hechos de forma que la cantidad de gramos disponibles para ese producto se decremente cada vez que es utilizado.

```
(defrule rellenoEmpanada
  ?fa <- (armario (atun ?at&:(>= ?at 600)) (cebolla ?ce&:(>= ?ce 3)) (dueño ?d))
  ?fn <- (nevera (tomate ?to&:(>= ?to 200)) (pimientoRojo ?pr&:(>= ?pr 0.5)) (dueño ?d))
  (sal)
  (not (relleno ?d))
=>
  (assert (relleno ?d))
  (modify ?fa (atun (- ?at 600)) (cebolla (- ?ce 3)))
  (modify ?fn (tomate (- ?to 200)) (pimientoRojo (- ?pr 0.5)))
)
```

```
(defrule masaEmpanada
  ?fa <- (armario (harina ?ha&:(>= ?ha 0.75)) (levadura ?le&:(>= ?le 40)) (dueño ?d))
  ?fn <- (nevera (huevo ?hu&:(>= ?hu 1)) (dueño ?d))
  (sal)
  (agua)
  (not (masa ?d))
=>
  (assert (masa ?d))
  (modify ?fa (harina (- ?ha 0.75)) (levadura (- ?le 40)))
  (modify ?fn (huevo (- ?hu 1)))
)
```

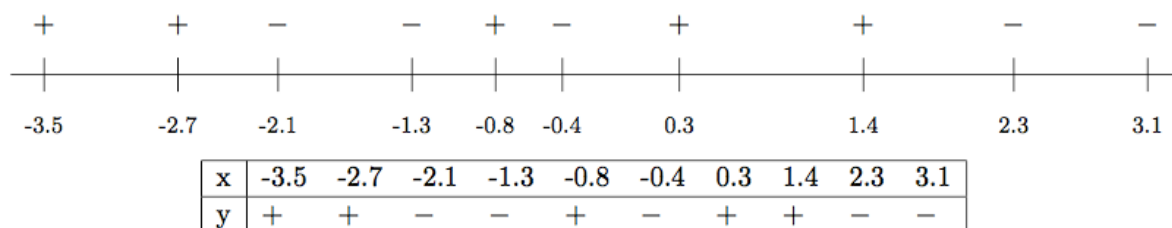
```
(defrule empanada
  ?m <- (masa ?d)
  ?r <- (relleno ?d)
=>
  (assert (empanada ?d))
  (retract ?r)
  (retract ?m)
)
```

- b. [0.25 puntos] Antonio dispone de los siguientes elementos: 750 gramos de atún, 12 huevos, 4 cebollas, 200 gramos de tomate, 1 kilo de harina, 100 gramos de levadura y medio pimiento rojo y. Incluye estos hechos en nuestro sistema experto utilizando CLIPS.

Solución:

```
(deffacts ingredientes
  (armario (atun 750.0) (cebolla 4) (harina 1.0) (levadura 100.0) (dueño Antonio))
  (nevera (huevo 12) (pimientoRojo 0.5) (tomate 200) (dueño Antonio))
)
```

4.- [3 puntos] a) Dado el siguiente conjunto de ejemplos con un atributo real y una variable objetivo que toma dos valores (+,-)



Utilizando como conjunto de test los ejemplos (-3.5, +), (-0.8, +) y (-0.4,-) y como conjunto de entrenamiento el resto de ejemplos, calcula razonadamente el porcentaje de error que se obtiene con K-NN, K=3.

Dado que solo hay una variable independiente, se trata de calcular los 3 valores más cercanos a cada uno de los ejemplos,

- Para el punto -3.5, los más próximos son -2.7, -2.1 y -1.3. EL primero está etiquetado con + y los dos últimos con -, por tanto al punto -3.5, según 3NN se le debería asignar la etiqueta -. Como la clase real del ejemplo es +, es un ERROR
- Para el punto -0.8, los más próximos son -1.3 (-), -0.4 (-) y 0.3 (+). Nuevamente, según 3NN se le debería asignar la etiqueta -. Como la clase real del ejemplo es +, es un ERROR
- Para el punto -0.4, los más próximos son -1.3 (-), -0.8 (+) y 0.3 (+). Según 3NN se le debería asignar la etiqueta +. Como la clase real del ejemplo es -, es un ERROR

Como consecuencia, el porcentaje de acierto del KNN para este conjunto de datos de test es 0

b) Dado un conjunto de datos con 7 variables binarias, Rain, Wind, Summer, Winter, Day y Night que toman el valor “yes” o “no” y Flight_Delay que toma los valores “Delayed” y “not Delayed”, considera la siguiente tabla de datos, que explica cómo se comporta la variable Flight_Delay con respecto a las otras 6.

Feature	Value = yes	Value = no
Rain	Delayed - 30, not Delayed - 10	Delayed - 10, not Delayed - 30
Wind	Delayed - 25, not Delayed - 15	Delayed - 15, not Delayed - 25
Summer	Delayed - 5, not Delayed - 35	Delayed - 35, not Delayed - 5
Winter	Delayed - 20, not Delayed - 10	Delayed - 20, not Delayed - 30
Day	Delayed - 20, not Delayed - 20	Delayed - 20, not Delayed - 20
Night	Delayed - 15, not Delayed - 10	Delayed - 25, not Delayed - 30

Si construimos un árbol de decisión utilizando c4.5, responde razonadamente a la pregunta ¿cuál es la variable raíz?

Para resolver este problema se debe calcular el Gain Ratio para cada variable. Comenzamos por calcular la Ganancia de Información :

$$IG(\text{Attribute}) = E(\text{Delayed}) - E(\text{Delayed}|\text{Attribute})$$

Calculamos entonces dichas Entropías ($E(\text{Delayed}|\text{Attribute}) = \sum_i P(\text{Attribute}=v_i) E(\text{Delayed}|\text{Attribute}=v_i)$)

Attribute puede tomar los valores Rain, Wind, Summer, Winter, Day y Night

Rain	P(Rain=xi)	E(Delayed/Rain=xi)
Yes	0.5	$-3/4 \log_2(3/4) - 3/4 \log_2(3/4)=0.81$
No	0.5	$-3/4 \log_2(3/4) - 3/4 \log_2(3/4)=0.81$

$$E(\text{Delayed}/\text{Rain})=0.5*0.81+0.5*0.81=0.81$$

Wind	P(Wind=xi)	E(Delayed/Wind=xi)
Yes	0.5	$-15/40 \log_2(15/40) - 25/40 \log_2(25/40)=0.69$
No	0.5	$-15/40 \log_2(15/40) - 25/40 \log_2(25/40)=0.69$

$$E(\text{Delayed}/\text{Wind})=0.5*0.69+0.5*0.69=0.69$$

Summer	P(Summer=xi)	E(Delayed/Summer=xi)
Yes	0.5	$-5/40 \log_2(5/40) - 35/40 \log_2(35/40)=0.6$
No	0.5	$-5/40 \log_2(5/40) - 35/40 \log_2(35/40)=0.6$

$$E(\text{Delayed}/\text{summer})=0.5*0.6+0.5*0.6=0.6$$

Winter	P(Winter=xi)	E(Delayed/Winter=xi)
Yes	3/8	$-2/3 \log_2(2/3) - 1/3 \log_2(1/3)=0.47$
No	5/8	$-5/40 \log_2(5/40) - 35/40 \log_2(35/40)=0.68$

$$E(\text{Delayed}/\text{Winter})=0.61$$

Day	P(Day=xi)	E(Delayed/Day=xi)
Yes	0.5	$-1/2 \log_2(1/2) - 1/2 \log_2(1/2)=1$
No	0.5	$-1/2 \log_2(1/2) - 1/2 \log_2(1/2)=1$

$$E(\text{Delayed}/\text{Day})=1$$

Night	P(Day=xi)	E(Delayed/Day=xi)
Yes	25/80	$-10/25 \log_2(10/25) - 15/25 \log_2(15/25)=0.68$
No	55/80	$-25/55 \log_2(25/55) - 30/55 \log_2(30/55)=0.66$

$$E(\text{Delayed}/\text{Day})=0.76$$

Por tanto, dado que $E(\text{Delayed})$ es constante, la variable que maximiza

$$IG(\text{Attribute}) = E(\text{Delayed}) - E(\text{Delayed}|\text{Attribute})$$

Por otro lado, el SplitInfo es 1 para las variables equidistribuidas, y superior a 0.99 para las que no lo son. Dado que Summer es una variable equidistribuida, $GR(\text{Summer})=IG(\text{Summer}) < GR(\text{Attribute})$ para cualquier atributo distinto de summer.

Así que Summer es la raíz del árbol generado por c4.5.

c) Supongamos que tenemos 50 libros de minería de datos de un total de 200 libros en una biblioteca. Si un clasificador predice que 10 libros son de minería de datos, pero solo 5 de ellos lo son realmente, ¿Cuál es la precisión y a cobertura?

Para calcular la precision necesitamos saber el número de TP y de FP. Como Se han retornado 10 libros etiquetados como “minería de DATos”, pero sabemos que solo 5 lo son, TP=5 y FP=5, así que la precision es

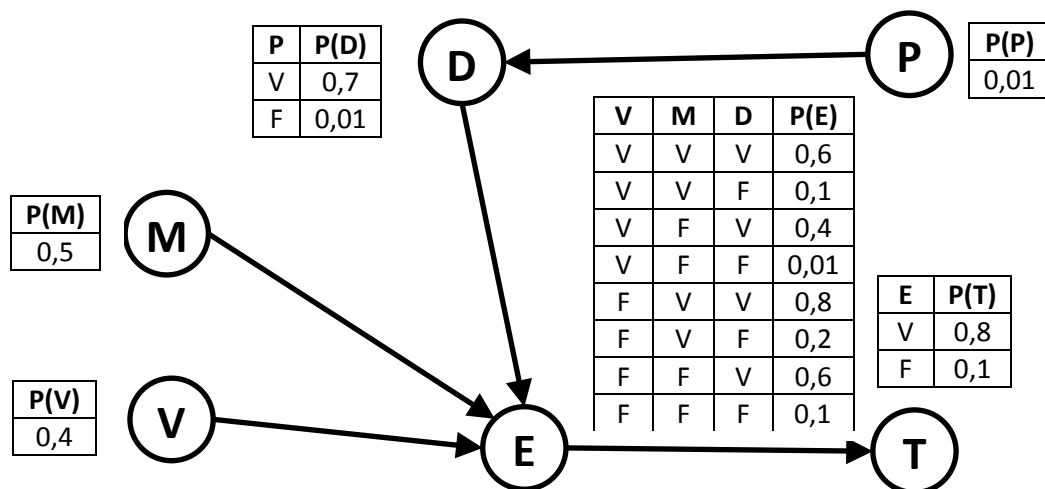
$$P=TP/(TP+FP)=5/10=0.5$$

Para calcular la cobertura, necesitamos conocer el número de Falsos negativos (FN). Sabemos que de los 50 libros de minería de datos, solo se han clasificado como de minería de datos 5, así que hay 45 falsos negativos. Entonces

$$R=TP/(TP+FN)=5/(5+45)=5/50=0.1$$

5.- Considera la siguiente red bayesiana, en donde hay una serie de factores que influyen en la probabilidad de tener una determinada enfermedad (variable E). Dichos factores son: que el individuo tenga hábitos de vida saludables o no (variable V), que el sexo del individuo sea masculino o no (variable M), y que el individuo tenga o no un determinado defecto genético

(variable D). La variable P indica si los padres del individuo son portadores del defecto genético, lo cual influye en la probabilidad de que el individuo tenga dicho defecto. Por último la variable T representa el resultado de un test médico que determina si el individuo tiene la enfermedad.



a) [0,3 puntos] Calcula $P(D, \neg V, P, M, T)$ (nota: si no tienes calculadora puedes dejar las operaciones indicadas, por ejemplo $0,3 * 0,2 + 0,5 + 0,8 * 0,3 * 0,4$)

Solución:

$$P(D, \neg V, P, M, T) = P(D, \neg V, P, M, T, \neg E) + P(D, \neg V, P, M, T, E) = 0,000042 + 0,001344 = 0,001386$$

$$P(D, \neg V, P, M, T, \neg E) = P(D|P) * P(\neg E|\neg V, M, D) * P(\neg V) * P(P) * P(M) * P(T|\neg E) = 0,7 * 0,2 * 0,6 * 0,01 * 0,5 * 0,1 = 0,000042$$

$$P(D, \neg V, P, M, T, E) = P(D|P) * P(E|\neg V, M, D) * P(\neg V) * P(P) * P(M) * P(T|E) = 0,7 * 0,8 * 0,6 * 0,01 * 0,5 * 0,8 = 0,001344$$

b) [0,3 puntos] Dada la estructura de la red, ¿podemos decir que P y M son independientes, si conocemos el valor de T? Debes razonar la respuesta utilizando el criterio de D-separación. También puedes utilizar la condición de Markov si ésta fuese suficiente.

Solución:

Aplicamos el criterio de D-Separación, ya que en este caso la condición de Markov no es suficiente para determinar la independencia (recordad que dicha condición no se puede aplicar si conocemos el valor de algún nodo de la red aparte del de los padres de los correspondientes nodos).

El único camino es P-D-E-M y no está bloqueado (recordad que el nodo E no bloquea, debido a que conocemos el valor de un descendiente suyo). Hemos encontrado un camino no bloqueado, luego no son independientes.

c) [0,4 puntos] Imagina que quieres añadir a la red la siguiente información:

- La máquina que realiza el test puede estar defectuosa (variable F), y en ese caso falla más a la hora de determinar si el individuo tiene o no tiene la enfermedad.
- La enfermedad puede producir un extraño síntoma (variable S) con una alta probabilidad, pero es poco probable que se manifieste si no se tiene esta enfermedad. Además, el síntoma aparece con más frecuencia en individuos de sexo femenino.

¿Cómo modificarías la red? Es suficiente con que indiques la nueva estructura (nodos y arcos), no hace falta que indiques ninguna tabla de probabilidad.

