



APELLIDOS:

PL:

NOMBRE:

DNI:

**ESCUELA DE INGENIERÍA INFORMÁTICA****SISTEMAS INTELIGENTES****Examen Extraordinario de Teoría. Lunes 16 de Junio de 2014.**

**1.- [3 puntos]** Se trata de resolver un problema con el algoritmo A\*. El problema consiste en calcular un árbol de expansión mínimo de un grafo dirigido con costes positivos en los arcos, pero con la restricción de que en el árbol calculado el grado máximo que puede tener un nodo es  $N \geq 2$ . Concretamente hay que responder a las cuestiones siguientes:

a) ¿Cómo es el espacio de búsqueda? Dibujar el espacio completo para una instancia trivial dada por un grafo totalmente conectado con 4 nodos y  $N=2$ . Indicar los costes de las reglas y el valor de  $C^*$ .

RESP: Un espacio de búsqueda posible se puede definir de la siguiente forma: cada nodo es un subárbol parcial que incluye un subconjunto de arcos de lo que puede llegar a ser un árbol de expansión mínimo con grado limitado. En el estado inicial tendremos el conjunto vacío. Para generar los sucesores del inicial, podemos tomar como referencia de partida uno de los nodos del grafo y hacer que cada sucesor del inicial contenga exactamente un arco de los que conectan el nodo referencia con otro nodo del grafo. Los sucesores de un nodo intermedio vendrán definidos por todas las posibilidades de añadir un nuevo arco de modo que el conjunto de arcos del nodo sucesor forme un sub-árbol de expansión con el grado limitado. Así, cada nodo objetivo contendrá un conjunto de arcos que formará un árbol de expansión con grado limitado. El que tenga menor coste será la solución óptima.

En la página siguiente se muestra un grafo con 4 nodos totalmente conectado y el espacio de búsqueda para  $N=2$ .

b) Teniendo en cuenta la descripción del espacio de búsqueda del apartado anterior, definir un heurístico admisible lo mejor informado posible. Indicar cómo se calcula el heurístico utilizando como ejemplo algún estado intermedio del problema anterior.

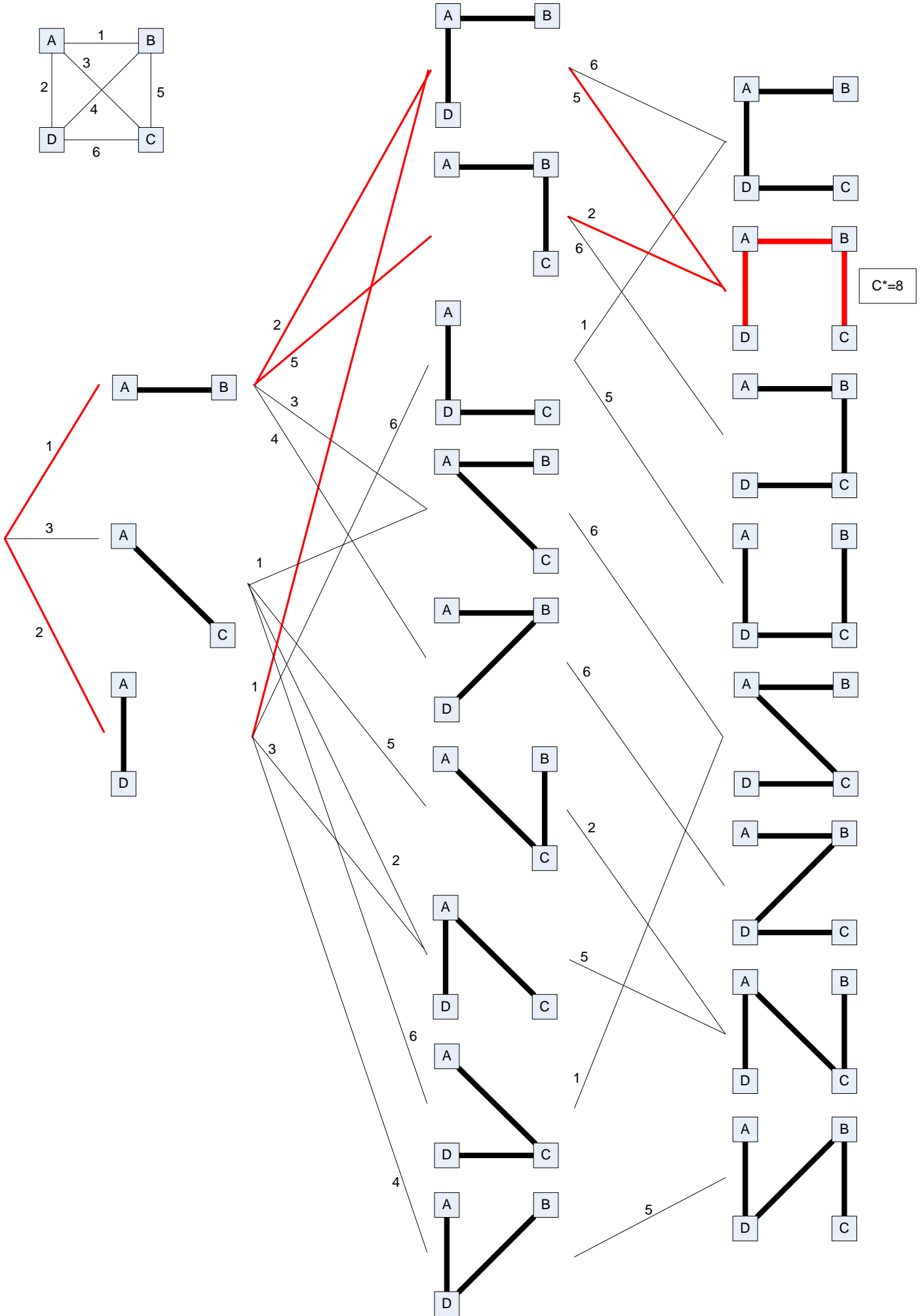
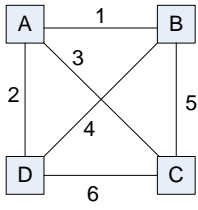
RESP: Dado un estado intermedio, como por ejemplo el que incluye el arco AB, para llegar desde ese nodo a un nodo solución hay que añadir un número de arcos igual al número de nodos “no incluidos” en el estado (en este ejemplo están solo A y B, luego quedan C y D), elegidos entre los arcos que conectan las ciudades no incluidas con otras ciudades, de forma que el conjunto total de arcos forme un arco de expansión mínimo con grado limitado. Podemos relajar el problema de modo que el conjunto de arcos no forme necesariamente un árbol de expansión mínimo, pero que incluya arcos que toquen a todas los nodos no visitados, y que además no toquen a las ciudades incluidas en el estado que ya han alcanzado el grado máximo. En otras palabras, permitimos que el conjunto de arcos resultante no sea un árbol de expansión y que para los nodos no incluidos el grado sea mayor que N.

El algoritmo de cálculo del heurístico consiste en dos pasos: 1.- Calcular los arcos candidatos, 2.- Para cada nodo no incluido en el estado, elegir el arco candidato que lo toque y que tenga menor coste.

Así, para el nodo  $n1=(AB)$ , salvo el propio arco AB todos son candidatos y el conjunto de nodos elegidos sería  $\{AD, AC\}$  y en consecuencia  $h(n1)=5$ .

Para el nodo  $n2=(AB, AD)$  no se puede elegir el arco AC, ya que A ya alcanzó el grado máximo en el estado y en consecuencia los candidatos son  $\{BC, DC\}$ ; como hay que elegir solo uno, éste sería BD y  $h(n1)=4$ .

Para el nodo  $n3=(AC)$ , los candidatos son  $\{AB, AD, BC, DC\}$  y los elegidos serían AB y AD con lo que  $h(n3)=3$ .



---

2.- ¿Cómo se podría resolver el problema de la pregunta anterior con un AG? Una idea: la codificación de las soluciones podría ser una permutación de los arcos del grafo.

Considerar el problema con 4 nodos de la pregunta anterior e indicar 2 ó 3 cromosomas posibles y la solución que representan.

RESP: Aunque hay otras opciones más eficientes, una forma de codificar soluciones es mediante una permutación de los arcos del grafo. Evidentemente, el inconveniente de esta codificación es que para grafos grandes, los cromosomas tienen un número de genes muy grande. Con este esquema, dos cromosomas posibles, para el ejemplo del ejercicio 1, son los siguientes (obviamente AD y DA representarían el mismo arco, luego basta con considerar uno de ellos):

(AB AC AD BD BC CD)  
(AD CD AC BD BC AB)

Un algoritmo de decodificación posible es el siguiente: partimos de un conjunto de arcos vacío. Los arcos se visitan en el orden en el que están en el cromosoma y cada uno de ellos se añade al conjunto de arcos si el conjunto resultante forma un subgrafo sin ciclos y además cada nodo tiene un grado menor o igual que N.

Uno de los inconvenientes de este algoritmo es que si el grafo no es totalmente conectado, hay cromosomas que no representan soluciones válidas, pero cualquier solución tiene un cromosoma que la representa.

Si aplicamos este algoritmo a los dos cromosomas anteriores, las soluciones resultantes vienen dadas por los conjuntos de arcos siguientes:

(AB AC CD) con coste  $1+3+6=10$   
(AD CD BC) con coste  $2+6+5=13$

---

3.- [2 puntos] Utilizando el lenguaje de la lógica de predicados como modelo de representación del conocimiento y la refutación como método de inferencia, probar si el siguiente razonamiento es o no correcto:

*Ninguna persona insegura es psicólogo. Todos los estudiosos de la conducta son psicólogos. Por tanto, ningún estudioso de la conducta es una persona insegura.*

RESP:

-----  
TFM Resolution, psicologosDemo 1  
-----

#newClauses: 2  
#newClauses: 2  
Psicologos Knowledge Base:  
NOT(EXISTS x (PersonaInsegura(x) AND Psicologo(x)))  
FORALL x (EstudiosoConducta(x) => Psicologo(x))

Query: NOT EXISTS x (EstudiosoConducta(x) AND PersonaInsegura(x))  
Proof, Answer Bindings: {x=null}

-----  
|Step | Proof

```

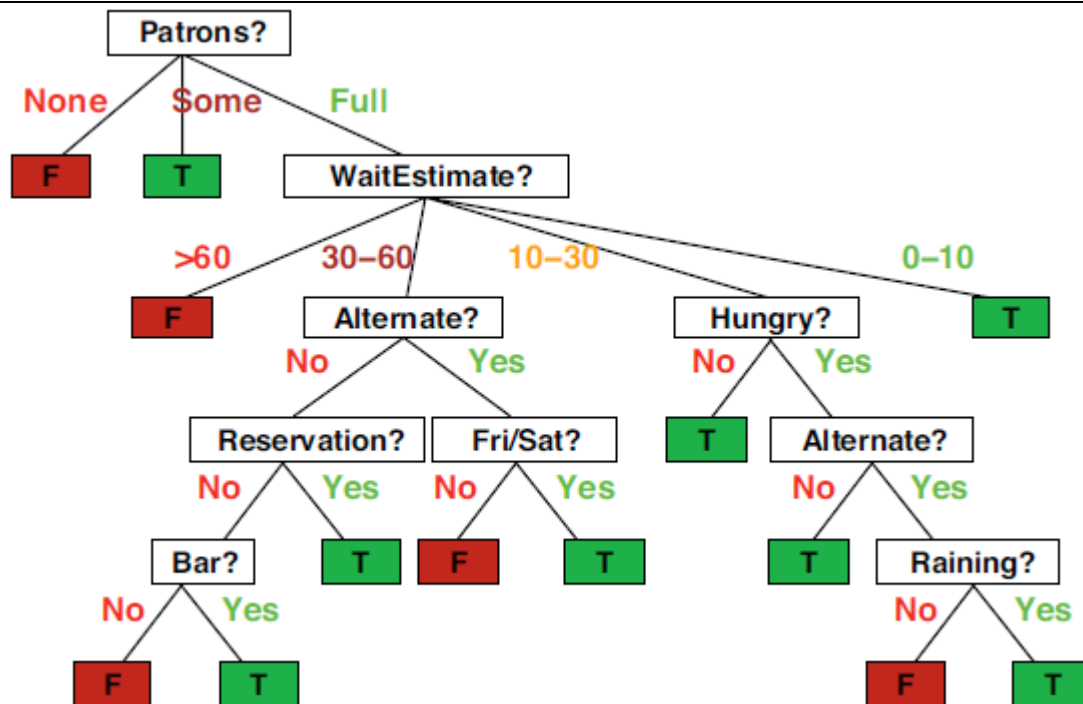
|Justification          |
-----
|1    | NOT(EXISTS x (PersonaInsegura(x) AND Psicologo(x))) |Premise
|
|2    | FORALL x (EstudiosoConducta(x) => Psicologo(x))      |Premise
|
|3    | [PersonaInsegura(SC0)]                                |Goal
|
|4    | [EstudiosoConducta(SC0)]                              |Goal
|
|5    | [~PersonaInsegura(x), ~Psicologo(x)]
|Clausified 1      |
|6    | [~PersonaInsegura(v0), ~Psicologo(v0)]
|Renaming of 5     |
|7    | [~PersonaInsegura(v2), ~Psicologo(v2)]
|Renaming of 6     |
|8    | [~Psicologo(SC0)]
|Resolution: 3,7 {}, {} |
|9    | [~EstudiosoConducta(x), Psicologo(x)]
|Clausified 2      |
|10   | [~EstudiosoConducta(v1), Psicologo(v1)]
|Renaming of 9     |
|11   | [~EstudiosoConducta(v3), Psicologo(v3)]
|Renaming of 10    |
|12   | [Psicologo(SC0)]
|Resolution: 4,11 {v3=SC0}, {} |
|13   | []
|Resolution: 8,12 {}, {} |
-----
-----

```

4.- [3 puntos] Dado el siguiente conjunto de ejemplos para el problema del restaurante

Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>Wait</i>
$X_1$	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T
$X_2$	T	F	F	T	Full	\$	F	F	Thai	30-60	F
$X_3$	F	T	F	F	Some	\$	F	F	Burger	0-10	T
$X_4$	T	F	T	T	Full	\$	F	F	Thai	10-30	T
$X_5$	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
$X_6$	F	T	F	T	Some	\$\$	T	T	Italian	0-10	T
$X_7$	F	T	F	F	None	\$	T	F	Burger	0-10	F
$X_8$	F	F	F	T	Some	\$\$	T	T	Thai	0-10	T
$X_9$	F	T	T	F	Full	\$	T	F	Burger	>60	F
$X_{10}$	T	T	T	T	Full	\$\$\$	F	T	Italian	10-30	F
$X_{11}$	F	F	F	F	None	\$	F	F	Thai	0-10	F
$X_{12}$	T	T	T	T	Full	\$	F	F	Burger	30-60	T

Un determinado algoritmo de aprendizaje ha calculado el siguiente árbol de decisión:



A la vista del resultado, nos planteamos dos cuestiones:

a) ¿Es posible que el “determinado algoritmo” sea el conocido constructor de árboles de decisión que vimos en clase, pero utilizando un heurístico diferente al de máxima ganancia? Para responder a esta pregunta hay que identificar las partes del árbol anterior que nunca serían generadas por el algoritmo de construcción de árboles de decisión.

RESP: No, el nodo T sucesor de WaitEstimate? Sería F y no T, ya que los ejemplos del nodo WaitEstimate? Son x2,x4,x5,x9,x10,x12; 4 son F y 2 son T, luego el nodo debería ser F al no tener ejemplos.

El nodo Reservation? no se generaría con esta etiqueta y en su lugar habría un nodo terminal con T o F al 50%, ya que no tiene ejemplos y los ejemplos de Alternate? Son x2 y x12, uno F y otro T.

b) ¿Cómo podemos reducir el tamaño de este árbol de 6 niveles de forma que se quede en 3 niveles, aunque se pierda calidad de clasificación? Para responder a esta pregunta lo que se pide es definir una estrategia de poda razonable que utilice solamente la información del árbol (y no la de los ejemplos) y aplicarla para reducir el árbol anterior. Luego se debe indicar el árbol resultante y el porcentaje de aciertos y fallos en la clasificación del conjunto de entrenamiento anterior.

RESP: Sin tener en cuenta los ejemplos, la única información que se puede utilizar es la de los nodos T y F para estimar la probabilidad de T o F para un determinado nodo intermedio. Por ejemplo, el nodo Bar? tendrá una probabilidad 0,5 de T, y el nodo Reservation? de 0,75, el Fri/Sat? de 0,5, y el Alternate? De 0,625. Luego en la poda, este nodo se sustituirá por T, ya que tiene una probabilidad estimada mayor que F.