

Sistemes Intel·ligents – Examen Bloc 1, 4 novembre 2022
Test A (1,75 punts) puntuació: max (0, (encerts – errors/3)*1,75/9)

| | | | | | | | | | | | | | | | | | | |
|-----------------|--|---|---|---|---|---|---|---|-------------|--|--|--|--|--|--|--|--|--|
| Cognoms: | | | | | | | | | Nom: | | | | | | | | | |
| Grup: | | A | B | C | D | E | F | G | 4IA | | | | | | | | | |

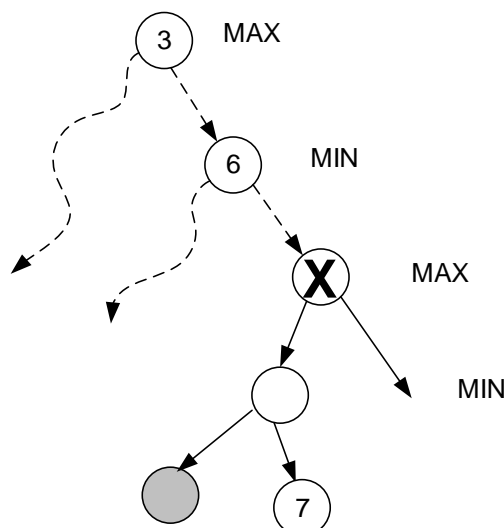
- 1) Donada la següent base de fets $BF = \{(ciutats \text{ València min } 14 \text{ max } 36 \text{ Alacant min } 18 \text{ max } 34 \text{ Castelló min } 12 \text{ max } 30 \text{ Requena min } 8 \text{ max } 38)(selecció \text{ Castelló})\}$, es desitja fer una regla que calcule la diferència de temperatura (màxima – mínima) de la ciutat seleccionada. Quin dels següents patrons **NO SERVIRIA** per a la part esquerra de la regla?

- A. (selecció ?c) (ciutats \$?x ?c ? ?t1 ? ?t2 \$?w)
- B. (selecció ?c) (ciutats \$? ?c ?n1 ?t1 ?n2 ?t2 \$?)
- C. (selecció ?c) (ciutats \$?x ?c ? ?t1 ? ?t2 \$?x)
- D. (selecció ?c) (ciutats \$?x ?c min ?t1 max ?t2 \$?w)

- 2) Es desitja realitzar una cerca en CLIPS. Per fer-ho, les regles no han de contenir la instrucció *retract* perquè:

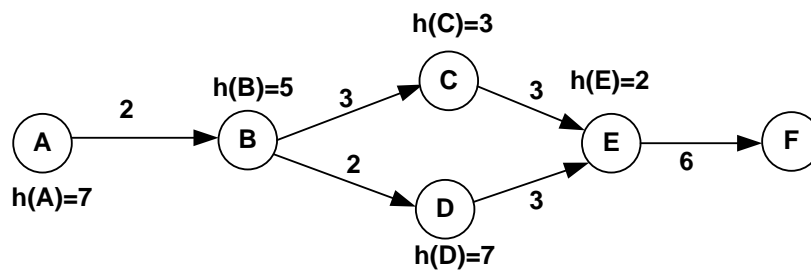
- A. No permet calcular el valor de $g(n)$
- B. No permet trobar la solució òptima
- C. No permetria explorar camins alternatius
- D. Cap de les anteriors

- 3) Quin valor hauria de tenir el node ombrejat perquè es produísca el tall indicat?



- A. Qualsevol valor comprès en $[6, \infty]$
 B. Qualsevol valor comprès en $[-\infty, 6]$
 C. Qualsevol valor comprès en $[3, \infty]$
 D. Qualsevol valor comprès en $[-\infty, 7]$

- 4) Donat l'espai d'estats de la figura, on A és l'estat inicial, F el node meta, i s'indiquen els costos de cada arc i l'estimació $h(n)$ de cada node, marca l'opció **CORRECTA**:

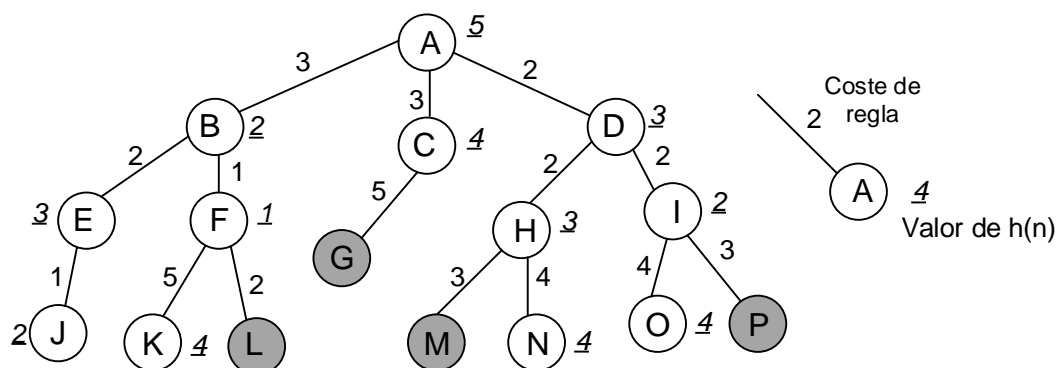


- A. L'aplicació d'un algorisme A (TREE SEARCH, amb control de nodes repetits en OPEN) no obtindrà la senda òptima
- B. L'aplicació d'un algorisme A (GRAPH SEARCH, amb control de nodes repetits en CLOSED, descartant directament el node si ja existeix en CLOSED) obtindrà la senda òptima
- C. La resposta A no és certa perquè $h(n)$ no és admissible
- D. La resposta B no és certa perquè $h(n)$ no és consistent**

- 5) Donats tres algorismes A per a un mateix problema, on A1 i A2 usen una heurística admissible, tal que $\forall n, h^*(n) \geq h_2(n) > h_1(n)$. L'heurística $h_3(n)$ de l'algorisme A3 no és admissible. Es pot assegurar que:

- A. L'algorisme A1 generarà menys nodes que A2
- B. La solució trobada per A1 serà millor que la trobada per A3
- C. L'algorisme A3 generarà més nodes que A2
- D. Els tres algorismes A podrien trobar la mateixa solució**

- 6) Assumint que apliquem diferents estratègies de cerca a l'espai d'estats de la figura, on a igualtat de criteri, es tria el node alfabèticament menor, indica quina de les següents afirmacions és **CORRECTA**:



- A. L'aplicació d'un algorisme en amplària retorna la solució òptima
- B. La solució que troba una cerca de tipus A és el node P
- C. Una estratègia de cost uniforme retornarà la mateixa solució que un algorisme de tipus A**
- D. La funció $h(n)$ no és consistent (monòtona)

7) Donat el SBR, indica l'afirmació **CORRECTA** després de realitzar-se el primer *pattern-matching*:

```
(deffacts prova
(prova 1 2 3 4 5 6 7 8 9 10))
```

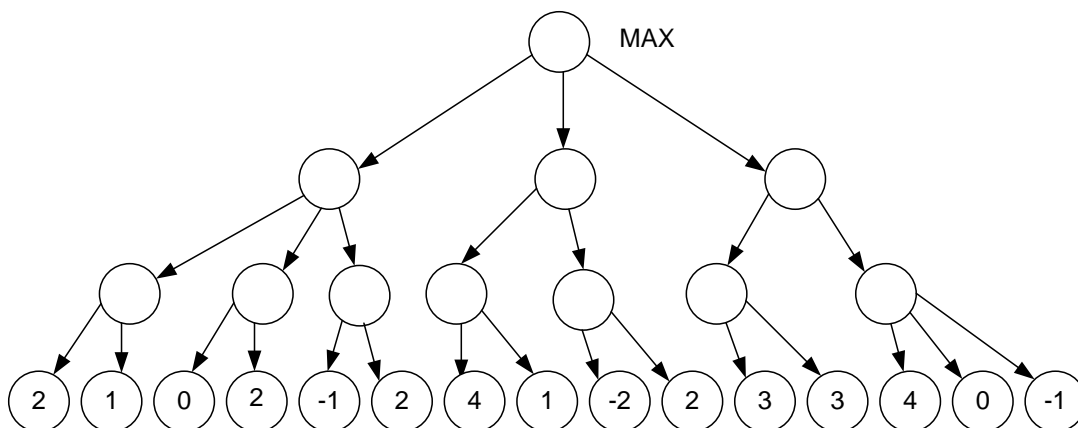
```
(defrule regla
?f1 <- (prova $?a ?b $?c ?d)
=>
(retract ?f1)
(assert (llista $?c)))
```

- A. Es produiran 9 instanciacions
- B. Es produiran 10 instanciacions
- C. Es produiran 11 instanciacions
- D. No es produirà cap instanciació

8) En una cerca en Profunditat, amb màxim nivell de profunditat $m \geq 1$, es troba una solució en el nivell m . Sabent que no existeix cap solució en un nivell menor que m , i que la solució trobada és l'única que existeix a nivell m , indica la resposta **INCORRECTA**:

- A. Una estratègia d'Amplària trobarà la mateixa solució que Profunditat
- B. Una estratègia d'Aprofundiment Iteratiu trobarà la mateixa solució que Profunditat
- C. El nombre de nodes que generarà Amplària serà sempre major que en Profunditat
- D. El nombre de nodes que generarà Aprofundiment Iteratiu serà sempre major que en Profunditat

9) Donat l'arbre de joc de la figura, on s'aplica un procediment alfa-beta, indica quants nodes evitem generar respecte a un algorisme MINIMAX si realitzem una exploració alfa-beta:



- A. Més de 4 i menys de 7
- B. Més de 6
- C. 2
- D. 4

Sistemes Intel·ligents – Examen Bloc 1, 4 novembre 2022

Problema: 2 punts

La companyia EuroTransport es dedica al transport aeri de mercaderies entre 6 ciutats europees: València, Barcelona, París, Roma, Berlín i Oslo. La companyia disposa de diversos avions situats en les 6 ciutats europees esmentades i realitza una planificació diària de les rutes que realitzaran els avions. La ruta de cada avió és fixa i conté un conjunt de ciutats que inclou la ciutat final de destinació i possibles ciutats intermèdies on l'avió fa escala. Per exemple, la ruta d'un avió situat a València pot ser València Berlín; la ruta d'un altre avió situat a Barcelona pot ser Barcelona París Oslo.

Existeix un conjunt de paquets a transportar que estan situats en alguna de les ciutats i el destí dels quals és una altra de les ciutats (per exemple, un paquet està a París i el seu destí és Oslo). Un paquet es pot carregar en un avió si la ruta de l'avió té una parada a la ciutat destí del paquet, bé siga com a escala intermèdia o com a destí final.

El patró que ha de seguir-se necessàriament per a representar la informació dinàmica d'aquest problema és:

`(eurotransport [avio num_as c_as p_am favio]m [paq c_ps cd_ps]m)`

on $c_a, p_a, c_p, cd_p \in \{VAL, BAR, PAR, ROM, BER, OSL\}$ i $num_a \in \text{INTEGER}$, sent:

num_a número de l'avió

c_a ciutat on es troba l'avió

p_a ciutat destí dels paquets carregats en l'avió (els paquets es representen mitjançant una etiqueta que és la seua ciutat destí; per exemple, si només hi ha dos paquets amb destí Berlín en un avió, aquesta variable serà BER BER)

c_p ciutat on es troba el paquet

cd_p ciutat destí del paquet

Utilitzarem a més el següent patró per a representar les rutes dels avions:

`(ruta num_a c_tm) on`

num_a és el número de l'avió

$c_t \in \{VAL, BAR, PAR, ROM, BER, OSL\}$ és la llista de ciutats que recorre l'avió sent la primera ciutat de la llista la ciutat origen de l'avió i l'última la ciutat del destí final de l'avió

Usant CLIPS i cerca en grafs (GRAPH-SEARCH), es demana:

- 1) (0.4 punts) Escriure la **base de fets inicial** amb la següent informació: l'avió 1 fa la ruta València-Berlín; l'avió 2 fa la ruta Barcelona-Roma-Berlín; l'avió 3 fa la ruta València-Barcelona-París-Oslo. Els avions se situen inicialment a la ciutat origen de la seua ruta. Hi ha tres paquets a transportar: un paquet està a Barcelona amb destí Oslo i un altre paquet està a Roma amb destí Berlín i el tercer paquet està a València amb destí Berlín. Inicialment els avions no contenen cap paquet.

- 2) (0.5 punts) Escriure una regla per a moure un avió des de la seua actual ubicació al seu següent destí segons la ruta establida, sempre que l'avió no es trobe ja al seu destí final. El resultat d'executar la regla és que l'avió estarà en el seu següent destí.
- 3) (0.7 punts) Escriu una regla per a carregar un paquet en un avió, el qual ha de trobar-se en la mateixa ciutat que el paquet, sempre que la ciutat destí del paquet aparega en la resta de les ciutats a visitar en la ruta de l'avió. Com a resultat d'executar la regla de carregar, el paquet (representat per l'etiqueta de la seua ciutat destí) estarà ara dins de l'avió i per tant s'elimina la informació [paq c_p^s cd_p^s] del fet.
- 4) (0.4 punts) Escriu una regla que mostre per pantalla els avions que han arribat al seu destí final i no tenen paquets. La regla haurà de mostrar el missatge "L'avió XXX ha arribat al seu destí final i no té paquets" per cadascun dels avions que complisquen les condicions, on XXX és el número de l'avió.

```
(deffacts dades
(ruta 1 VAL BER)
(ruta 2 BAR ROM BER)
(ruta 3 VAL BAR PAR OSL)
(eurotransport avio 1 VAL favio avio 2 BAR favio avio 3 VAL favio paq BAR OSL paq ROM BER paq VAL
BER))
```

```
(defrule moure_avio
(eurotransport $?x avio ?numa ?esta $?y)
(ruta ?numa $?z ?esta ?destí $?w)
=>
(assert (eurotransport $?x avio ?numa ?destí $?y))
)
```

```
(defrule carregar_paquet_mateix_destí
(eurotransport $?x avio ?numa ?esta $?y paq ?esta ?dest $?h)
(ruta ?numa $? ?esta $? ?dest $? )
=>
(assert (eurotransport $?x avio ?numa ?esta ?dest $?y favio $?z $?h))
)
```

```
(defrule avio_en_destí
(eurotransport $?x avio ?numa ?esta favio $?y)
(ruta ?numa $? ?esta)
=>
(printout t "L'avió " ?numa " ha arribat al seu destí final i no té paquets " crlf)
)
```