



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

SIN2223: examen del bloc 1

4 de novembre de 2022

Alfons Juan

DSIC

Departament de Sistemes
Informàtics i Computació

2022-11-04: Qüestió 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)`

La C no val perquè repeteix la x

2022-11-04: Qüestió 2

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

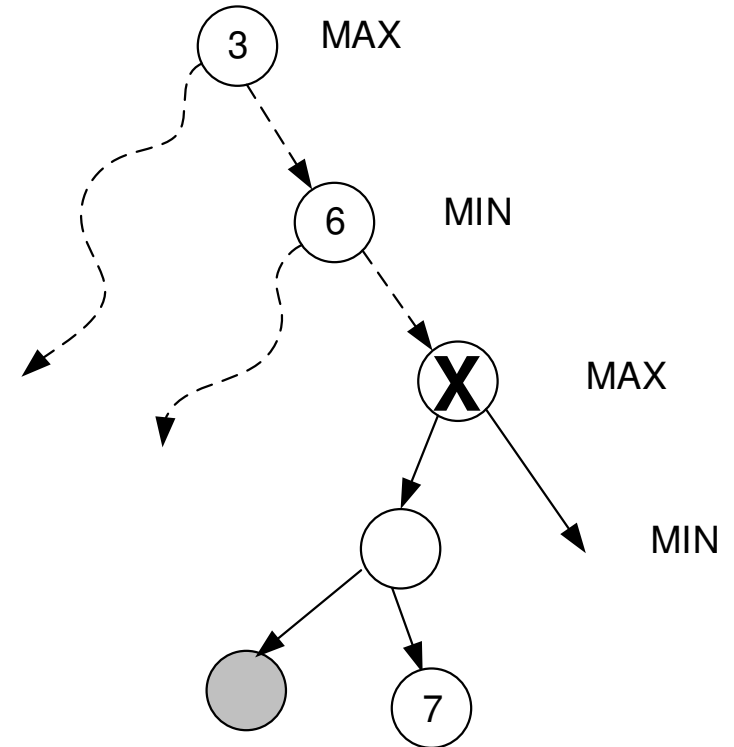
- A) No permet calcular el valor de $g(n)$
- B) No permet trobar la solució òptima
- C) No permetria explorar camins alternatius**
- D) Cap de les anteriors

CLIPS no exploraria camins alternatius corresponents a activacions esborrades per dependre del fet esborrat amb retract

2022-11-04: Qüestió 3

Quin valor hauria de tenir el node ombrejat perquè es produïska 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 $[-\infty, 7]$

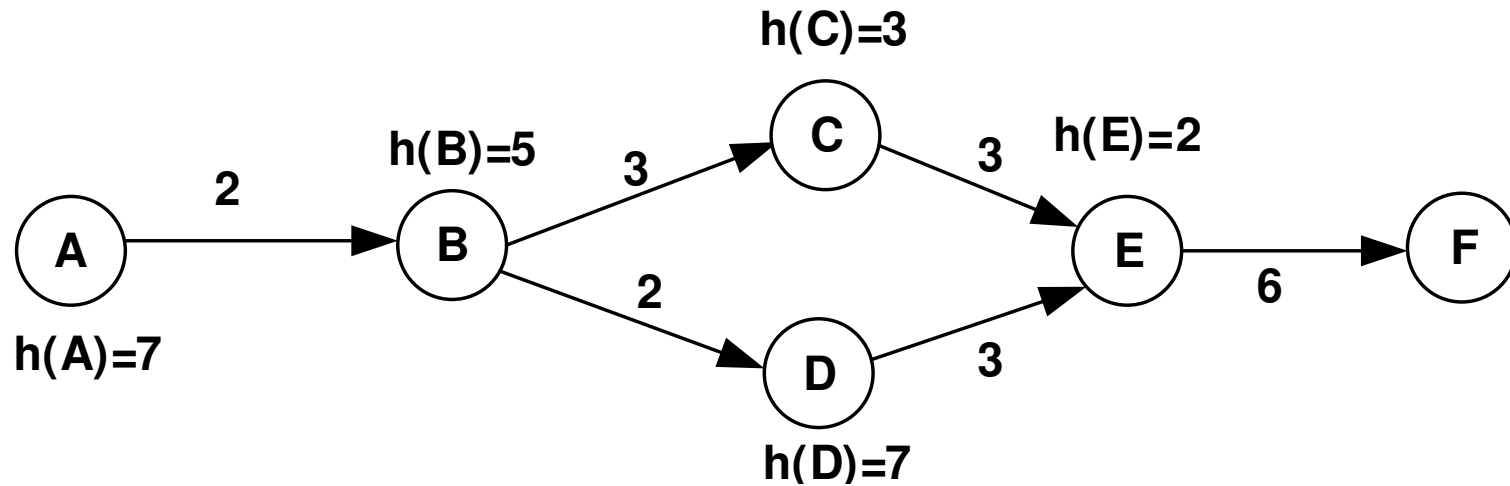


$3 \ a \geq 3 \ b \rightarrow 6 \ a \geq 3 \ b \leq 6 \rightarrow X: \neg i \ a \geq 3 \ b \leq 6$
 $\rightarrow \text{Fill}X: i \ a \geq 3 \ b \leq 6 \rightarrow \text{Fill}X: T \ a \geq 3 \ b \leq \min(6, T)$
 $\rightarrow \text{Fill}X: T \ a \geq 3 \ b \leq \min(6, T)$
 $\rightarrow \text{Fill}X: \min(T, 7) \ a \geq 3 \ b \leq \min(6, T)$
 $\rightarrow X: \min(T, 7) \ a \geq \max(3, \min(T, 7)) \ b \leq 6$

$$\alpha \geq \beta \rightarrow \max(3, \min(T, 7)) \geq 6 \rightarrow \min(T, 7) \geq 6 \rightarrow T \geq 6$$

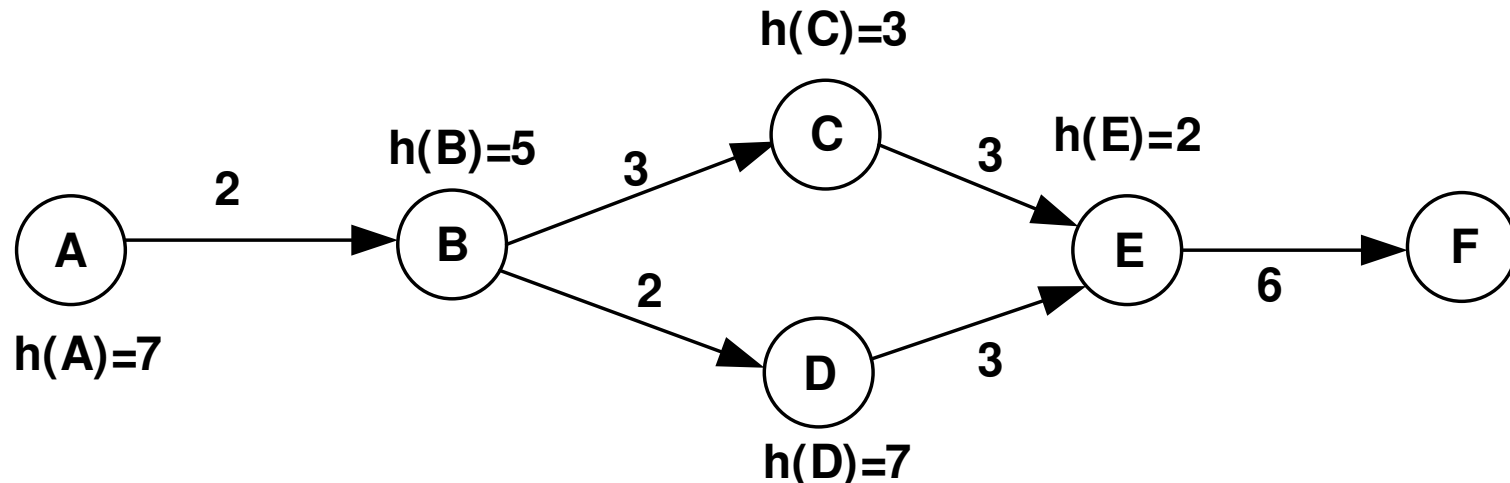
2022-11-04: Qüestió 4

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 una certa perquè $h(n)$ no és consistent



- A) A, cerca en arbre, control de nodes repetits en OPEN:
 $A0+7 \rightarrow B2+5 \rightarrow C5+3 \ D4+7 \rightarrow E8+2 \ D4+7 \rightarrow D4+7 \ F14+0 \rightarrow E7+2 \ F14+0 \rightarrow F13+0 \dots$ obté òptim
- B) A, cerca en graf, control de nodes repetits en CLOSED però descartant node si ja està en CLOSED: com abans fins a ...
 $D4+7 \ F14+0 \rightarrow F14+0 \dots$ no obté òptim
- C) h sí és admissible
- D) h no és consistent: $h(D) \leq w(D, E) + h(E) \Leftrightarrow 7 \leq 3 + 3 \Leftrightarrow \text{Fals}$
 Si una heurística no és consistent, el control de nodes repetits en CLOSED ha de comprovar si el nou node generat està en CLOSED amb pitjor $g+h$; si és així, ha d'eliminar-lo de CLOSED i re-encuar-lo en OPEN

2022-11-04: Qüestió 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:

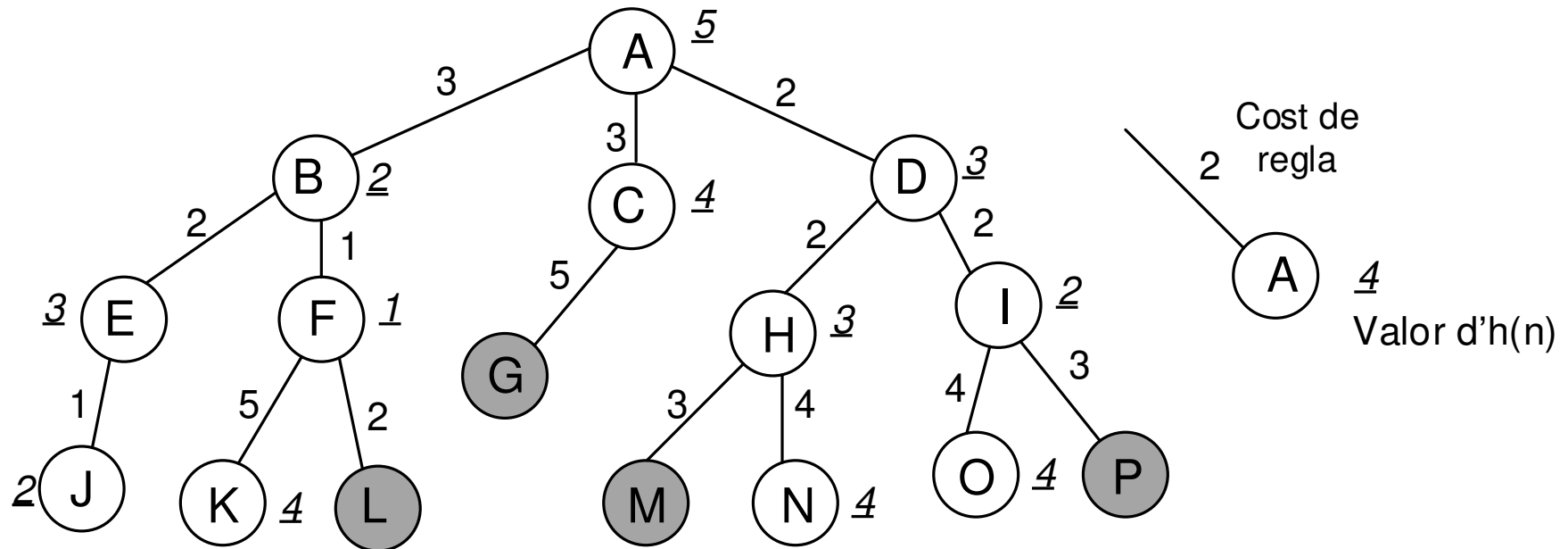
- A) L'algorisme A1 generarà menys nodes que A2
- B) La solució trobada per A1 serà millor que la trobada per A3
- C) L'algorisme A3 generarà més nodes que A2

D) Els tres algorismes A podrien trobar la mateixa solució

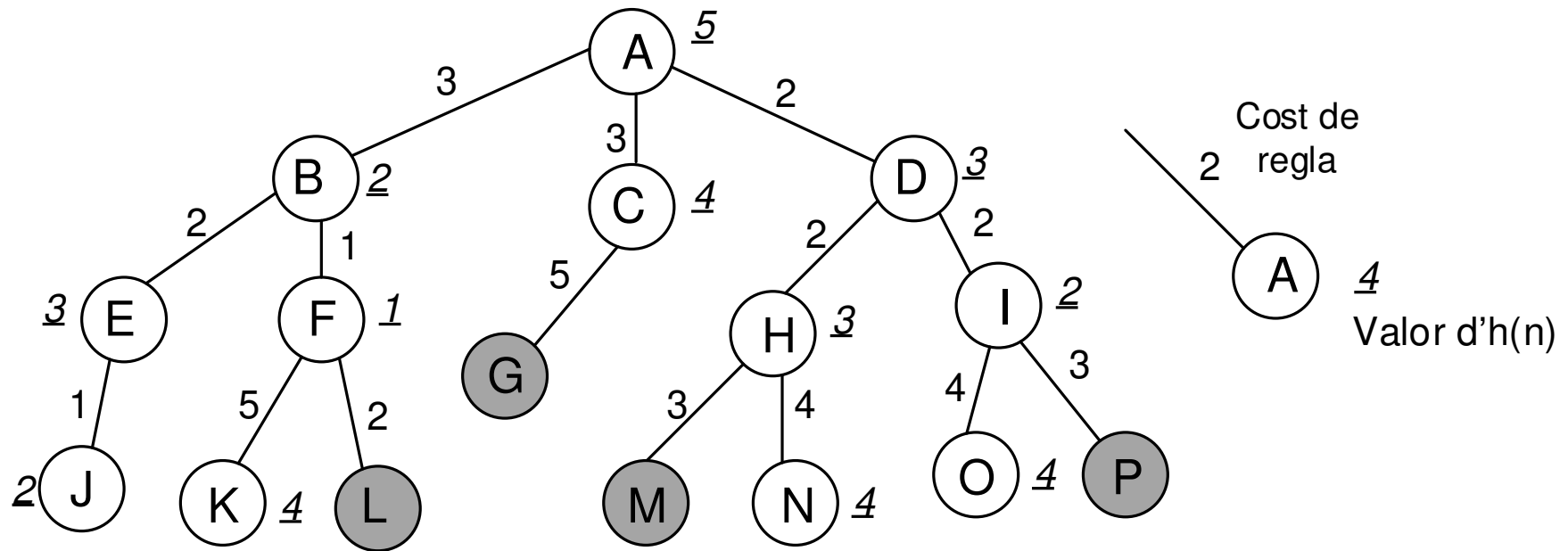
- A) **En general serà al revés; A2 generarà menys nodes que A1 perquè h_2 domina h_1**
- B) **A1 trobarà una solució òptima perquè h_1 és admissible; A3 també pot trobar una solució òptima encara que no estiga garantit ja que h_3 no és admissible**
- C) **No, perquè h_3 pot ser qualsevol heurística no admissible, com ara $h^* + \epsilon$, la qual es comportaria de manera òptima**
- D) **Sí, per exemple si només hi ha una solució possible**

2022-11-04: Qüestió 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)



A) Costs: L 6, M i P 7, i G 8; amplària torna G

B) $A_0+5 \rightarrow B_3+2 \ D_3+2 \ C_3+4 \rightarrow D_3+2 \ F_4+1 \ C_3+4 \ E_5+3 \rightarrow F_4+1$
 $I_4+2 \ C_3+4 \ H_4+3 \ E_5+3 \rightarrow I_4+2 \ L_6+0 \ C_3+4 \ H_4+3 \ E_5+3 \ K_9+4 \rightarrow$
 $L_6+0 \ C_3+4 \ H_4+3 \ P_7+0 \ E_5+3 \ O_8+4 \ K_9+4 \rightarrow$ torna L, no P

C) UCS també torna L

D) Es pot comprovar que és consistent: $h(A) \leq w(A, B) + h(B), \dots$

2022-11-04: Qüestió 7

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

```
1 (defacts prova
2   (prova 1 2 3 4 5 6 7 8 9 10))
3
4 (defrule regla
5   ?f1 <- (prova $?a ?b $?c ?d)
6   =>
7   (retract ?f1)
8   (assert (llista $?c)))
```

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

La ?d encaixa sempre amb el 10 i ?b va de 1 a 9

```

1 <== f-0      (initial-fact)
2 ==> f-0      (initial-fact)
3 ==> f-1      (prova 1 2 3 4 5 6 7 8 9 10)
4 ==> Activation 0      regla: f-1
5 ==> Activation 0      regla: f-1
6 ==> Activation 0      regla: f-1
7 ==> Activation 0      regla: f-1
8 ==> Activation 0      regla: f-1
9 ==> Activation 0      regla: f-1
10 ==> Activation 0     regla: f-1
11 ==> Activation 0     regla: f-1
12 ==> Activation 0     regla: f-1
13 <== f-1      (prova 1 2 3 4 5 6 7 8 9 10)
14 <== Activation 0     regla: f-1
15 <== Activation 0     regla: f-1
16 <== Activation 0     regla: f-1
17 <== Activation 0     regla: f-1
18 <== Activation 0     regla: f-1
19 <== Activation 0     regla: f-1
20 <== Activation 0     regla: f-1
21 <== Activation 0     regla: f-1
22 ==> f-2      (llista 2 3 4 5 6 7 8 9)

```

2022-11-04: Qüestió 8

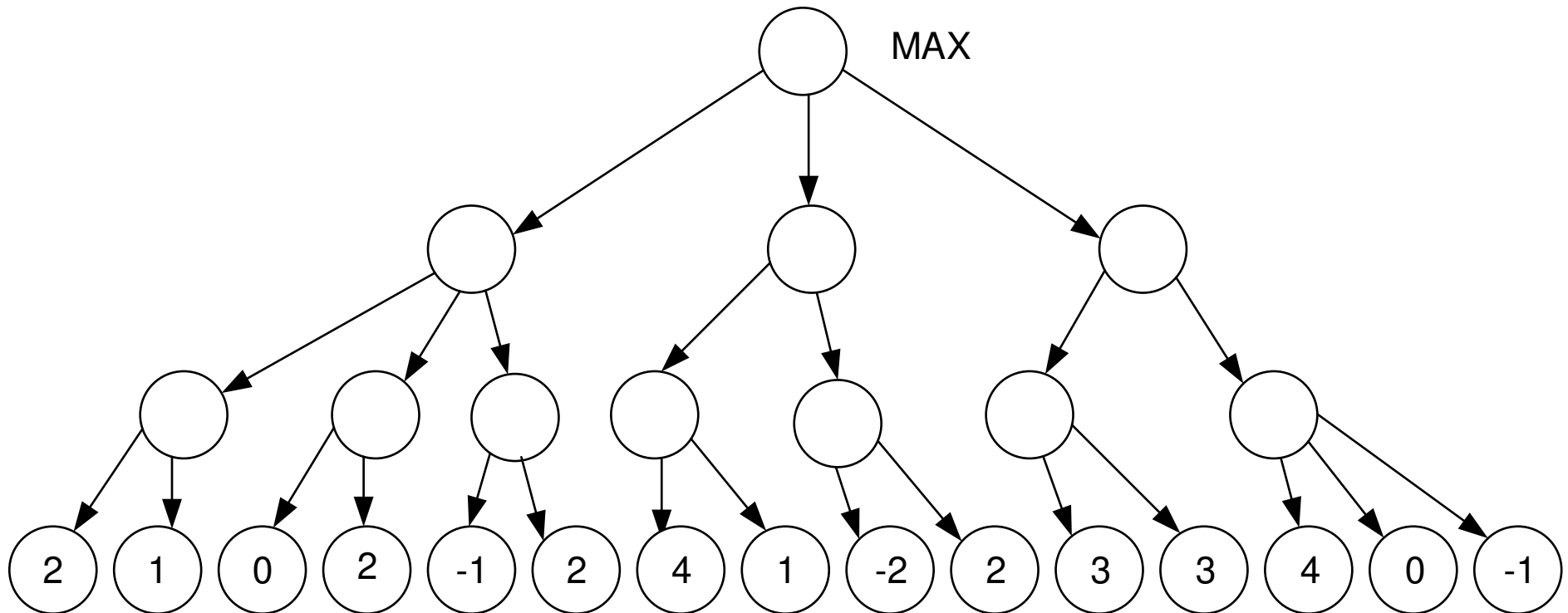
En una cerca en Profunditat, amb màxim nivell de profunditat $m \geq 1$, es troba una solució en el nivell m . Sabent que no existeix cap solució en un nivell menor que m , i que la solució trobada és l'única que existeix a nivell m , indica la resposta INCORRECTA:

- 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

Si la solució és el primer node del nivell m , amplària (amb prova d'objectiu després de selecció) generarà el mateix nombre de nodes que profunditat limitada a profunditat m ; amplària amb prova d'objectiu en generació encara genera menys nodes

2022-11-04: Qüestió 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:

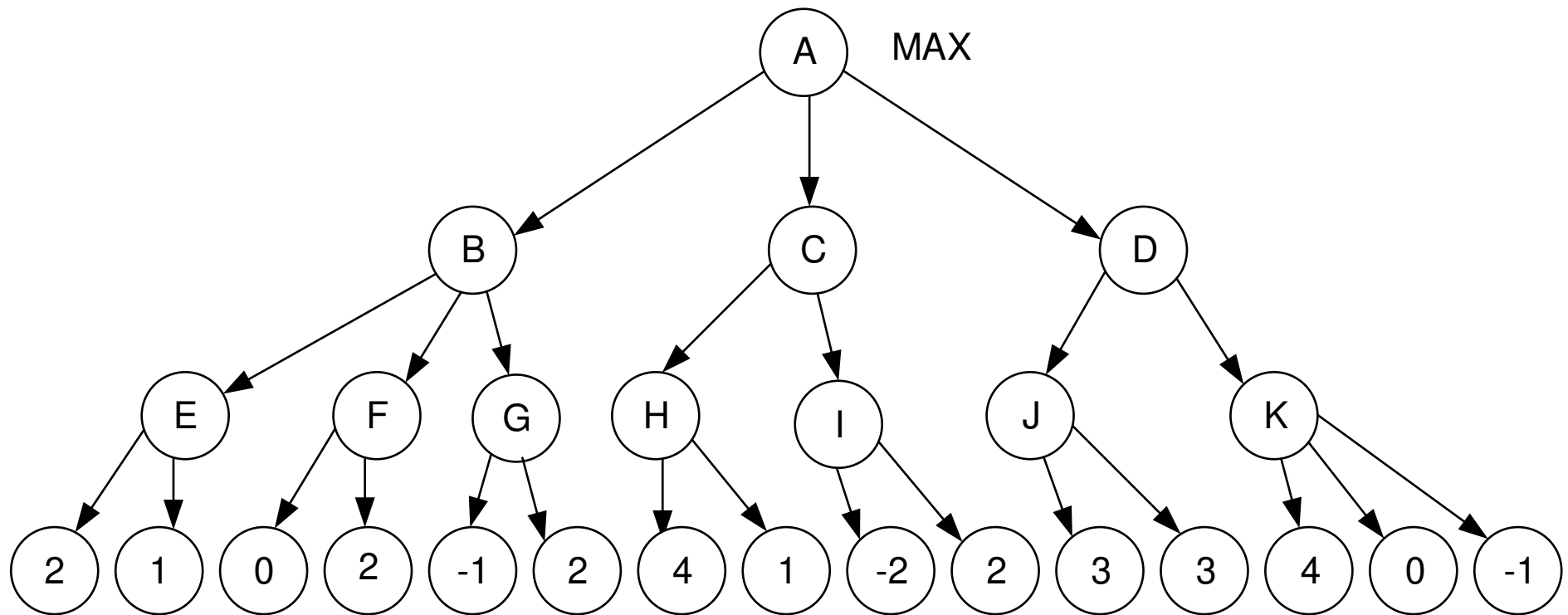


A) Més de 4 i menys de 7

B) Més de 6

C) 2

D) 4



A -i -i i \rightarrow B i -i i \rightarrow E -i -i i \rightarrow E 2 2 i \rightarrow B 2 -i 2 \rightarrow F -i -i 2 \rightarrow F 0 0 2
 \rightarrow **F 2 2 2** \rightarrow B 2 -i 2 \rightarrow G -i -i 2 \rightarrow G -1 -1 2 \rightarrow **G 2 2 2** \rightarrow B 2 -i 2
 \rightarrow A 2 2 i \rightarrow C i 2 i \rightarrow H -i 2 i \rightarrow H 4 4 i \rightarrow C 4 2 4 \rightarrow I -i 2 4 \rightarrow I -2 2
 4 \rightarrow I 2 2 4 \rightarrow **C 2 2 2** \rightarrow A 2 2 i \rightarrow D i 2 i \rightarrow J -i 2 i \rightarrow J 3 3 i \rightarrow D 3
 2 3 \rightarrow K -i 2 3 \rightarrow **K 4 4 3** i es poden les branques 2a i 3a de K

2022-11-04: Problema

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 destí 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` $\in \{\text{VAL, BAR, PAR, ROM, BER, OSL}\}$
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:

$$(ruta \text{ num_a } c_t^m)$$

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:

(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.

```
1 (defacts dades
2   (ruta 1 VAL BER)
3   (ruta 2 BAR ROM BER)
4   (ruta 3 VAL BAR PAR OSL)
5   (eut av 1 VAL fav av 2 BAR fav av 3 VAL fav paq BAR OSL paq ROM
    ↪   BER paq VAL BER) )
```

(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.

```
6 (defrule moure_av
7   (eut $?x av ?numa ?esta $?y)
8   (ruta ?numa $?z ?esta ?destí $?w)
9   => (assert (eut $?x av ?numa ?destí $?y)))
```

(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.

```
10 (defrule carregar_paquet_mateix_destí
11   (eut $?x av ?numa ?esta $?y paq ?esta ?dest $?h)
12   (ruta ?numa $? ?esta $? ?dest $?)
13   => (assert (eut $?x av ?numa ?esta ?dest $?y $?h)))
```

(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.

```
14 (defrule av_en_destí
15   (eut $?x av ?numa ?esta fav $?y)
16   (ruta ?numa $? ?esta)
17   => (printout t "L'avio " ?numa " ha arribat al seu destí final
      ↪   i no té paquets " crlf))
```

Primeres 37 línies de l'eixida:

```
1 <== f-0      (initial-fact)
2 ==> f-0      (initial-fact)
3 ==> f-1      (ruta 1 VAL BER)
4 ==> f-2      (ruta 2 BAR ROM BER)
5 ==> f-3      (ruta 3 VAL BAR PAR OSL)
6 ==> f-4      (eut av 1 VAL fav av 2 BAR fav av 3 VAL fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
7 ==> Activation 0      carregar_paquet_mateix_destí: f-4,f-1
8 ==> Activation 0      moure_av: f-4,f-3
9 ==> Activation 0      moure_av: f-4,f-2
10 ==> Activation 0     moure_av: f-4,f-1
11 ==> f-5      (eut av 1 BER fav av 2 BAR fav av 3 VAL fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
12 ==> Activation 0     av_en_destí: f-5,f-1
13 ==> Activation 0     moure_av: f-5,f-3
14 ==> Activation 0     moure_av: f-5,f-2
15 ==> f-6      (eut av 1 BER fav av 2 ROM fav av 3 VAL fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
16 ==> Activation 0     av_en_destí: f-6,f-1
17 ==> Activation 0     carregar_paquet_mateix_destí: f-6,f-2
18 ==> Activation 0     moure_av: f-6,f-3
19 ==> Activation 0     moure_av: f-6,f-2
20 ==> f-7      (eut av 1 BER fav av 2 BER fav av 3 VAL fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
21 ==> Activation 0     av_en_destí: f-7,f-2
22 ==> Activation 0     av_en_destí: f-7,f-1
23 ==> Activation 0     moure_av: f-7,f-3
24 ==> f-8      (eut av 1 BER fav av 2 BER fav av 3 BAR fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
25 ==> Activation 0     av_en_destí: f-8,f-2
26 ==> Activation 0     av_en_destí: f-8,f-1
27 ==> Activation 0     carregar_paquet_mateix_destí: f-8,f-3
28 ==> Activation 0     moure_av: f-8,f-3
29 ==> f-9      (eut av 1 BER fav av 2 BER fav av 3 PAR fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
30 ==> Activation 0     av_en_destí: f-9,f-2
31 ==> Activation 0     av_en_destí: f-9,f-1
32 ==> Activation 0     moure_av: f-9,f-3
33 ==> f-10     (eut av 1 BER fav av 2 BER fav av 3 OSL fav paq BAR OSL paq ROM BER paq VAL
  ↘ BER)
34 ==> Activation 0     av_en_destí: f-10,f-3
35 ==> Activation 0     av_en_destí: f-10,f-2
36 ==> Activation 0     av_en_destí: f-10,f-1
37 L'avio 1 ha arribat al seu destí final i no té paquets
```