

1r. Cognom

2n. Cognom

Nom

DNI

Especialitat:

Curs:

Grup:

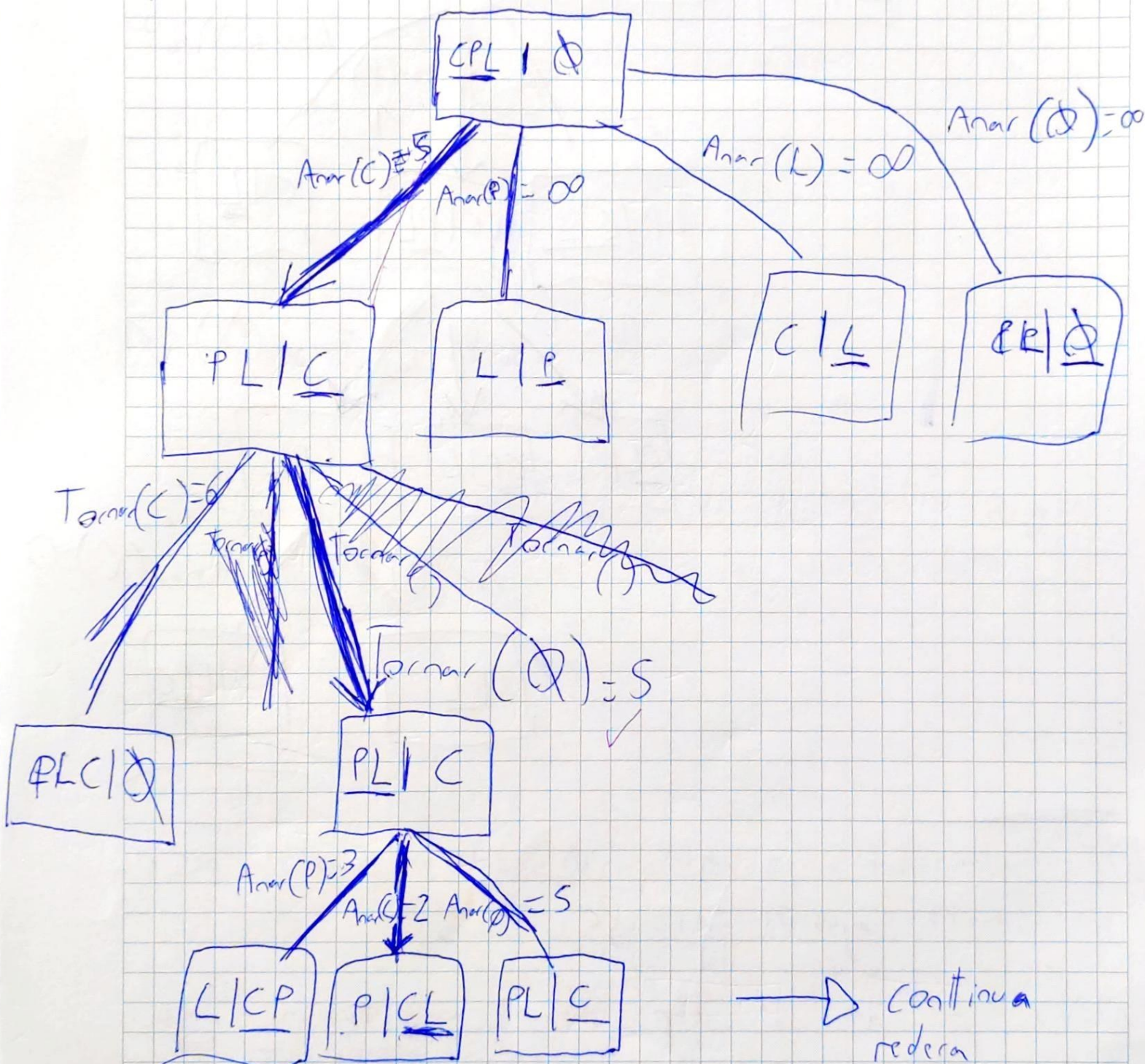
Assignatura: PROP

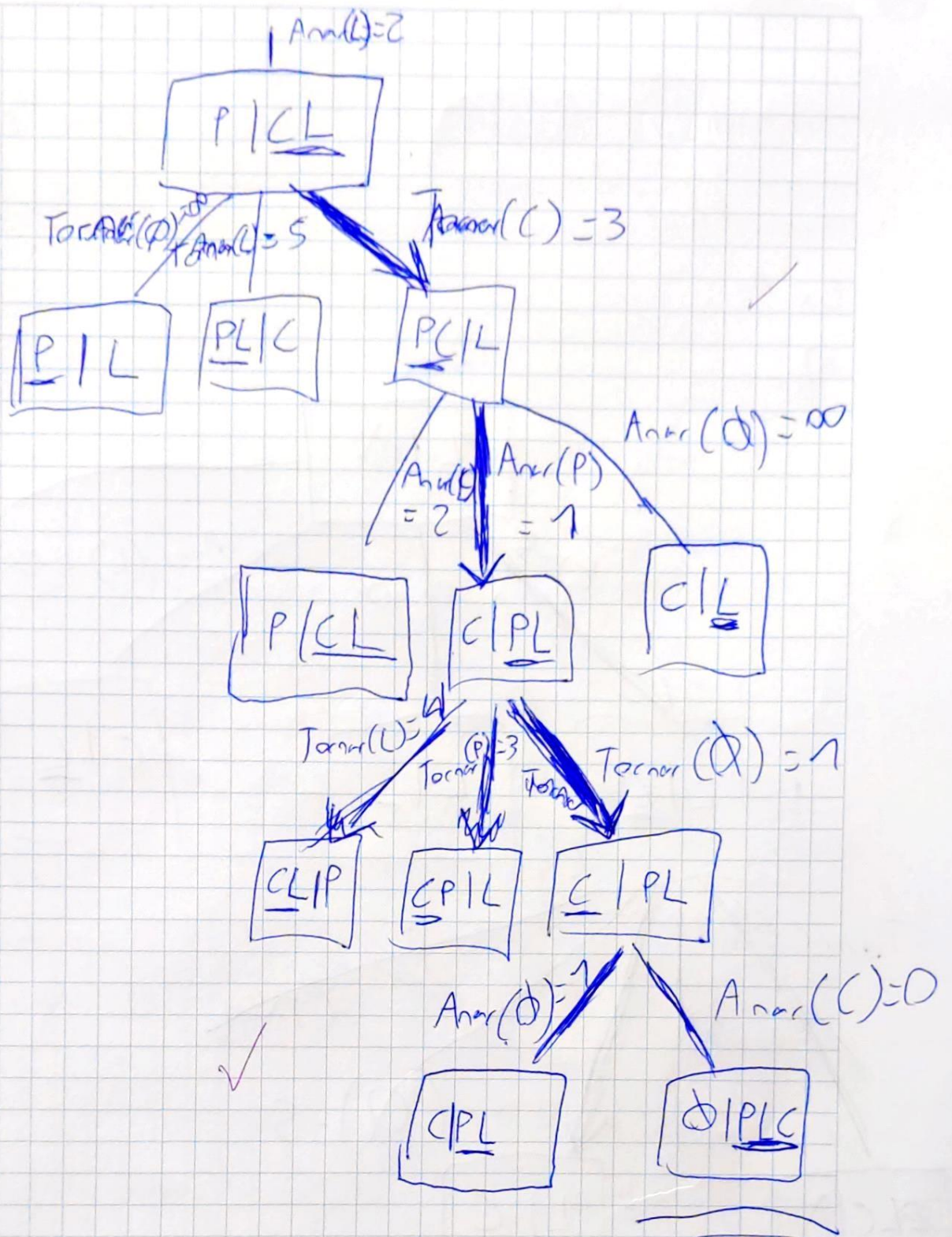
Data:

/ /

Ex 1

a)





b) És admissible ja que els valors donats per ella són molt ajustats al gc.

c) No ja que en cap moment es trobem amb cap cicle el que faria inútil el LNT.

d) Sabent que tindrà uns 3 fills per node i 8 nivells de profunditat amb cerca d'amplada recorriem aprox $\frac{3^8}{2}$ nodes.

1r. Cognom

GUZMAN

2n. Cognom

BOY

Nom

GUILLEM

DNI

48044457

Especialitat:

Curs:

Grup:

Assignatura:

PROP

Data:

/ /

ex 2
1)

Mov:

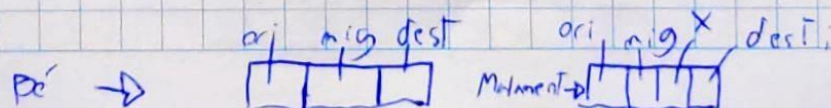
- Precondicions:
- S'ha de tenir el personatge en la casella d'origen seleccionada
 - No pot haver cap bloc en la casella destí seleccionada
 - Les caselles d'origen i destí han de ser veïnes, i assignar han d'estar de costat

Efectes:

- Desvincula el personatge de la casella origen
- Vincula el personatge a la casella destí

Salta:

- Precondicions:
- S'ha de tenir el personatge a la casella personatge.
 - No hi pot haver cap bloc a la casella destí
 - La casella del personatge, mig i destí han de ser veïnes.



Efectes:

- Desvincula el personatge de la casella origen.
- Vincula el personatge a la casella destí

2)

action EmpentaBloc

: parameters (?Casella_Personatge

?Casella_mig
~~?Casella_extra~~
?Casella_Destí - T-Casella)

: precondition

(and

(not (TeBloc ?Casella_Extra))

(TePersonatge ?Casella_Personatge) ✓

(not (TeBloc ?Casella_Destí)) ✓

(TeBloc ?Casella_mig) ✓

for

(and (EsVeina ?Casella_Personatge ?Casella_mig) ✓

(EsVeina ?Casella_mig ?Casella_Destí)) ✓

(and (EsVeina ?Casella_Destí ?Casella_mig) ✓

(EsVeina ?Casella_mig ?Casella_Personatge))

(and (EsVeina ?Casella_Destí ?Casella_extra) ✓

(EsVeina ?Casella_Destí ?Casella_mig)) X

)
: effect

(and

(not (TeBloc ?Casella_mig))

(TeBloc ?Casella_Destí))

(not (TePersonatge ?Casella_Personatge))

(TePersonatge ?Casella_mig))

)

)

1r. Cognom

2n. Cognom

Nom

DNI

GUEMAN

BOY

GUILLEM

48044453

Especialitat:

Curs:

Grup:

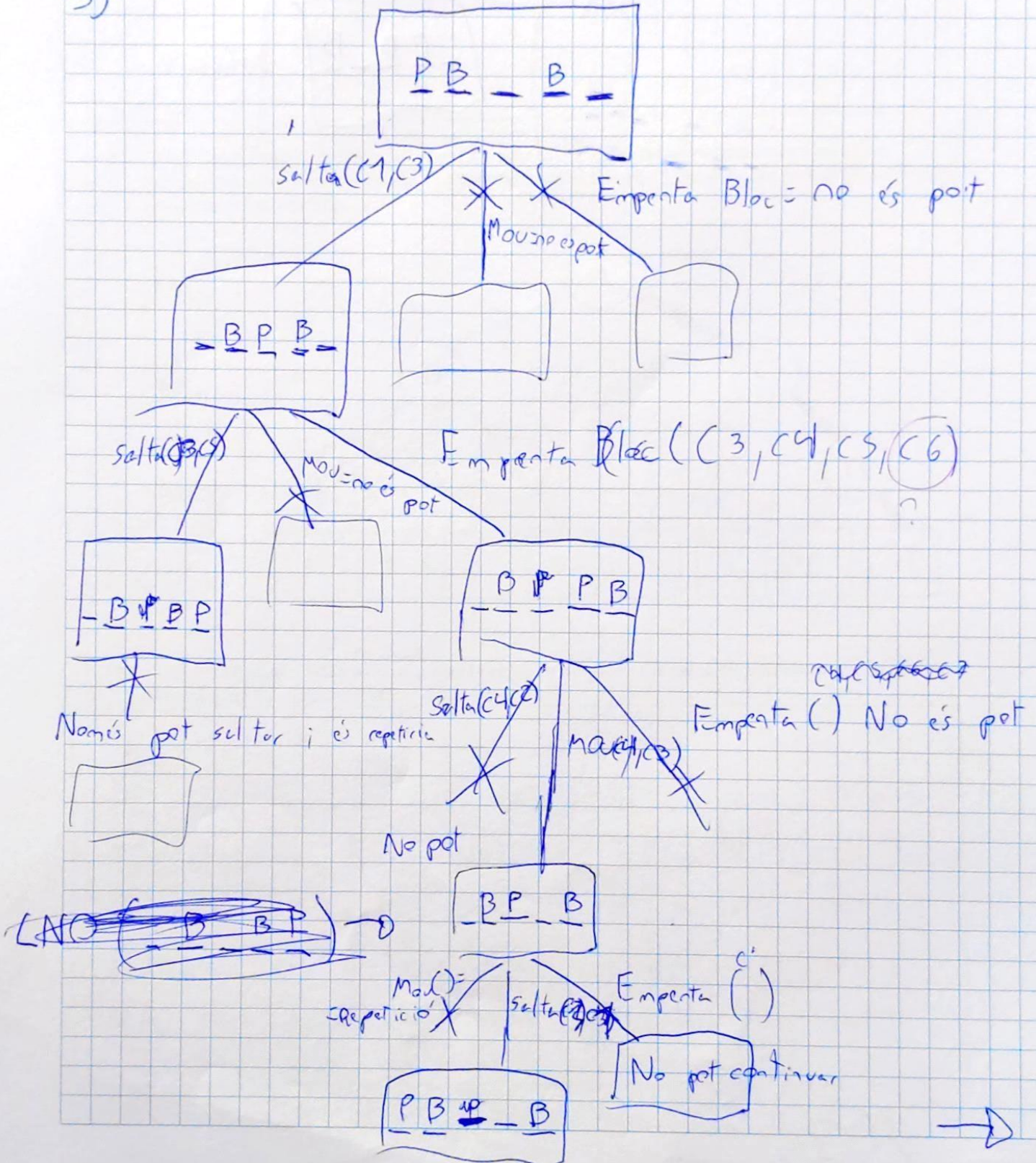
Assignatura: PROP

Data:

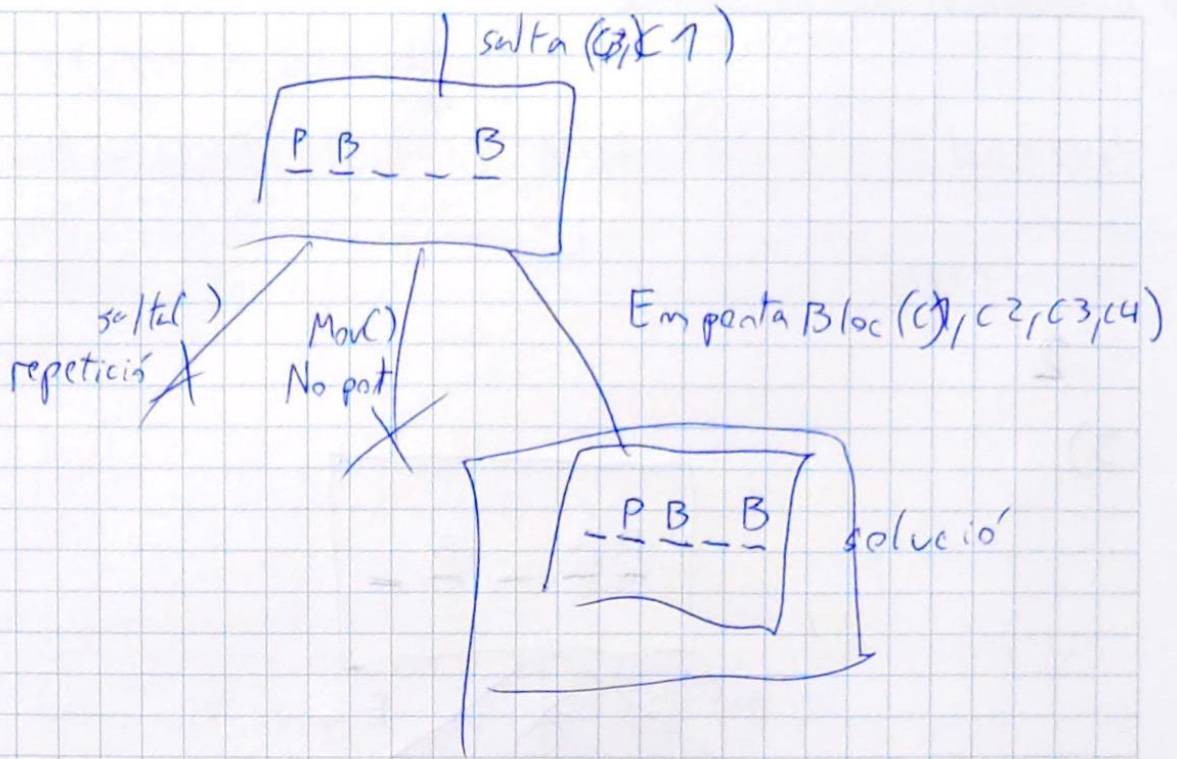
/ /

Ex 3

3)



~~LN4~~



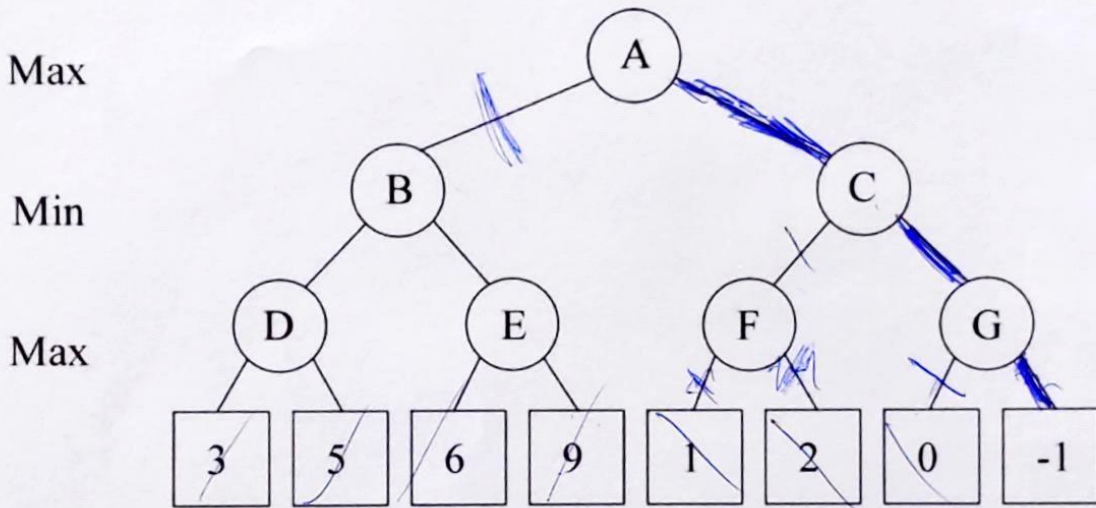
Problema 4: Teoria II [2.5 punts]

A^* = ordenats per $f_c + g_h$

1. [0.5p] Explica quines avantatges pot tenir un Beam-Search respecte una cerca A^* .

Beam-search pot trobar el cas millor més rapidament a l'buscar per profunditat en el cas de que la solució sea profunda.

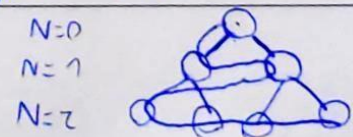
2. [1.5p] Donat l'arbre minimax proposat a continuació, amb els valors heurístics a les fulles, indica el moviment que es triaria, quines branques es tallarien i quins nodes no s'avaluarien si apliquem poda alfa-beta.



3. [0.5p] L'affirmació "IDS és millor que la cerca en amplada i la cerca en profunditat", en que és fonamenta? És sempre certa?

Es fonamenta en que la cerca iterada utilitza tan una cerca amb profunditat com d'amplada per funcionar. Expandeix els fills del nivell següent de profunditat, els ordena per millors valors i després els recorre. Això vol dir que sempre serà com a mínim millor que la cerca en amplada però si la solució és en un nivell profund serà més

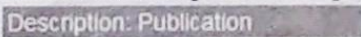
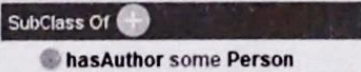
Optima la cerca en profunditat.



- 3) Donat el problema descrit i el domini, **incloent** la nova funció "EmpentaBloc", dibuixa l'arbre de cerca resultant d'una cerca en amplada (limitat a profunditat 2 sense comptar l'arrel) partint de l'estat inicial usant eliminació de cicles.
- Dibuixeu l'estat del joc en cada node .
 - Indiqueu l'acció i paràmetres en cada transició entre nodes.
 - Marqueu amb una creu els nodes eliminats per cicle.

Problema 3: Teoria I [2 punts][les preguntes errònies descomten el 50% del seu valor]

Marca les següents afirmacions com a certes o falses

1. $\log(n) \cdot n = O(n)$	F
2. La cerca en un arbre binari balancejat és $\Theta(\log_2(n))$	C
3. La cerca a una taula Hash té cost $\Theta(n)$	F
4. DTIME(n^2) són algorismes que requereixen temps quadràtic en una màquina de Turing no determinista.	
5. Els costos de memòria d'una cerca A^* usant una heurística admissible i monòtona són $O(b^m)$, on m és la profunditat de la solució i b és el factor de ramificació.	F C
6. En el 8-puzzle, l'heurística basada en distància de Manhattan és més eficient que la que compta caselles mal col·locades.	F
7. IDA* millora el temps d'execució de A^* .	F
8. Les xarxes neurals són sistemes d'aprenentatge no supervisats.	
9. Un arbre decisió és una tècnica de regressió, i no de classificació.	
10. Hem d'agrupar països segons les seves similituds en un conjunt de variables macroeconòmiques. K-means és un algorisme adequat.	C
11. Una xarxa neural d'una sola capa amb 4 entrades i 3 neurones té 12 paràmetres a entrenar.	
12. El overfitting o sobreajustament es dona quan tinc un excés de dades d'entrenament.	C
13. Si els valors {1, 2, 4} són mostres "sa" i {5, 3, 7} són mostres "malalt", el valor 3.7 és "malalt" segons l'algorisme K-Nearest Neighborhoods amb $K=3$.	
14. La cerca de Montecarlo pot tenir millors resultats que una cerca Minimax+heurística tot i basar-se en rollouts aleatoris.	C
15. En una ontologia hospitalària, les classes <i>Metge</i> i <i>Pacient</i> són disjunctes.	
16. En una ontologia de geometria, trobem la classe <i>Rombe</i> definida com "és Polígon" i "té 4 costats". La classe <i>Rectangle</i> està definida igual. Podem dir que són classes definides i que no són primitives.	F
17. Donada la següent definició, una publicació pot tenir 3 autors:	
 	

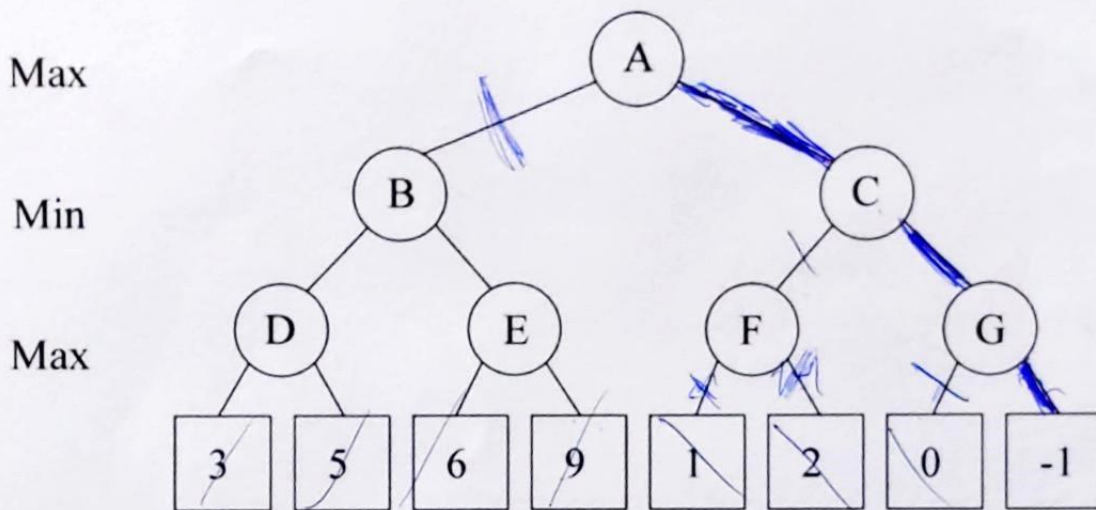
Problema 4: Teoria II [2.5 punts]

A^* = ordenats per $f_c + g_c$

1. [0.5p] Explica quines avantatges pot tenir un Beam-Search respecte una cerca A^* .

Beam-search pot trobar el cas millor més ràpidament a l'buscar per profunditat en el cas de que la solució sea profunda.

2. [1.5p] Donat l'arbre minimax proposat a continuació, amb els valors heurístics a les fulles, indica el moviment que es triaria, quines branques es tallarien i quins nodes no s'avaluarien si apliquem poda alfa-beta.



3. [0.5p] L'afirmació "IDS és millor que la cerca en amplada i la cerca en profunditat", en que és fonamenta? És sempre certa?

És fonamenta en que la cerca iterada utilitza tan una cerca amb profunditat com d'amplada per funcionar. Expandeix els fills del nivell següent de profunditat, els ordena per millors valors i després els recorre. Això vol dir que sempre serà com a mínim millor que la cerca en amplada però si la solució és en un nivell profund serà més

Optima la cerca en profunditat.

N=0
 N=1
 N=2

