Intel·ligència Artificial

Aula 2

Consultora: Jasmina Casals Terré

Alumne: Manuel Antonio Álvarez Araujo

PAC1: Representació de problemes

Sumem 15 és un joc de dos jugadors molt senzill. Inicialment disposem dels nombres de l'1 al 9 i dos jugadors que en torns alterns trien un número (primer un i després l'altre). A cada torn cada jugador tria un nombre entre els que queden per triar, i se'l queda (ja no el pot triar ningú més). Guanya el primer que, entre els nombres triats, tingui tres nombres que sumin 15.

Fixem-nos que a nosaltres no ens importa que sigui un joc. Tractarem el problema com un problema de formalització i cerca en un espai d'estats.

Es demana formalitzar aquest problema i contestar les preguntes enunciades als següents apartats:

1. Quina informació hi haurà a cada estat? Quants estats possibles hi haurà al graf d'estats? Tots els estats són accessibles des de qualsevol estat inicial?

En aquest cas concret hi ha 9 nombres i cada jugador comença sense cap nombre però **com a màxim podrà agafar 5** ja que si comencen agafant d'un a un i no en poden repetir cap, al final de tot, al torn 8 (com a màxim) un jugador tindria 5 nombres i l'altre 4.

Les regles diuen que guanya qui tingui **3 nombres que sumin 15**, dóna igual si en té 4 nombres que sumen més, mentre 3 d'ells sumin 15.

La combinació més petita que suma 15 seria (6 i 9) però no seria valida, en canvi (1, 5 i 9) sí.

La combinació més gran que suma 15 seria (1, 2, 3, 4 i 5) però cap combinació dels 3 nombres sumen 15, o sigui que tampoc seria valida.

Això vol dir que fins el torn 5 com a mínim no hi hauria cap guanyador, ja que és necessiten tres nombres que sumir 15, no es pot amb només dos.

Podem dir que **cada estat es compon de la llista de nombres del jugador A i la llista de nombres del jugador B**, per exemple: **(a1,a2,a3,a4,a5)(a6,a7,a8,a9)**

On a_i és un nombre entre 1 i 9, a més cap és pot repetir, a1 no por ser igual a2 o a9, per exemple.

També podem dir que si comença sempre el jugador A, **el jugador B mai podrà tenir 5 nombres**, si arribes el cas seria el jugador A qui agafaria el novè nombre i el jugador B es quedaria amb 4 com a màxim.

Seguint l'exemple anterior, un estat podria ser: (1,2,3,4,5)(6,7,8,9)

Però aquest estat no tindria guanyador ja que no hi ha combinació de 3 nombres que sumir 15, si hi ha combinacions de 2 o més, però justament de 3 no.

En teoria, **els estats possibles** de (a1,a2,a3,a4,a5)(a6,a7,a8,a9), sense repetir cap dels 9 nombres però en posicions diferents, **serien de 362.880** combinacions diferents, com podrien ser (1)(*null*), (5,6)(7) o (1,2,3,4)(9,5,7).

Tal i com hem definit l'estat, en teoria **tots els estats serien accessibles des de l'estat inicial (*null*)(*null*) on no s'ha escollit cap nombre encara**, si és un altre estat inicial lògicament hi hauria estats que no serien accessibles perquè no es pot repetir cap nombre ni esborrar cap escollit ja.

També cal que dir que, tot i que els estats siguin accessibles, si un jugador guanya la partida ja no caldria continuar generant més estats, per exemple, si el jugador B arriba al següent estat (2,3,4)(9,5,1) ja hauria guanyat i no tindria sentit continuar al següent torn, com per exemple (2,3,4,6)(9,5,1), això no el fa un estat inaccessible en teoria però si en la practica.

2. Quants operadors tindrem? Quins seran aquests operadors? Com relacionen els operadors els estats que s'han descrit més amunt?

Com només hi ha 9 nombres es podrien fer 9 operadors que fossin a un jugador o a un altre respectivament per torn, però crec que és millor **fer 2 operadors, un per jugador**, que agafin un nombre de la llista.

Per exemple:

jugadorA(llista_nombresA, nombres_lliures)

jugadorB(llista_nombresB, nombres_lliures)

"nombres_lliures" serien els nombres que encara no s'han utilitzat dels 9 disponibles.

"llista_nombresA" correspon als nombres escollits pel jugador A (mínim 0, màxim 5).

"llista_nombresB" correspon als nombres escollits pel jugador B (mínim 0, màxim 4).

Es podria parametritzar també el torn però si sempre comença el jugador A i després el B no caldria aquesta dada. **Cada operador agafa un nombre que encara no s'hagi utilitzat i el fica a la llista, el primer operador omple la primera llista de l'estat (a1,a2,a3,a4,a5) i el segon la segona llista de l'estat (a6,a7,a8,a9), mai al reves**.

3. Doneu la definició de l'estat en que A ha triat els nombres 2, 3 i 5 i B ha triat els nombres 4 i 8. Doneu també la definició de l'estat inicial on no hi ha cap nombre triat encara (segons la vostra representació) i descriviu com identificar l'estat objectiu.

La definició de l'estat seria: **(2,3,5)(4,8)**.

L'estat inicial, on no s'ha triat encara res, abans del primer torn, seria **(*null*)(*null*)** on "null" significa buit, res. També podríem posar **()()**.

Només **hi ha 48 combinacions de 3 nombres que puguin fer que un estat sigui objectiu**, o sigui, que tinguin 3 nombres dels 3, 4 o 5 que tingui el jugador i que sumin 15.

Per exemple, la combinació (9,5,1) o (7,6,2) serien part d'un estat objectiu, però també (9,5,1,2) o (7,6,2,1,3) ja que les dues tenen a dins la combinació correcte.

Per identificar l'estat objectiu **caldria una funció que fes la comprovació de que 3 dels nombres sumen 15 un cop fet el moviment de cada jugador**, la funció valdria pels dos jugadors i tindria que tenir en compte:

- Només s'ha de validar quan hi ha 3 o més nombres.
- Si hi ha més de 3 nombres ha de verificar les diferents combinacions fins trobar una que sumi 15, per exemple, si el jugador A té (9,5,2,1) primer faria la suma de 9+5+2 = 16 i com no seria correcte després faria la suma de 9+5+1 = 15.

Podem dir que la funció per identificar l'estat objectiu generaria per si mateixa un arbre d'estats fins trobar la combinació correcte.

Lògicament aquesta funció no s'executaria en els primers torns ja que cap jugador tindria 3 nombres per poder fer la comprovació.

4. Aplica l'algorisme de cerca en amplada a la situació inicial en que A ha triat els nombres 2, 3 i 5 i B ha triat els nombres 4 i 8. Dibuixa l'arbre de cerca i respon als apartats següents.

En ser un joc de dos jugadors cal tenir en compte que la cerca consistirà en buscar la primera tria que acaba la partida, és a dir, que li dóna a algun jugador la victoria (fixeu-vos que a l'arbre de cerca tindrem un nivell per jugador; això, però, no importa pel que es demana).

[Important: noteu que, tal i com està explicat el tema de cerca al material de l'assignatura, l'estat final s'identifica *quan s'agafa per generar els seus successors*, i no quan s'inclou a la llista de pendents. *En aquest problema, però, identificarem l'estat final quan s'inclou a la llista de pendents*].

a. En quin ordre heu aplicat els operadors sobre cada node?

L'arbre de cerca d'amplada és troba al final del document. Com l'estat inicial era (2,3,5)(4,8) **el torn és del jugador B, en aquest cas primer apliquem el seu operador** i com no trobem solució correcte el continuem aplicant amb la resta de nombres lliures fins que no quedin més. En aquest cas aquesta comprovació es pot fer dins del propi operador.

Un cop hem utilitzat tots els nombres lliures amb el jugador B, o sigui, hem fet una cerca horitzontal al mateix nivell, baixem un nivell i continuem amb el jugador A amb l'estat inicial (2,3,5)(4,8,9).

Després de fer dues operacions veiem que **el jugador A guanya la partida amb l'estat (2,3,5,7)(4,8,1)**.

b. Podeu estar segurs que la solució trobada és la més curta possible?

En realitat no ho es, perquè abans d'aquesta solució hem fet una operació amb el nombre 6 en comptes del 7, que no era la guanyadora. Però **com algorisme en teoria si que és la recerca de solució més curta** (6 passos).

c. Quants nodes heu generat? Què heu fet amb els repetits?

7 nodes. No s'ha repetit cap.

d. Quina ha estat la profunditat màxima a la que heu hagut d'arribar?

Si l'estat inicial és "n" hem arribat a "n+2" o nivell 2, la part que juga el jugador A.

Cua de pendents - Arbre
(2,3,5)(4,8)
(2,3,5)(4,8,1) · (2,3,5)(4,8,6) · (2,3,5)(4,8,7) · (2,3,5)(4,8,9)
(2,3,5)(4,8,6) · (2,3,5)(4,8,7) · (2,3,5)(4,8,9) · (2,3,5,6)(4,8,1) · (2,3,5,7)(4,8,1)

Arbre de cerca - Amplada