

Estructures de dades

Curs 2018-19

Pràctica II: Tres en ratlla

30 de març de 2019

Data d'entrega: 03 / 05 / 2019; 23:55 h

Aquesta pràctica consisteix en generar i mostrar l'**espai d'estats** pel joc **tres en ratlla**.

La realització d'aquesta pràctica pot ser individual o en parelles. L'entrega constarà d'una carpeta comprimida que contindrà el projecte GPS amb el codi font (en cas de no disposar de projecte GPS, heu d'entregar una carpeta comprimida amb els fitxers .ads i .adb).

Programa principal

El programa principal ha d'encarregar-se de:

1. Demanar a l'usuari que introdueixi per teclat l'estat del joc pel qual es vol generar l'espai d'estats.
2. Generar l'espai d'estats utilitzant el **TAD arbre ordinari** vist a classe de pràctiques.
3. Mostrar l'espai d'estats generats utilitzant el **recorregut en amplitud** de l'arbre ordinari vist a classe de pràctiques.

Estat del joc

L'estat del joc consisteix en el tauler amb els moviments realitzats juntament amb la informació del jugador que ha realitzat el darrer moviment:

```
type state is record
    jugador: integer;
    t: tauler;
end record;
```

Per exemple, un estat del joc podria ser:

jugador = 2

O	-	X
X	-	-
X	O	O

Obtenció de l'estat inicial

Per tal d'obtenir l'estat inicial per generar l'espai d'estats, l'usuari ha d'introduir per teclat un *String* de longitud 9 amb la informació de les peces que contenen cada una de les 9 caselles del tauler.

La primera posició de l'string ha de contenir la peça que ocupa la casella (1,1) del tauler, la segona posició de l'string ha de contenir la peça que ocupa la casella (1,2) del tauler, la tercera posició de l'string ha de contenir la peça que ocupa la casella (1,3) del tauler, la quarta posició de l'string ha de contenir la peça que ocupa la casella (2,1) del tauler; i així successivament.

Per exemple, l'string "O - X X - - X O O" hauria de generar el tauler:

O	-	X
X	-	-
X	O	O

La peça 'X' correspon a la peça del jugador 1, la peça 'O' correspon a la peça del jugador 2 i la peça '-' correspon a una casella desocupada del tauler.

A més de la informació de les peces ocupades per cada un dels jugadors al tauler, l'usuari també ha d'introduir quin jugador ha realitzat el darrer moviment (jugador 1, jugador 2).

Espai d'estats

Un espai d'estats d'un joc consisteix en l'arbre de tots els possibles estats que poden generar-se a partir d'aplicar tots els moviments possibles a un estat concret.

Per exemple, l'espai d'estats generat a partir de l'estat d'exemple de la secció *Estat del joc* seria:

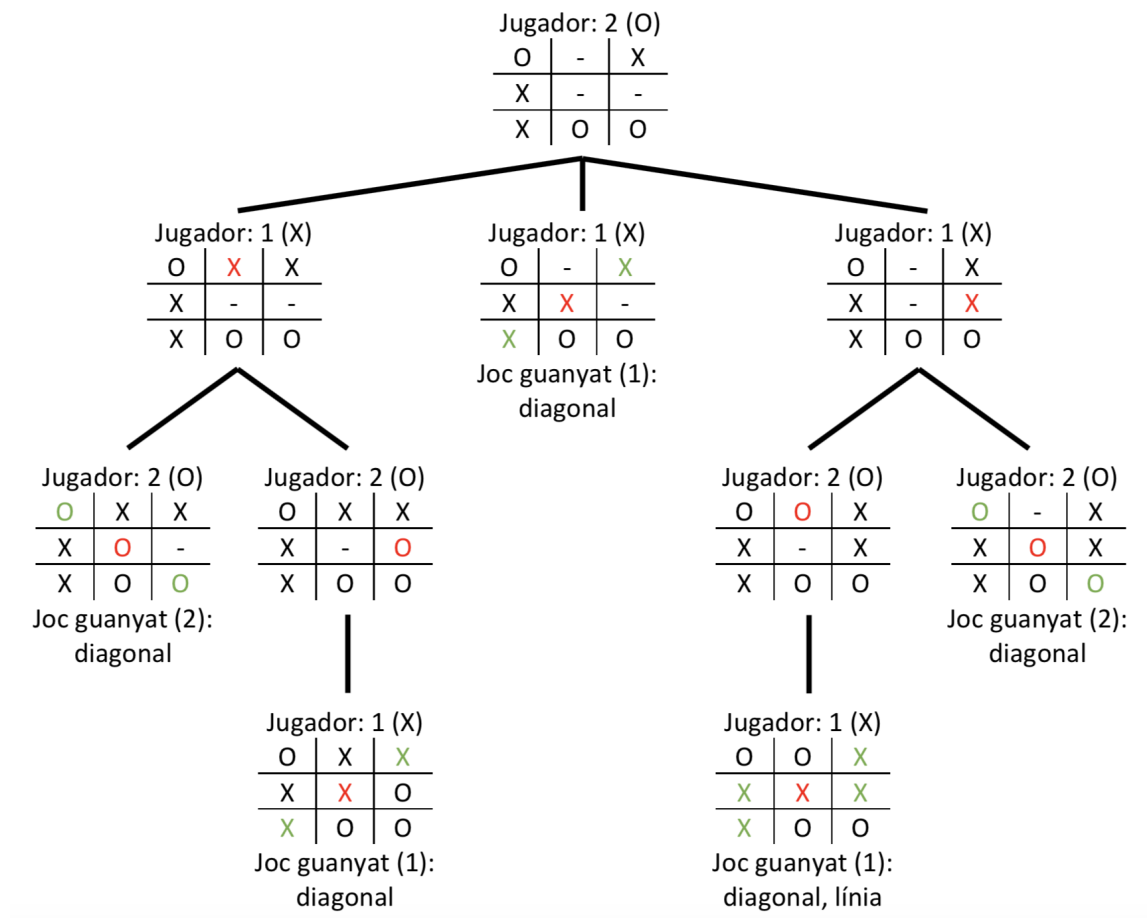


Figura 1: Exemple d'espai d'estats.

Recursos

Per tal de realitzar la pràctica, compteu amb els següents recursos:

- *dcola*: especificació i implementació del TAD cua vist a classe de teoria.
- *darbolordinario*: TAD arbre ordinari vist a classe de pràctiques.
- *dtauler*: paquet que vos proporciona el tipus *tauler* i les funcions associades amb la manipulació del tauler (**aquest tipus no és un TAD**). Els fitxers *dtauler.ads* i *dtauler.adb* es troben juntament amb l'enunciat de la pràctica. L'especificació del paquet es troba completa; de la implementació haureu de completar les funcions: *isJocGuanyat*, *isLinia*, *isDiagonal* i *isTaulerComplet*.

— *Funció que retorna si el tauler està complet:*

— *totes les caselles ocupades per peces 'X' o 'O'*

```
function isTaulerComplet(t: in tauler) return Boolean;
```

— *Funció que retorna si el tauler representa*

— *una disposició de peces corresponents*

— *a una situació guanyadora pel jugador 'jugador'*

```
function isJocGuanyat (t: in tauler; jugador: in integer)
return boolean;
```

— *Funció que retorna si el tauler conté*

— *una disposició de peces pel jugador 'jugador'*

— *que formin una diagonal (normal o inversa)*

```
function isDiagonal (t: in tauler; jugador: in integer)
return Boolean;
```

— *Funció que retorna si el tauler conté*

— *una disposició de peces pel jugador 'jugador'*

— *que formin una línia (horitzontal o vertical)*

```
function isLinia (t: in tauler; jugador: in integer)
return Boolean;
```

- *espaiestats*: fitxer corresponent al programa principal que conté exemples de les instàncies genèriques que possiblement necessitareu per implementar el programa principal.

Es demana

1. Implementar una cua (utilitzant cursors o punters).
 - (a) Ha de ser **genèrica**: ha de poder contenir ítems de qualsevol tipus de dades.
 - (b) Ha de ser un **TAD**.
2. Implementar un arbre ordinari (segons l'especificació vista a classe de pràctiques).
 - (a) Ha de ser **genèric**: ha de poder contenir ítems de qualsevol tipus de dades.
 - (b) Ha de ser un **TAD**.
3. Realitzar un programa principal que demani a l'usuari l'estat inicial del joc (estat de les peces del tauler i darrer jugador que ha realitzat un moviment) i generi l'espai d'estats corresponent a aquest estat inicial (un espai d'estats correspon a l'arbre de tots els possibles estats que poden generar-se a partir de l'estat inicial i els moviments possibles). Finalment, l'estat d'estats generat ha de mostrar-se mitjançant un recorregut en amplitud sobre l'arbre d'estats generat.