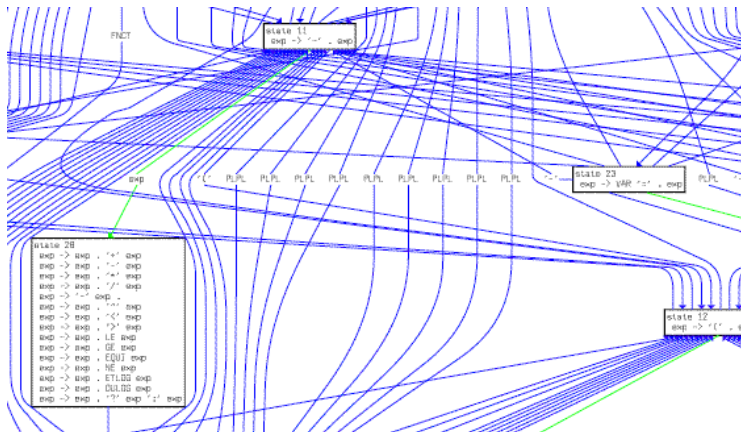


Quatrième cours

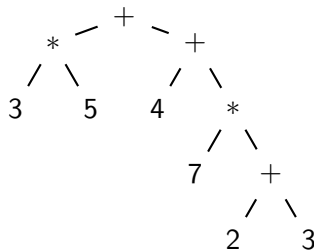
Arbres syntaxiques, grammaires



Grammaire pour des calculs

$$3 * 5 + 4 + 7 * (2 + 3)$$

- ▶ représentation sous forme d'un arbre.
- ▶ description de la méthode de calcul (priorité)



Grammaire naïve pour la calculatrice

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow (E)$$

$$E \rightarrow nb$$

$nb + * ()$: terminaux

E : expression

$$2 * 3 + 4 * 5$$

Grammaire naïve pour la calculatrice

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow (E)$$

$$E \rightarrow nb$$

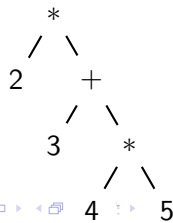
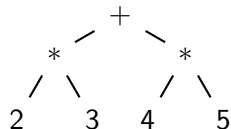
$nb + * ()$: terminaux

E : expression

Problème

Ne donne pas de priorité à la lecture. (Pas de parenthèses implicites)

$$2 * 3 + 4 * 5$$



Grammaire moins naïve

$$S \rightarrow S + P$$
$$S \rightarrow P$$
$$P \rightarrow P * N$$
$$P \rightarrow N$$
$$N \rightarrow (S)$$
$$N \rightarrow nb$$

$nb + * ()$: terminaux

E : expression

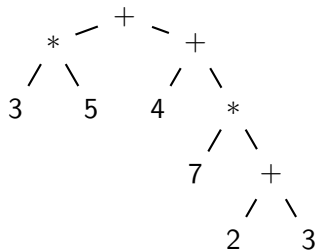
S : somme

P : produit

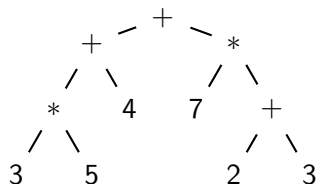
N : nombre

Plusieurs arbres pour une même formule

$$3 * 5 + 4 + 7 * (2 + 3)$$



$$(3 * 5) + (4 + (7 * (2 + 3)))$$



$$((3 * 5) + 4) + (7 * (2 + 3))$$

Quelle stratégie ?

En TP

Stratégie de parenthésage à droite :

- ▶ plus facile à parcourir dans un premier temps.
- ▶ pose des problèmes plus tard

$$3 - 4 - 5 \neq 3 - (4 - 5)$$

Plus tard, (poursuite d'études...) utilisation d'un automate et d'une pile.

XML, des arbres de données

```
1 <?xml version="1.1" encoding="UTF-8"?>
2 <!DOCTYPE bibliotheque SYSTEM "oc.dtd">
3 <bibliotheque>
4   <livre genre="aventure">
5     <titre>Le tour du monde en 80 jours</titre>
6     <auteur>Jules Verne</auteur>
7     <edition>Flammarion</edition>
8   </livre>
9   <livre genre="policier">
10    <titre>Le crime de l'orient express</titre>
11    <auteur>Agatha Christie</auteur>
12    <collection>Le masque</collection>
13  </livre>
14  <livre genre="theatre">
15    <titre>Le Cid</titre>
16    <auteur>Corneille</auteur>
17    <edition>Hachette éducation</edition>
18  </livre>
19  <livre genre="aventure">
20    <titre>Le hussard sur le toit</titre>
21    <auteur>Jean Giono</auteur>
22    <memeauteur>
23      <livre genre="roman">
24        <titre>Colline</titre>
25        <auteur>Jean Giono</auteur>
26      </livre>
27    </memeauteur>
28  </livre>
29 </bibliotheque>
```

Règles :

- ▶ En-tête (facultative)
- ▶ Une balise englobante.
- ▶ toute balise ouverte est fermée.
- ▶ Imbrication propre de balises (pas d'entrelacement)

→ arbre sémantique

Une grammaire pour XML

Spécifie des règles sur le contenu

- ▶ contenu des éléments
- ▶ types des attributs
- ▶ Très utile pour être sûr que le format des données est respecté.

L'exemple précédent

```
<?xml version="1.1" encoding="UTF-8"?>
<!DOCTYPE bibliotheque SYSTEM "oc.dtd">
<bibliotheque>
  <livre genre="aventure">
    <titre>Le tour du monde en 80 jours</titre>
    <auteur>Jules Verne</auteur>
    <edition>Flammarion</edition>
  </livre>
  <livre genre="policier">
    <titre>Le crime de l'orient express</titre>
    <auteur>Agatha Christie</auteur>
    <collection>Le masque</collection>
  </livre>
  <livre genre="theatre">
    <titre>Le Cid</titre>
    <auteur>Corneille</auteur>
    <edition>Hachette éducation</edition>
  </livre>
  <livre genre="aventure">
    <titre>Le hussard sur le toit</titre>
    <auteur>Jean Giono</auteur>
    <memeauteur>
      <livre genre="roman">
        <titre>Colline</titre>
        <auteur>Jean Giono</auteur>
      </livre>
    </memeauteur>
  </livre>
</bibliotheque>
```

```
<?xml version="1.1" encoding="UTF-8"?>
<!ELEMENT bibliotheque (livre)+>
<!ELEMENT livre (titre,auteur,(edition|collection|memeauteur)*)>
<!ATTLIST livre genre (roman|theatre|aventure|policier) #REQUIRED>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT collection (#PCDATA)>
<!ELEMENT memeauteur (livre)>
```

Syntaxe DTD

Description des balises

`<!ELEMENT balise contenu>`

avec le contenu pouvant être :

- ▶ `#PCDATA` pour du texte.
- ▶ `EMPTY` pour une balise vide.
- ▶ `ANY` pour du contenu libre.
- ▶ `(balise1, balise2...)` pour l'ensemble des balises dans l'ordre (et logique)
- ▶ `(balise1 | balise2...)` pour une balise parmi celles indiquées (ou logique)
- ▶ les caractères `?` `+` `*` ont le sens habituel.

Syntaxe DTD

Description des attributs

`<!ATTLIST balise attribut contenu type>`

le contenu étant (non exhaustif) :

- ▶ `#CDATA`
- ▶ `(att1|att2|att3)` pour l'un des attributs de la liste
- ▶ `ID` pour une chaîne identifiante.
- ▶ `IDREF` pour une référence vers un `ID`.

le type étant :

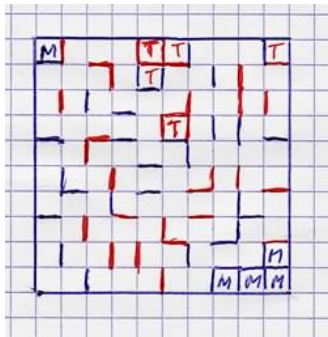
- ▶ `#REQUIRED` : obligatoire
- ▶ `#IMPLIED` : facultatif
- ▶ `#FIXED` : systématique pour l'élément.

Un petit jeu pour finir !

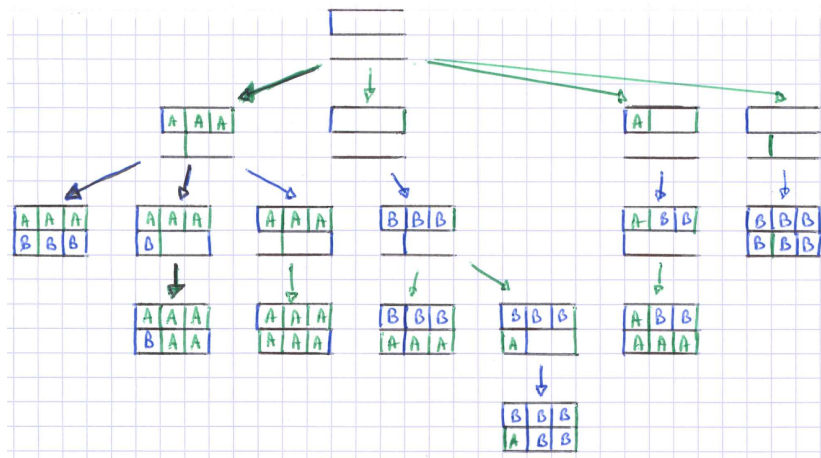
La Pipopipette (ou jeu des petits carrés)

Chacun son tour :

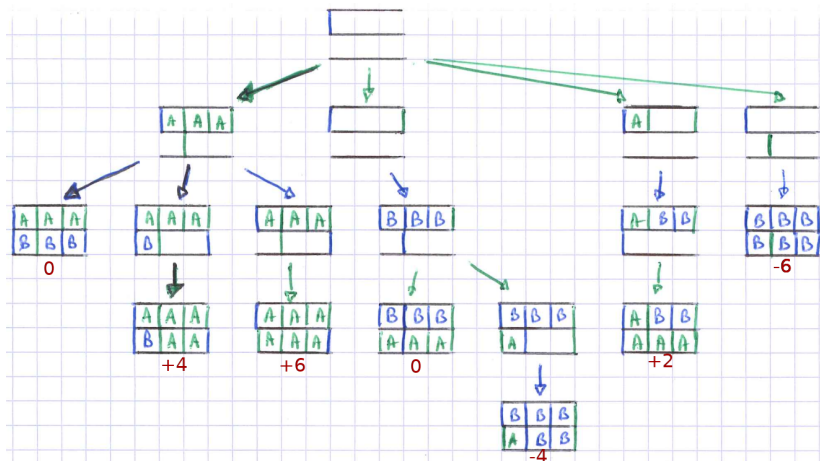
- ▶ tracer un coté d'une case
- ▶ si on ferme une case, +1pt et on rejoue.



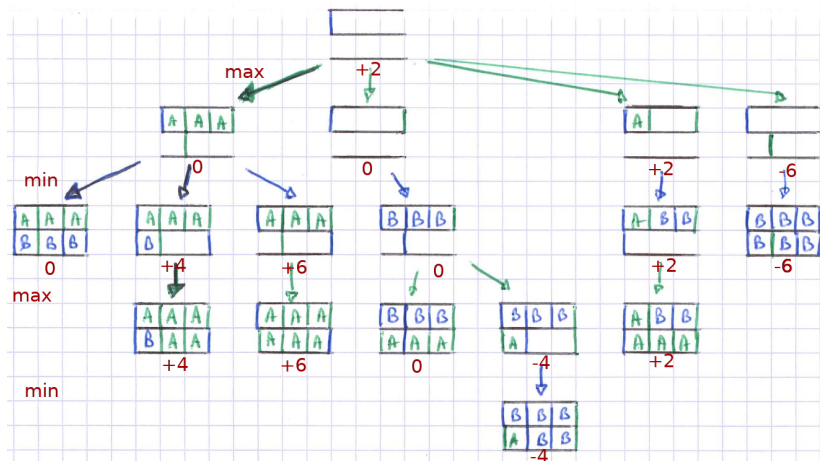
Arbre de stratégie



Arbre de stratégie



Arbre de stratégie



Méthode

Stratégie

- ▶ Pour Ann, choisir le coup qui maximise le score.
- ▶ Pour Bob, choisir le coup qui minimise le score.

Comment traiter cela ?

Solution 1 : Récursivité croisée

```
boolean scoreAnn(DotsAndBoxes game){
    // scoreBob(DotsAndBoxes game){
    if(game.isOver())
        return game.Score;
    else {
        Iterator<Move> iter = game.availableMoves();
        game.play(iter.next());
        best = scoreBob(game); // scoreAnn(game)
        game.cancelLast():
        while(iter.hasNext()){
            game.play(iter.next())
            best = max(best,          // best = min(best,
                        scoreBob(game)); // scoreAnn(game));
            game.cancelLast():
        }
        return best;
    }
}
```

Solution 2 : Par inversion

```
boolean scoreNext(DotsAndBoxes game){
    if(game.isOver())
        return game.Score;
    else {
        Iterator<Move> iter = game.availableMoves();
        game.play(iter.next());
        best = -scoreNext(game);
        game.cancelLast():
        while(iter.hasNext()){
            game.play(iter.next())
            best = max(best, -scoreNext(game));
            game.cancelLast():
        }
        return best;
    }
}
```

Couper des branches

Algorithme Alpha-beta

- Parfois inutile d'aller plus loin.

