# Compilation

Analyse syntaxique : Analyse descendante

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Mise en oeuvre d’un analyseur syntaxique

- L’analyseur syntaxique
  - Reçoit une suite d’unités lexicales de l’analyseur lexical
  - Doit dire si cette phrase (suite de mots) est syntaxiquement correcte
    - Essaie de construire l’arbre de dérivation associé
      - Si l'arbre est construit : phrase syntaxiquement correcte
      - Sinon : phrase syntaxiquement incorrecte

# Mise en oeuvre d'un analyseur syntaxique (suite)

- Deux approches pour construire l'arbre de dérivation
  - Approche descendante (top-down)
    - Partir de la grammaire pour retrouver la phrase en question
      - De la racine aux feuilles
  - Approche ascendante (bottom-up)
    - Partir de la phrase et remonter pour arriver aux règles de la grammaire
      - Des feuilles à la racine

# Mise en oeuvre d'un analyseur syntaxique (suite)

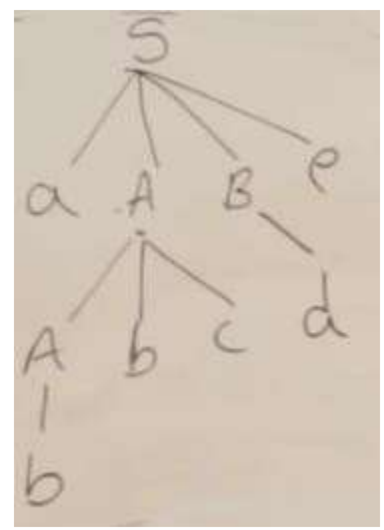
$S \rightarrow aABe$

$A \rightarrow Abc \mid b$

$B \rightarrow d$

Le mot à valider : abbcde

- Approche descendante (top-down)
  - De la racine aux feuilles



# Mise en oeuvre d'un analyseur syntaxique (suite)

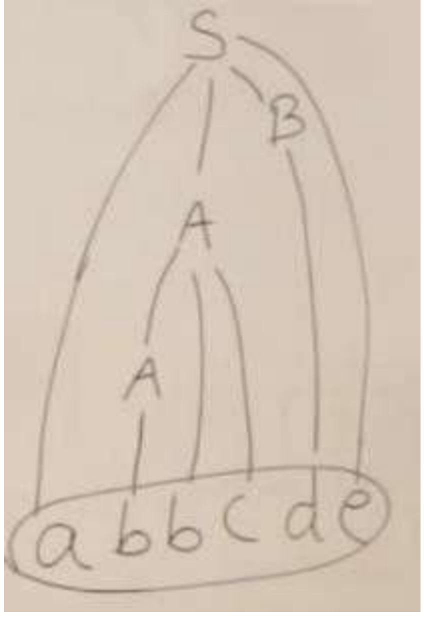
$S \rightarrow aABe$

$A \rightarrow Abc \mid b$

$B \rightarrow d$

Le mot à valider : abbcde

- Approche ascendante (bottom-up)
  - Des feuilles à la racine



# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Analyse descendante

- Objectif
  - En commençant du symbole de départ (axiome), retrouver la suite de règles nécessaires pour la construction de l'arbre de dérivation
- Principe de fonctionnement
  - Partir de l'axiome
    - À chaque pas :
      - Remplacer un non-terminal A par une expression $\alpha$ si $A \rightarrow \alpha$ est une règle de la grammaire
    - Arrêter lorsqu'il ne reste plus de non-terminaux

# Analyse descendante

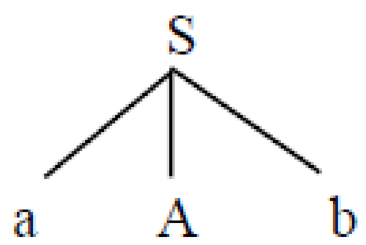
- Les non-terminaux permettent de décider quelle dérivation appliquer

- Exemple: $A \rightarrow aA \mid bA \mid \varepsilon$
  - On a trois règles,
  - Pour choisir laquelle appliquer,on regarde le terminal courant dans la phrase à valider
  - Dans le cas de la phrase "aab" l'ordre d'application des règles est:
    - $A \rightarrow aA \rightarrow aaA \rightarrow aabA \rightarrow aab\varepsilon = aab$

# Analyse descendante

- **Exercice:**
  - Soit la grammaire suivante:
    - S → aAb
    - A → cd | c
  - Phrase à valider :
    - w = "acb"

- La lecture de la première lettre (a) nous indique d’appliquer la règle S→ aAb




w=~~a~~cb

# Analyse descendante

- **Exercice:**
  - Soit la grammaire suivante:
    - $S \rightarrow aAb$
    - $A \rightarrow cd \mid c$
  - Phrase à valider :
    - w = "acb"

- La lecture de la deuxième lettre (c) nous indique que deux règles peuvent être appliquées :
  - $A \rightarrow cd$ et $A \rightarrow c$
- Pour savoir quelle règle appliquer, il faudra **lire la lettre suivante** (b)... ou avoir la possibilité de faire **des retours en arrière** : essayer la première règle $A \rightarrow cd$ et lorsqu'on aboutit à un échec, on revient en arrière pour appliquer la deuxième règle $A \rightarrow c$

# Analyse descendante

- Conclusion
  - La lecture d'une lettre n'est pas toujours suffisante pour décider quelle règle à appliquer
  - Pour savoir à tous les coups quelle règle à appliquer, une table indiquant le choix de la règle à un moment donné sera nécessaire
    - Cette table s'appelle: Table d'analyse

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Table d'analyse

- Table d'analyse LL
  - Caractéristiques
    - La table indique quelle transition (dérivation) effectuer lorsqu'on lit une lettre
    - Lecture des lettres de la gauche (Left : le premier L de LL) vers la droite
    - Dérivation des non-terminaux les plus à gauche (Left : le deuxième L de LL)

# Table d'analyse

- Table d'analyse LL
  - Pour construire la table, on a besoin de calculer deux types d'ensembles de terminaux :
    - PREMIER
    - SUIVANT

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# First()

- Calcul des ensembles PREMIER "First()"
  - Pour toute chaîne $\alpha$ composée de symboles terminaux et de non-terminaux on calcule PREMIER($\alpha$)
    - PREMIER($\alpha$) = terminaux pouvant commencer une chaîne dérivée de $\alpha$
      - a ϵ PREMIER($\alpha$) si et seulement si : $\alpha \xrightarrow{*} a\beta$ où $\beta$ peut être composé de terminaux et/ou de non-terminaux
      - ε ϵ PREMIER($\alpha$) si $\alpha \xrightarrow{*} \varepsilon$

# First()

- Exemple de calcul des ensembles PREMIER avec la grammaire:
  - S → Ba
  - B → cP|bP |P |ε
  - P → dS

- PREMIER(S)
  - On a S→Ba → εa = a, donc a ϵ PREMIER(S)
  - On a S→Ba→cPa ,donc c ϵ PREMIER(S)
  - On a S→Ba→bPa ,donc b ϵ PREMIER(S)
  - On a S→Ba→Pa→dSa ,donc d ϵ PREMIER(S)
  - Conclusion:
    - PREMIER(S)={a,b,c,d}

# First()

- PREMIER(aB)
  - $aB \overset{*}{\rightarrow} a\alpha$, d'où PREMIER(aB)={a}

- PREMIER(BS)
  - $B \rightarrow \varepsilon$ donc $BS \rightarrow S$, d'où PREMIER(S) est un sous-ensemble de PERMIER(BS)

# First()

- **Algorithme de calcul des PREMIER(X) avec X composé de terminaux et de non-terminaux**
  1. Si X est un non-terminal et $X \rightarrow Y_1Y_2...Y_n$ avec $Y_i$ symbole terminal ou non-terminal alors :
     1. Ajouter à l'ensemble PREMIER(X) tous les éléments de PREMIER($Y_1$) sauf ε
     2. Si ε∈PREMIER($Y_i$) pour i ∈[1..j] alors PREMIER($Y_{i+1}$) ⊂ PREMIER(X) sauf ε
     3. Si ε∈PREMIER($Y_i$) pour i ∈[1..n] alors ε∈PREMIER(X)
  2. Si X est un non-terminal et X→ε une production, alors
     1. ε∈PREMIER(X)
  3. Si X est un terminal, alors PREMIER(X) ={X}
  4. Recommencer jusqu'à ce qu'on ne trouve rien à ajouter aux ensembles PREMIERS

# First()

- Exemple d'application de l'algorithme
  - E→ TE'
  - E'→+TE' | -TE' | ε
  - T→FT'
  - T'→*FT' | /FT' | ε
  - F→(E) | nb

- PREMIER(E)
- PREMIER(T)
- PREMIER(E')
- PREMIER(F)
- PREMIER(T')

# First()

- Exemple d'application de l'algorithme
  - E→ TE'
  - E'→+TE' |-TE' |ε
  - T→FT'
  - T'→*FT' | /FT' | ε
  - F→(E) | nb

- PREMIER(E)
- PREMIER(T)
- PREMIER(E') <={+,-, ε}
- PREMIER(F) <={(,nb}
- PREMIER(T') <={*,/, ε}

- Algorithme de calcul des PREMIER(X) avec X composé de terminaux et de non-terminaux
  1. Si X est un non terminal et $X \rightarrow Y_1 Y_2 \ldots Y_n$ avec $Y_i$ symbole terminal ou non terminal alors :
     1. Ajouter à l'ensemble PREMIER(X) tous les éléments de PREMIER($Y_1$) sauf ε
     2. Si ε ∈ PREMIER($Y_i$) pour i ∈ [1..j] alors PREMIER($Y_{j+1}$) ⊂ PREMIER(X) sauf ε
     3. Si ε ∈ PREMIER($Y_i$) pour i ∈ [1..n] alors ε ∈ PREMIER(X)
  2. Si X est un non-terminal et X→ε une production, alors ε ∈ PREMIER(X)
  3. Si X est un terminal, alors PREMIER(X)={X}
  4. Recommencer jusqu'à ce qu'on ne trouve rien à ajouter dans les ensembles PREMIERS

# First()

- Exemple d'application de l'algorithme
  - E→ TE'
  - E'→+TE' | -TE' | ε
  - T→FT'
  - T'→*FT' | /FT' | ε
  - F→(E) | nb

- PREMIER(E) <= PREMIER(T) - {ε}
- PREMIER(T) <= PREMIER(F) - {ε}
- PREMIER(E’) <={+,-, ε}
- PREMIER(F) <={(,nb}
- PREMIER(T’) <={*,/, ε}

- Algorithme de calcul des PREMIER(X) avec X composé de terminaux et de non-terminaux
  1. Si X est un non terminal et $X \rightarrow Y_1 Y_2 \ldots Y_n$ avec $Y_i$ symbole terminal ou non terminal alors :
     1. Ajouter à l'ensemble PREMIER(X) tous les éléments de PREMIER($Y_1$) sauf ε
     2. Si ε ϵ PREMIER($Y_i$) pour i ϵ [1..j] alors PREMIER($Y_{i+1}$) ⊂ PREMIER(X) sauf ε
     3. Si ε ϵ PREMIER($Y_i$) pour i ϵ [1..n] alors ε ϵ PREMIER(X)
  2. Si X est un non-terminal et X→ε une production, alors ε ϵ PREMIER(X)
  3. Si X est un terminal, alors PREMIER(X) ={X}
  4. Recommencer jusqu'à ce qu'on ne trouve rien à ajouter dans les ensembles PREMIERS

# First()

- Exemple d'application de l'algorithme
  - E→ TE'
  - E'→+TE' | -TE' | ε
  - T→FT'
  - T'→*FT' | /FT' | ε
  - F→(E) | nb

- PREMIER(E) <= PREMIER(T) - {ε}
- PREMIER(T) <= PREMIER(F) - {ε} = {(,nb}
- PREMIER(E') <={+,-, ε}
- PREMIER(F) <={(,nb}
- PREMIER(T ') <={*,/, ε}

- Algorithme de calcul des PREMIER(X) avec X composé de terminaux et de non-terminaux
  1. Si X est un non terminal et $X \rightarrow Y_1 Y_2 \ldots Y_n$ avec $Y_i$ symbole terminal ou non terminal alors :
     1. Ajouter à l'ensemble PREMIER(X) tous les éléments de PREMIER($Y_1$) sauf ε
     2. Si ε ϵ PREMIER($Y_i$) pour i ϵ [1..j] alors PREMIER($Y_{j+1}$) ⊂ PREMIER(X) sauf ε
     3. Si ε ϵ PREMIER($Y_i$) pour i ϵ [1..n] alors ε ϵ PREMIER(X)
  2. Si X est un non-terminal et X→ε une production, alors ε ϵ PREMIER(X)
  3. Si X est un terminal, alors PREMIER(X) ={X}
  4. Recommencer jusqu'à ce qu'on ne trouve rien à ajouter dans les ensembles PREMIERS

# First()

- Exemple d'application de l'algorithme
  - E→ TE'
  - E'→+TE' | -TE' | ε
  - T→FT'
  - T'→*FT' | /FT' | ε
  - F→(E) | nb

- Algorithme de calcul des PREMIER(X) avec X composé de terminaux et de non-terminaux
  1. Si X est un non terminal et X → $Y_1 Y_2 \ldots Y_n$ avec $Y_i$ symbole terminal ou non terminal alors :
     1. Ajouter à l'ensemble PREMIER(X) tous les éléments de PREMIER($Y_1$) sauf ε
     2. Si ε ϵ PREMIER($Y_i$) pour i ϵ [1..j] alors PREMIER($Y_{j+1}$) ⊂ PREMIER(X) sauf ε
     3. Si ε ϵ PREMIER($Y_i$) pour i ϵ [1..n] alors ε ϵ PREMIER(X)
  2. Si X est un non-terminal et X→ε une production, alors ε ϵ PREMIER(X)
  3. Si X est un terminal, alors PREMIER(X) ={X}
  4. Recommencer jusqu'à ce qu'on ne trouve rien à ajouter dans les ensembles PREMIERS

- PREMIER(E) <= PREMIER(T) - {ε}= {(,nb}
- PREMIER(T) <= PREMIER(F) - {ε}= {(,nb}
- PREMIER(E') <={+,-, ε}
- PREMIER(F) <={(,nb}
- PREMIER(T ') <={*,/, ε}

# First()

- Exemple d'application de l'algorithme
  - S→ ABCDE
  - A→ a | ε
  - B→ b | ε
  - C→ c
  - D→ d | ε
  - E→ e | ε

- PREMIER(S)
- PREMIER(A)
- PREMIER(B)
- PREMIER(C)
- PREMIER(D)
- PREMIER(E)

# First()

- Exemple d'application de l'algorithme
  - S→ ABCDE
  - A→ a | ε
  - B→ b | ε
  - C→ c
  - D→ d | ε
  - E→ e | ε

- PREMIER(S) <={a, b, c}
- PREMIER(A) <={a, ε}
- PREMIER(B) <={b, ε}
- PREMIER(C) <={c}
- PREMIER(D) <={d, ε}
- PREMIER(E) <={e, ε}

# Follow()

- Calcul des ensembles SUIVANT "Follow()"
  - Pour tout symbole non-terminal A, on calcul SUIVANT(A)
    - SUIVANT(A) est l'ensemble des terminaux pouvant apparaître immédiatement à droite de A dans une dérivation $S \xrightarrow{*} \alpha A a \beta$
      - Dans la dérivation ci-dessus "a" est un élément de l'ensemble des suivants de A

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Follow()

- Exemple de calcul des ensembles SUIVANT avec la grammaire:
  - S → Ba | Sc
  - B → cP | bPb | P | ε
  - P → dS

- SUIVANT(S)
  - S→ Sc, d'où c ∈SUIVANT(S)
  - S→ Ba→Pa→dSa, d'où a ∈SUIVANT(S)
  - S → Ba→bPba→bdSba, d'où b ∈SUIVANT(S)

# Follow()

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production $A \rightarrow \alpha B \beta$ où B est un non-terminal, alors
    1. Ajouter le contenu de PREMIER($\beta$) dans SUIVANT(B) sauf $\varepsilon$
3. Pour chaque production $A \rightarrow \alpha B$, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production $A \rightarrow \alpha B \beta$ avec $\varepsilon \in$ PREMIER($\beta$), ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

# Follow()

- Exemple

$$\begin{cases} S \rightarrow aSb \mid cd \mid SAe \\ A \rightarrow aAdB \mid \varepsilon \\ B \rightarrow bb \end{cases}$$

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($ par exemple) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production A→ αBβ où B est un non-terminal, alors
   1. Ajouter le contenu de PREMIER(β) dans SUIVANT(B) sauf ε
3. Pour chaque production A→ αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production A→ αBβ avec ε є PREMIER(β) , ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

| | PREMIER | SUIVANT |
|---|---|---|
| S | a c | $ |
| A | a ε | |
| B | b | |

# Follow()

- Exemple

$$\begin{cases} S \to aSb|cd|SAe \\ A \to aAdB|\varepsilon \\ B \to bb \end{cases}$$

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($ par exemple) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production A→ αBβ où B est un non-terminal, alors
   1. Ajouter le contenu de PREMIER(β) dans SUIVANT(B) sauf ε
3. Pour chaque production A→ αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production A→ αBβ avec ε є PREMIER(β) , ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

| | PREMIER | SUIVANT |
|---|---|---|
| S | a c | $ b |
| A | a ε | |
| B | b | |

# Follow()

- Exemple

$$\begin{cases} \boxed{S \rightarrow aSb|cd|SAe} \\ A \rightarrow aAdB|\varepsilon \\ B \rightarrow bb \end{cases}$$

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($ par exemple) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production A→ αBβ où B est un non-terminal, alors
    1. Ajouter le contenu de PREMIER(β) dans SUIVANT(B) sauf ε
3. Pour chaque production A→ αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production A→ αBβ avec ε ϵ PREMIER(β) , ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

PREMIER(A) est inclus dans SUIVANT(S) sauf ε

| | PREMIER | SUIVANT |
|---|---|---|
| S | a c | $ b a |
| A | a ε | |
| B | b | |

# Follow()

- Exemple

$$\begin{cases} \boxed{S \to aSb|cd|SAe} \\ A \to aAdB|\varepsilon \\ B \to bb \end{cases}$$

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($ par exemple) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production A→ αBβ où B est un non-terminal, alors
   1. Ajouter le contenu de PREMIER(β) dans SUIVANT(B) sauf ε
3. Pour chaque production A→ αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production A→ αBβ avec ε є PREMIER(β) , ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

ε appartient à PREMIER(A) => e est inclus dans PREMIER(Ae)
=> e est inclus dans SUIVANT(S)
La règle appliquée est #2 et dans ce cas B est S, β est Ae et α=ε

| | PREMIER | SUIVANT |
|---|---|---|
| S | a c | $ b a e |
| A | a ε | |
| B | b | |

# Follow()

- Exemple

$$\begin{cases} S \rightarrow aSb|cd|SAe \\ A \rightarrow aAdB|\varepsilon \\ B \rightarrow bb \end{cases}$$

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($ par exemple) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production A→ αBβ où B est un non-terminal, alors
   1. Ajouter le contenu de PREMIER(β) dans SUIVANT(B) sauf ε
3. Pour chaque production A→ αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production A→ αBβ avec ε є PREMIER(β) , ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

| | PREMIER | SUIVANT |
|---|---|---|
| S | a c | $ b a e |
| A | a ε | e d |
| B | b | |

# Follow()

- Exemple

$$\begin{cases} S \rightarrow aSb|cd|SAe \\ A \rightarrow aAdB|\varepsilon \\ B \rightarrow bb \end{cases}$$

- Algorithme de calcul des suivants

1. Ajouter un marqueur de fin de chaîne ($ par exemple) à SUIVANT(S) où S est l'axiome de départ
2. Pour chaque production A→ αBβ où B est un non-terminal, alors
    1. Ajouter le contenu de PREMIER(β) dans SUIVANT(B) sauf ε
3. Pour chaque production A→ αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. Pour chaque production A→ αBβ avec ε ϵ PREMIER(β) , ajouter SUIVANT(A) à SUIVANT(B)
5. Recommencer à partir de l'étape 3 jusqu'à ce que les ensembles SUIVANT restent inchangés

SUIVANT(A) est inclus dans SUIVANT(B)

| | PREMIER | SUIVANT |
|---|---|---|
| S | a c | $ b a e |
| A | a ε | e d |
| B | b | e d |

# Follow()

- Exemple d'application de l'algorithme
  - S→ ABCDE
  - A→ a | ε
  - B→ b | ε
  - C→ c
  - D→ d | ε
  - E→ e | ε

- PREMIER(S) <={a, b, c}
- PREMIER(A) <={a, ε}
- PREMIER(B) <={b, ε}
- PREMIER(C) <={c}
- PREMIER(D) <={d, ε}
- PREMIER(E) <={e, ε}

- SUIVANT(S)
- SUIVANT(A)
- SUIVANT(B)
- SUIVANT(C)
- SUIVANT(D)
- SUIVANT(E)

# Follow()

- Exemple d'application de l'algorithme
  - S→ ABCDE
  - A→ a | ε
  - B→ b | ε
  - C→ c
  - D→ d | ε
  - E→ e | ε

- PREMIER(S) <={a, b, c}
- PREMIER(A) <={a, ε}
- PREMIER(B) <={b, ε}
- PREMIER(C) <={c}
- PREMIER(D) <={d, ε}
- PREMIER(E) <={e, ε}

- SUIVANT(S) <={$}
- SUIVANT(A) <={b, c}
- SUIVANT(B) <={c}
- SUIVANT(C) <={d, e, $}
- SUIVANT(D) <={e, $}
- SUIVANT(E) <={$}

# Exercice

- Exemple d'application de l'algorithme
  - S→ Bb|Cd
  - B→ aB |ε
  - C→ cC |ε

| | PREMIER | SUIVANT |
|---|---|---|
| S | | |
| B | | |
| C | | |

# Exercice

- Exemple d'application de l'algorithme
  - S→ Bb|Cd
  - B→ aB |ε
  - C→ cC |ε

| | PREMIER | SUIVANT |
|---|---|---|
| S | {a, b, c, d} | {$} |
| B | {a, ε} | {b} |
| C | {c, ε} | {d} |

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Table d'analyse

- Construction de la table d'analyse LL
  - Une table d'analyse est une matrice M
    - Dans la première colonne, on représente les non-terminaux
    - Dans la première ligne on place les terminaux et le symbole $ (sauf le mot vide)
    - Chaque case M[X,x] comporte la règle de production à appliquer
      - On utilise un algorithme pour remplir la matrice

# Table d'analyse

- Construction de la table d'analyse LL (suite)
  - Procédure de construction
    - Pour chaque production $A \rightarrow \alpha$ faire
      1. Pour tout $a \in$ PREMIER($\alpha$) (si a est différent de $\varepsilon$), rajouter la production $A \rightarrow \alpha$ dans la case M[A,a]
      2. Si $\varepsilon \in$ PREMIER($\alpha$), alors pour chaque $b \in$ SUIVANT(A) ajouter $A \rightarrow \alpha$ dans M[A,b]
    - Chaque case M[A,a] vide constitue une erreur syntaxique

# Table d'analyse

- Exemple

$$\begin{cases} E \rightarrow TE' \\ E' \rightarrow +TE' \mid -TE' \mid \varepsilon \\ T \rightarrow FT' \\ T' \rightarrow *FT' \mid /FT' \mid \varepsilon \\ F \rightarrow (E) \mid \text{nb} \end{cases}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | | | | | | | | |
| E' | | | | | | | | |
| T | | | | | | | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' | -TE' | \varepsilon \\ T \to FT' \\ T' \to *FT' | /FT' | \varepsilon \\ F \to (E) | \text{ nb} \end{cases}$$

$$\begin{array}{lll} \text{PREMIER}(E) & = \text{PREMIER}(T) & = \{(, \text{nb}\} \\ \text{PREMIER}(E') & = \{+, -, \varepsilon\} & \\ \text{PREMIER}(T) & = \text{PREMIER}(F) & = \{(, \text{nb}\} \\ \text{PREMIER}(T') & = \{*, /, \varepsilon\} & \\ \text{PREMIER}(F) & = \{(, \text{nb}\} & \end{array}$$

$$\begin{array}{ll} \text{SUIVANT}(E) & = \{ \ \$, ) \ \} \\ \text{SUIVANT}(E') & = \{ \ \$, ) \ \} \\ \text{SUIVANT}(T) & = \{ \ +, -, \ ), \$ \ \} \\ \text{SUIVANT}(T') & = \{ \ +, -, \ ), \$ \ \} \\ \text{SUIVANT}(F) & = \{ \ *, /, \ ), +, -, \$ \ \} \end{array}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | | | | | | | | |
| E' | | | | | | | | |
| T | | | | | | | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$
\begin{cases}
E \to TE' \\
E' \to +TE' \mid -TE' \mid \varepsilon \\
T \to FT' \\
T' \to *FT' \mid /FT' \mid \varepsilon \\
F \to (E) \mid \text{nb}
\end{cases}
$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$
\begin{aligned}
\text{PREMIER}(E) &= \text{PREMIER}(T) = \{(, \text{nb}\} \\
\text{PREMIER}(E') &= \{+, -, \varepsilon\} \\
\text{PREMIER}(T) &= \text{PREMIER}(F) = \{(, \text{nb}\} \\
\text{PREMIER}(T') &= \{*, /, \varepsilon\} \\
\text{PREMIER}(F) &= \{(, \text{nb}\}
\end{aligned}
$$

$$
\begin{aligned}
\text{SUIVANT}(E) &= \{ \$, ) \} \\
\text{SUIVANT}(E') &= \{ \$, ) \} \\
\text{SUIVANT}(T) &= \{ +, -, ), \$ \} \\
\text{SUIVANT}(T') &= \{ +, -, ), \$ \} \\
\text{SUIVANT}(F) &= \{ *, /, ), +, -, \$ \}
\end{aligned}
$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | | | | | | | |
| T | | | | | | | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' | -TE' | \varepsilon \\ T \to FT' \\ T' \to *FT' | /FT' | \varepsilon \\ F \to (E) | \text{ nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(, \text{nb}\} \\ \text{PREMIER}(E') &= \{+, -, \varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(, \text{nb}\} \\ \text{PREMIER}(T') &= \{*, /, \varepsilon\} \\ \text{PREMIER}(F) &= \{(, \text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{ \$, ) \} \\ \text{SUIVANT}(E') &= \{ \$, ) \} \\ \text{SUIVANT}(T) &= \{ +, -, ), \$ \} \\ \text{SUIVANT}(T') &= \{ +, -, ), \$ \} \\ \text{SUIVANT}(F) &= \{ *, /, ), +, -, \$ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | | | | | | |
| T | | | | | | | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- **Exemple**

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(, \text{nb}\} \\ \text{PREMIER}(E') &= \{+, -, \varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(, \text{nb}\} \\ \text{PREMIER}(T') &= \{*, /, \varepsilon\} \\ \text{PREMIER}(F) &= \{(, \text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{ \$, ) \} \\ \text{SUIVANT}(E') &= \{ \$, ) \} \\ \text{SUIVANT}(T) &= \{ +, -, ), \$ \} \\ \text{SUIVANT}(T') &= \{ +, -, ), \$ \} \\ \text{SUIVANT}(F) &= \{ *, /, ), +, -, \$ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | | |
| T | | | | | | | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(, \text{nb}\} \\ \text{PREMIER}(E') &= \{+, -, \varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(, \text{nb}\} \\ \text{PREMIER}(T') &= \{*, /, \varepsilon\} \\ \text{PREMIER}(F) &= \{(, \text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{\ \$, )\ \} \\ \text{SUIVANT}(E') &= \{\ \$, )\ \} \\ \text{SUIVANT}(T) &= \{\ +, -, ), \$\ \} \\ \text{SUIVANT}(T') &= \{\ +, -, ), \$\ \} \\ \text{SUIVANT}(F) &= \{\ *, /, ), +, -, \$\ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | | | | | | | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(, \text{nb}\} \\ \text{PREMIER}(E') &= \{+, -, \varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(, \text{nb}\} \\ \text{PREMIER}(T') &= \{*, /, \varepsilon\} \\ \text{PREMIER}(F) &= \{(, \text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{ \$, ) \} \\ \text{SUIVANT}(E') &= \{ \$, ) \} \\ \text{SUIVANT}(T) &= \{ +, -, ), \$ \} \\ \text{SUIVANT}(T') &= \{ +, -, ), \$ \} \\ \text{SUIVANT}(F) &= \{ *, /, ), +, -, \$ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | | | | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned}
\text{PREMIER}(E) &= \text{PREMIER}(T) = \{(, \text{nb}\} \\
\text{PREMIER}(E') &= \{+, -, \varepsilon\} \\
\text{PREMIER}(T) &= \text{PREMIER}(F) = \{(, \text{nb}\} \\
\text{PREMIER}(T') &= \{*, /, \varepsilon\} \\
\text{PREMIER}(F) &= \{(, \text{nb}\}
\end{aligned}$$

$$\begin{aligned}
\text{SUIVANT}(E) &= \{ \$, ) \} \\
\text{SUIVANT}(E') &= \{ \$, ) \} \\
\text{SUIVANT}(T) &= \{ +, -, ), \$ \} \\
\text{SUIVANT}(T') &= \{ +, -, ), \$ \} \\
\text{SUIVANT}(F) &= \{ *, /, ), +, -, \$ \}
\end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | | | T'→*FT' | | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(,\text{nb}\} \\ \text{PREMIER}(E') &= \{+,-,\varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(,\text{nb}\} \\ \text{PREMIER}(T') &= \{*,/,\varepsilon\} \\ \text{PREMIER}(F) &= \{(,\text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{\ \$,)\ \} \\ \text{SUIVANT}(E') &= \{\ \$,)\ \} \\ \text{SUIVANT}(T) &= \{\ +,-,\ ),\$\ \} \\ \text{SUIVANT}(T') &= \{\ +,-,\ ),\$\ \} \\ \text{SUIVANT}(F) &= \{\ *,/,\ ),+,-,\$\ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | | | T'→*FT' | T'→/FT' | | | |
| F | | | | | | | | |

# Table d'analyse

- Exemple

$$\begin{cases} E \rightarrow TE' \\ E' \rightarrow +TE' \mid -TE' \mid \varepsilon \\ T \rightarrow FT' \\ T' \rightarrow *FT' \mid /FT' \mid \varepsilon \\ F \rightarrow (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(,\text{nb}\} \\ \text{PREMIER}(E') &= \{+,-,\varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(,\text{nb}\} \\ \text{PREMIER}(T') &= \{*,/,\varepsilon\} \\ \text{PREMIER}(F) &= \{(,\text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{\ \$,)\ \} \\ \text{SUIVANT}(E') &= \{\ \$,)\ \} \\ \text{SUIVANT}(T) &= \{\ +,-,\ ),\$\ \} \\ \text{SUIVANT}(T') &= \{\ +,-,\ ),\$\ \} \\ \text{SUIVANT}(F) &= \{\ *,/,\ ),+,-,\$\ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | | | | | | | | |

# Table d'analyse

- **Exemple**

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$\begin{aligned} \text{PREMIER}(E) &= \text{PREMIER}(T) = \{(,\text{nb}\} \\ \text{PREMIER}(E') &= \{+,-,\varepsilon\} \\ \text{PREMIER}(T) &= \text{PREMIER}(F) = \{(,\text{nb}\} \\ \text{PREMIER}(T') &= \{*,/,\varepsilon\} \\ \text{PREMIER}(F) &= \{(,\text{nb}\} \end{aligned}$$

$$\begin{aligned} \text{SUIVANT}(E) &= \{\ \$,)\ \} \\ \text{SUIVANT}(E') &= \{\ \$,)\ \} \\ \text{SUIVANT}(T) &= \{\ +,-,\ ),\$\ \} \\ \text{SUIVANT}(T') &= \{\ +,-,\ ),\$\ \} \\ \text{SUIVANT}(F) &= \{\ *,/,\ ),+,-,\$\ \} \end{aligned}$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | | | | | | F→ (E) | | |

# Table d'analyse

- Exemple

$$
\begin{cases}
E \rightarrow TE' \\
E' \rightarrow +TE' \mid -TE' \mid \varepsilon \\
T \rightarrow FT' \\
T' \rightarrow *FT' \mid /FT' \mid \varepsilon \\
F \rightarrow (E) \mid \text{nb}
\end{cases}
$$

- Pour chaque production A→ α faire
  1. Pour tout a ε PREMIER(α) (et a est différent de ε), rajouter la production A→ α dans la case M[A,a]
  2. Si ε ε PREMIER(α) , alors pour chaque b ε SUIVANT(A) ajouter A → α dans M[A,b]
- Chaque case M[A,a] vide constitue une erreur syntaxique

$$
\begin{array}{lll}
\text{PREMIER}(E) & = \text{PREMIER}(T) & = \{(, \text{nb}\} \\
\text{PREMIER}(E') & = \{+, -, \varepsilon\} & \\
\text{PREMIER}(T) & = \text{PREMIER}(F) & = \{(, \text{nb}\} \\
\text{PREMIER}(T') & = \{*, /, \varepsilon\} & \\
\text{PREMIER}(F) & = \{(, \text{nb}\} &
\end{array}
$$

$$
\begin{array}{ll}
\text{SUIVANT}(E) & = \{ \ \$, ) \ \} \\
\text{SUIVANT}(E') & = \{ \ \$, ) \ \} \\
\text{SUIVANT}(T) & = \{ \ +, -, \ ), \$ \ \} \\
\text{SUIVANT}(T') & = \{ \ +, -, \ ), \$ \ \} \\
\text{SUIVANT}(F) & = \{ \ *, /, \ ), +, -, \$ \ \}
\end{array}
$$

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→ (E) | | |

# Table d'analyse

- Exercice

Donner la table d'analyse de la grammaire :

$S \rightarrow (S) \mid \varepsilon$

# Table d'analyse

- Exercice

Donner la table d'analyse de la grammaire :

S → (S) | ε

| | PREMIER | SUIVANT |
|---|---|---|
| S | (, ε | ), $ |

# Table d'analyse

- Exercice

Donner la table d'analyse de la grammaire :

$S \rightarrow (S) \mid \varepsilon$

| | PREMIER | SUIVANT |
|---|---|---|
| S | (, ε | ), $ |

| | ( | ) | $ |
|---|---|---|---|
| S | $S \rightarrow (S)$ | $S \rightarrow \varepsilon$ | $S \rightarrow \varepsilon$ |

Vérifier le mot : (()())

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Déterminer si un mot est accepté par une grammaire

- S'il existe un arbre de dérivation de l'axiome de départ au mot, alors le mot est accepté

- Pour générer l'arbre de dérivation, on utilise:
  - La table d'analyse,
  - Une pile et
  - Un algorithme

# Déterminer si un mot est accepté par une grammaire

- Algorithme :
  - Données d'entrée : le mot m avec $ à la fin et la table d'analyse M
  - Initialisation de la pile : $S où S est l'axiome de départ et un pointeur ps sur la première lettre du mot

```
repeter
    Soit X le symbole en sommet de pile
    Soit a la lettre pointée par ps
    Si X est un non terminal alors
        Si M[X,a] = X → Y_1 ... Y_n alors
            enlever X de la pile
            mettre Y_n puis Y_{n-1} puis ...puis Y_1 dans la pile
            émettre en sortie la production X → Y_1 ... Y_n
        sinon        (case vide dans la table)
            ERREUR
        finsi
    Sinon
        Si X =$ alors
            Si a = $ alors ACCEPTER
            Sinon ERREUR
            finsi
        Sinon
            Si X = a alors
                enlever X de la pile
                avancer ps
            sinon
                ERREUR
            finsi
        finsi
    finsi
jusqu'à ERREUR ou ACCEPTER
```

# Déterminer si un mot est accepté par une grammaire

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

m=3+4*5

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→(E) | | |

| Pile | Entrée | Sortie |
|---|---|---|
| $E | 3+4*5$ | E→TE' |
| $E'T | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

**repeter**
Soit $X$ le symbole en sommet de pile
Soit $a$ la lettre pointée par $ps$
**Si** $X$ est un non terminal **alors**
**Si** $M[X, a] = X \to Y_1 \ldots Y_n$ **alors**
enlever $X$ de la pile
mettre $Y_n$ puis $Y_{n-1}$ puis . . .puis $Y_1$ dans la pile
émettre en sortie la production $X \to Y_1 \ldots Y_n$
**sinon** (case vide dans la table)
ERREUR
**finsi**
**Sinon**
**Si** $X$ =$ **alors**
**Si** $a$ = $ **alors** ACCEPTER
**Sinon** ERREUR
**finsi**
**Sinon**
**Si** $X = a$ **alors**
enlever $X$ de la pile
avancer $ps$
**sinon**
ERREUR
**finsi**
**finsi**
**finsi**
**jusqu'à** ERREUR ou ACCEPTER

# Déterminer si un mot est accepté par une grammaire

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

m=3+4*5

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→ (E) | | |

| Pile | Entrée | Sortie |
|---|---|---|
| $E | 3+4*5$ | E→TE' |
| $E'T | 3+4*5$ | T→FT' |
| $E'T'F | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

**repeter**
  Soit $X$ le symbole en sommet de pile
  Soit $a$ la lettre pointée par $ps$
  **Si** $X$ est un non terminal **alors**
    **Si** $M[X, a] = X \to Y_1 \dots Y_n$ **alors**
      enlever $X$ de la pile
      mettre $Y_n$ puis $Y_{n-1}$ puis ...puis $Y_1$ dans la pile
      émettre en sortie la production $X \to Y_1 \dots Y_n$
    **sinon** (case vide dans la table)
      ERREUR
    **finsi**
  **Sinon**
    **Si** $X$ =$ **alors**
      **Si** $a$ = $ **alors** ACCEPTER
      **Sinon** ERREUR
      **finsi**
    **Sinon**
      **Si** $X = a$ **alors**
        enlever $X$ de la pile
        avancer $ps$
      **sinon**
        ERREUR
      **finsi**
    **finsi**
  **finsi**
**jusqu'à** ERREUR ou ACCEPTER

# Déterminer si un mot est accepté par une grammaire

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

m=3+4*5

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→(E) | | |

| Pile | Entrée | Sortie |
|---|---|---|
| $E | 3+4*5$ | E→TE' |
| $E'T | 3+4*5$ | T→FT' |
| $E'T'F | 3+4*5$ | F→nb |
| $E'T'3 | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

```
repeter
    Soit X le symbole en sommet de pile
    Soit a la lettre pointée par ps
    Si X est un non terminal alors
        Si M[X, a] = X → Y1 ... Yn alors
            enlever X de la pile
            mettre Yn puis Yn−1 puis ...puis Y1 dans la pile
            émettre en sortie la production X → Y1 ... Yn
        sinon        (case vide dans la table)
            ERREUR
        finsi
    Sinon
        Si X =$ alors
            Si a = $ alors ACCEPTER
            Sinon ERREUR
            finsi
        Sinon
            Si X = a alors
                enlever X de la pile
                avancer ps
            sinon
                ERREUR
            finsi
        finsi
    finsi
jusqu'à ERREUR ou ACCEPTER
```

# Déterminer si un mot est accepté par une grammaire

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

m=3+4*5

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→ (E) | | |

| Pile | Entrée | Sortie |
|---|---|---|
| $E | 3+4*5$ | E→TE' |
| $E'T | 3+4*5$ | T→FT' |
| $E'T'F | 3+4*5$ | F→nb |
| $E'T'3 | 3+4*5$ | |
| $E'T' | 3+4*5$ | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

**repeter**
Soit $X$ le symbole en sommet de pile
Soit $a$ la lettre pointée par $ps$
**Si** $X$ est un non terminal **alors**
**Si** $M[X, a] = X \to Y_1 \dots Y_n$ **alors**
enlever $X$ de la pile
mettre $Y_n$ puis $Y_{n-1}$ puis ...puis $Y_1$ dans la pile
émettre en sortie la production $X \to Y_1 \dots Y_n$
**sinon** (case vide dans la table)
ERREUR
**finsi**
**Sinon**
**Si** $X$ =$ **alors**
**Si** $a$ = $ **alors** ACCEPTER
**Sinon** ERREUR
**finsi**
**Sinon**
**Si** $X = a$ **alors**
enlever $X$ de la pile
avancer $ps$
**sinon**
ERREUR
**finsi**
**finsi**
**finsi**
**jusqu'à** ERREUR ou ACCEPTER

# Déterminer si un mot est accepté par une grammaire

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \mathrm{nb} \end{cases}$$

m=3+4*5

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→ (E) | | |

| Pile | Entrée | Sortie |
|---|---|---|
| $E | 3+4*5$ | E→TE' |
| $E'T | 3+4*5$ | T→FT' |
| $E'T'F | 3+4*5$ | F→nb |
| $E'T'3 | 3+4*5$ | |
| $E'T' | 3+4*5$ | T'→ε |
| $E' | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

```
repeter
    Soit X le symbole en sommet de pile
    Soit a la lettre pointée par ps
    Si X est un non terminal alors
        Si M[X, a] = X → Y1 ... Yn alors
            enlever X de la pile
            mettre Yn puis Yn−1 puis ...puis Y1 dans la pile
            émettre en sortie la production X → Y1 ... Yn
        sinon        (case vide dans la table)
            ERREUR
        finsi
    Sinon
        Si X =$ alors
            Si a = $ alors ACCEPTER
            Sinon ERREUR
            finsi
        Sinon
            Si X = a alors
                enlever X de la pile
                avancer ps
            sinon
                ERREUR
            finsi
        finsi
    finsi
jusqu'à ERREUR ou ACCEPTER
```

# Déterminer si un mot est accepté par une grammaire

- Exemple

$$\begin{cases} E \to TE' \\ E' \to +TE' \mid -TE' \mid \varepsilon \\ T \to FT' \\ T' \to *FT' \mid /FT' \mid \varepsilon \\ F \to (E) \mid \text{nb} \end{cases}$$

m=3+4*5

| | nb | + | - | * | / | ( | ) | $ |
|---|---|---|---|---|---|---|---|---|
| E | E→TE' | | | | | E→TE' | | |
| E' | | E'→+TE' | E'→-TE' | | | | E'→ε | E'→ε |
| T | T→FT' | | | | | T→FT' | | |
| T' | | T'→ε | T'→ε | T'→*FT' | T'→/FT' | | T'→ε | T'→ε |
| F | F→nb | | | | | F→(E) | | |

| Pile | Entrée | Sortie |
|---|---|---|
| $E | 3+4*5$ | E→TE' |
| $E'T | 3+4*5$ | T→FT' |
| $E'T'F | 3+4*5$ | F→nb |
| $E'T'3 | 3+4*5$ | |
| $E'T' | 3+4*5$ | T'→ε |
| $E' | 3+4*5$ | E'→ +TE' |
| $E'T+ | 3+4*5$ | |
| $E'T | 3+4*5$ | T→FT' |
| $E'T'F | 3+4*5$ | F→nb |
| $E'T'4 | 3+4*5$ | |
| $E'T' | 3+4*5$ | T'→*FT' |
| $E'T'F* | 3+4*5$ | |
| $E'T'F | 3+4*5$ | F→nb |
| $E'T'5 | 3+4*5$ | |
| $E'T' | 3+4*5$ | T'→ε |
| $E' | 3+4*5$ | E'→ε |
| $ | 3+4*5$ | ACCEPTER |

**repeter**
    Soit $X$ le symbole en sommet de pile
    Soit $a$ la lettre pointée par $ps$
    **Si** $X$ est un non terminal **alors**
        **Si** $M[X,a] = X \to Y_1 \dots Y_n$ **alors**
            enlever $X$ de la pile
            mettre $Y_n$ puis $Y_{n-1}$ puis ...puis $Y_1$ dans la pile
            émettre en sortie la production $X \to Y_1 \dots Y_n$
        **sinon** (case vide dans la table)
            ERREUR
        **finsi**
    **Sinon**
        **Si** $X$ =$ **alors**
            **Si** $a$ = $ **alors** ACCEPTER
            **Sinon** ERREUR
            **finsi**
        **Sinon**
            **Si** $X = a$ **alors**
                enlever $X$ de la pile
                avancer $ps$
            **sinon**
                ERREUR
            **finsi**
        **finsi**
    **finsi**
**jusqu'à** ERREUR ou ACCEPTER

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Grammaire LL(1)

- L'algorithme précédent ne s'applique qu'aux grammaires dites LL(1)
- C'est quoi une grammaire LL(1) ?
  - Une grammaire LL(1) est une grammaire où une cellule de la table d'analyse ne contient pas plus qu'une règle de production

Intuition: une grammaire est LL(1) si la lecture d'un seul symbole terminal permet la sélection de la règle à appliquer sans ambigüité (dans la table d'analyse)

# Grammaire LL(1)

- Exemple de grammaire qui n'est pas LL(1)

$$\begin{cases} S \rightarrow aAb \\ A \rightarrow cd|c \end{cases}$$

| | $a$ | $c$ | $b$ | $d$ | $\$$ |
|---|---|---|---|---|---|
| $S$ | $S \rightarrow aAb$ | | | | |
| $A$ | | $A \rightarrow cd$<br>$A \rightarrow c$ | | | |

Une grammaire est dite LL(K) si le choix de la règle à appliquer nécessite la lecture de K symboles terminaux

# Grammaire LL(1)

- Y a-t-il des grammaires qui sont LL(1) par défaut?
  - Si les productions de chaque non-terminal commencent par un symbole terminal différent, alors la grammaire est LL(1)

# Grammaire LL(1)

- Quelles sont les grammaires qui sont par défaut non LL(1)?
  - Une grammaire immédiatement récursive à gauche n’est pas LL(1)
  - Une grammaire ambigüe n'est pas LL(1)

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Récursivité à gauche immédiate

- Récursivité à gauche immédiate
  - Une grammaire est dite immédiatement récursive à gauche si elle contient un non-terminal A tel qu'il existe une règle de production $A \rightarrow A\alpha$ où $\alpha$ est une chaîne quelconque

$$\begin{cases} S \rightarrow ScA \mid B \\ A \rightarrow Aa \mid \varepsilon \\ B \rightarrow Bb \mid d \mid e \end{cases}$$

# Récursivité à gauche immédiate

- Élimination de la récursivité à gauche immédiate

Remplacer toute règle de la forme $A \rightarrow A\alpha|\beta$ par

$$\begin{cases} A \rightarrow \beta A' \\ A' \rightarrow \alpha A'|\varepsilon \end{cases}$$

- Exemple:
  - S→ScA|B => S→BS’,S’→cAS’|ε
  - A→Aa| ε => A→A’, A’→aA’| ε
  - B→Bb|d|e => B→dB’|eB’, B’→ bB’|ε

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - Résumé
  - Analyse ascendante

# Enlever l'ambigüité d'une grammaire

- Pour enlever l'ambigüité d'une grammaire, on utilise la technique de factorisation à gauche des parties communes d'une grammaire

- Exemple:
  - Grammaire ambigüe :
    - ExpIf → if Cond then Inst
    - ExpIf → if Cond then Inst else Inst
    - …
  - Après transformation en introduisant un nouveau non-terminal:
    - ExpIf → if Cond then Inst A
    - A → else Inst | ε
    - …

# Plan

- Analyseurs syntaxiques
  - Analyse descendante
    - Table d'analyse LL
    - Calcul des ensembles PREMIER
    - Calcul des ensembles SUIVANT
    - Construction de la table LL
    - Déterminer si un mot est accepté par une grammaire
    - Grammaire LL(1)
    - Récursivité à gauche immédiate
    - Enlever l'ambigüité d'une grammaire
    - **Résumé**
  - Analyse ascendante

# Résumé

- Approche de dérivation descendante:
  - Table d'analyse LL
  - Pile
  - Algorithme
- Pour rendre une grammaire LL(1):
  - Éliminer la récursivité à gauche (voir aussi l'élimination de l' ambigüité dans la partie 01 du cours sur l'analyse syntaxique)
  - Factoriser les parties communes

# Références


- A.Aho, R. Sethi, J.Ullman,"Compilateurs : principes, techniques et outils". Deuxième édition, Pearson Education. 2007
- Andrew W.Appel and Jens Palsberg,"Modern Compiler Implementation in Java, Second Edition". Second Edition, Cambridge University Press, 2002
- Ronald Mak, "Writing Compilers and Interpreters: A Modern Software Engineering Approach Using Java". Third Edition, John Wiley & Sons, 2009