ANALYSE LL(1)

Schwarzentruber / Legendre
p 70 → 77.

L'analyse LL(1) est une méthode d'analyse syntaxique, c-à-d dont le but est de déterminer si un texte dérive d'une grammaire ou non ; et souvent aussi de donner dans le cas positif un arbre de dérivation correspondant.

Plus précisément il s'agit ici d'une analyse descendante c-à-d qu'on va chercher à dériver l'axiome de la grammaire (S) jusqu'à obtenir le texte (si possible).
Ce faisant on réalise une "left-most derivation", une dérivation la plus à gauche ce qui signifie que le non terminal que l'on dérive à chaque étape est celui le + à gauche.
C'est ce que signifie le 2ème L.

Le premier L est là pour "left to right", qui signifie qu'on lit notre texte de gauche à droite, sous-entendu sens retour arrière et une seule fois.
Enfin le 1 indique qu'on pourra résoudre des conflits, se choisir quelle règle de dérivation appliquer, en lisant 1 caractère.
On s'appuiera sur les premier(α) pour les règles X→α et sur les suivant(X) pour les règles X→ε. On dira que notre grammaire est LL 1 si on peut appliquer cette analyse c-à-d si la connaissance de premier et suivant avec le caractère lu permettent de trancher, de choisir quelle dérivation appliquer.
C'est ce que formalise la définition suivante

Déf Soit $G = (\Sigma, \Gamma, R, S)$ une grammaire algé. pointée.
G est LL 1 ⇔ pour toute variable $X \in \Gamma$ les ensembles premier(α) de chaque règle $X \to \alpha$, et suivant(X) si la règle $(X \to \varepsilon) \in R$, sont 2 à 2 disjoints

ex G : $S \to b \mid Ta$, $T \to a \mid \varepsilon$

premier(b) = {b}
premier(Ta) = {α, a}
premier(a) = {a}
premier(ε) = {α}

suivant(S) = {$}
suivant(T) = {a}

Ici premier(a) ∩ suivant(T) ≠ ∅
Si on est dans "l'état courant" T, et qu'on lit un a on ne pourra pas choisir entre T→ε ; T→a

G n'est pas LL 1

ex

G : $S \to bTd$ | $T \to a \mid \varepsilon$

premier (b) = {b}
premier (Td) = {a, d}
premier (a) = {a}
premier (ε) = {ε}

suivant (S) = {$\widetilde{\ }$}
suivant (T) = {d}.

Pour S : premier (b) ∩ premier (Td) = ∅ ok ! (pas besoin de suivant(S) car la règle $S \to \varepsilon$ n'apparaît pas dans G)

Pour T : premier (a) ∩ premier (ε) = ∅
premier (a) ∩ suivant (T) = ∅
premier (ε) ∩ suivant (T) = ∅
ok

G est LL 1

Algo

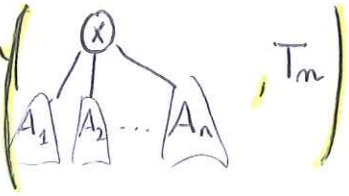
Pour G une grammaire LL 1 fixée.

LL1_aux (A : arbre, t : texte)

match rac(A) with

| $a \in \Sigma$ ->
  si $t_1 = a$
  alors retourner ([a], $t_2 \dots t_{|t|}$)
  sinon ERREUR

| $X \in \Gamma$ ->
  si $t = \varepsilon$
  alors
    si $\alpha \in p(X)$ et $\widetilde{\ } \in s(X)$
    alors soit α tel que $X \to \alpha$ et $\alpha \in p(\alpha)$
    sinon ERREUR
  sinon // $t \neq \varepsilon$
    si $\alpha \in p(X)$ et $t_1 \in s(X)$
    alors soit α tel que $X \to \alpha$ et $\alpha \in p(\alpha)$
    sinon s'il existe α' tq $X \to \alpha'$ et $t_1 \in p(\alpha')$
      alors $\alpha' \leftarrow \alpha$
      sinon ERREUR.

  si $\alpha = \varepsilon$
  alors retourner ((X) — [ε] , t)
  sinon $T_0 \leftarrow t$
    pour i allant de 1 à $n = |\alpha|$
      $(A_i, T_i) \leftarrow$ LL1_aux ([α_i], T_{i-1})
    retourner ( , T_n)

LL 1 (t : texte)

(A, T) = LL1_aux ((S), t)

Si T ≠ E
alors ERREUR
sinon retourner A .

ex

G : S → U | V
U → uU | E
V → vW | WU
W → wW | e

Calcul de premier

p(E) = {α} ; p(u) = {u} , p(v) = {v} , p(W) = {w}, p(e) = {e}

	p_0	p_1	p_2	p_3	p_4
S	∅	{α}	{α, v}	{α, v, u, e}	{α, v, u, e, w}
U	{α}	{α}	{α, u}	{α, u}	—
V	∅	{v}	{v, e}	{v, w, e}	—
W	∅	{e}	{w, e}	{w, e}	—
uU	∅	{u}	{u}	{u}	—
vW	∅	{v}	{v}	{v}	—

d'où
p(S) = {α, u, v, w, e}
p(U) = {α, u}
p(V) = {v, w}
p(W) = {w}
p(uU) = {u}
p(vW) = {v}

Calcul de suivant

	s_0	s_1	s_2	s_3
S	{$}	{$}	{$}	= s_2
U	∅	{$}	{$}	
V	∅	{$}	{$}	
W	∅	{u}	{u, $}	

premier(U), sans α ∪ s_0(V)
premier(U), sans α ∪ s_1(V)

d'où
s(S) = {$}
s(U) = {$}
s(V) = {$}
s(W) = {u, $}

Vérification

- p(U), p(V) sont 2 à 2 disjoints
- p(uU), p(E) et s(U) _____
- p(v), p(WU) _____
- p(vW), p(e) _____

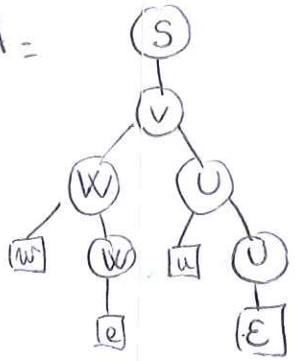
donc G est bien LL 1

LL$_1$(weu)

- LL$_1$-aux(S, weu)
  - $\alpha = V$
  - LL$_1$-aux(V, weu)
    - $\alpha = WU$
    - LL$_1$-aux(W, weu)
      - $\alpha = wW$
      - LL$_1$-aux(w, weu) = (w, eu)
      - LL$_1$-aux(W, eu)
        - $\alpha = e$
        - LL$_1$-aux(e, eu) = (e, u)
        - = (W(w, We), u)
      - = (W(w, W(e)), u)
    - LL$_1$-aux(U, u)
      - $\alpha = u$
      - LL$_1$-aux(u, u) = (u, ε)
      - LL$_1$-aux(U, ε)
        - $\alpha = \varepsilon$
        - = (U(ε), ε)
      - = (U(u, U(ε)), ε)
    - = (V(W(w, W(e)), U(u, U(ε))), ε)
  - = (A, ε)
  - retourner A

finalement LL$_1$(weu) renvoit A car ε

où A =



FAIRE COMPLEXITÉ