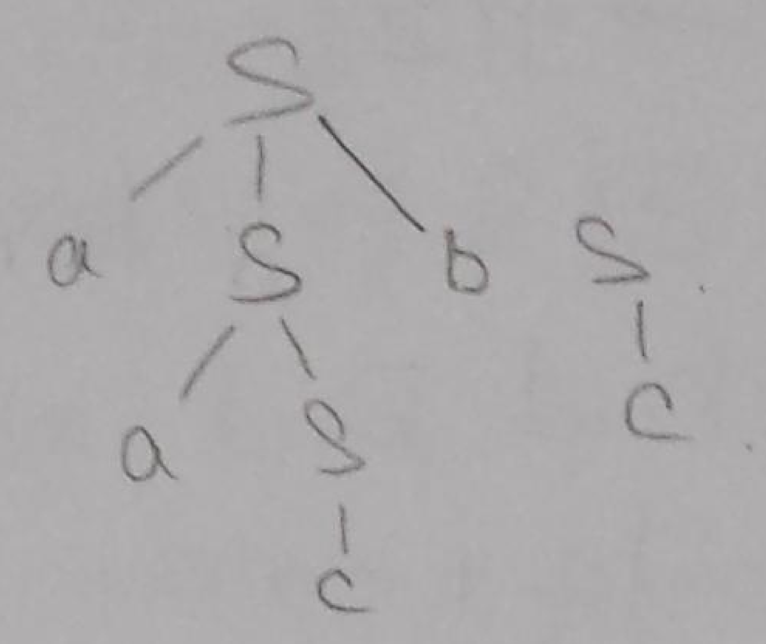
$\{ (e, i, \alpha, \beta)$ if $A = S$ and $i = 1$

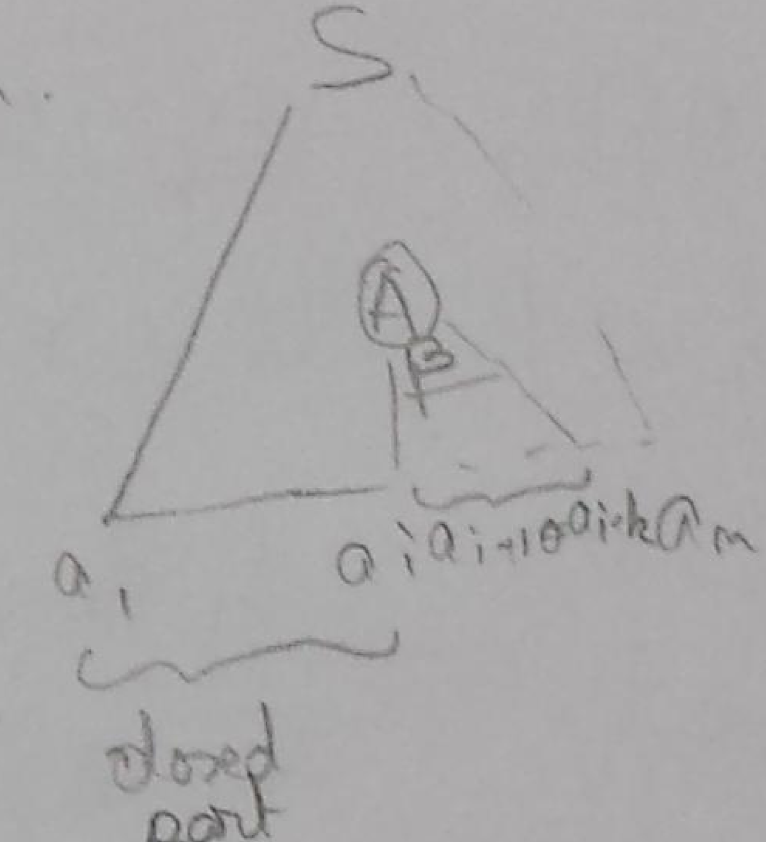
6) Success. $(q, n+1, \alpha, \varepsilon) \vdash (f, n+1, \alpha, \varepsilon)$

$(f, 6, \overline{S_1 a S_2 a S_3 c b S_3 c}, \varepsilon)$



LL(1) Parser.

$w = a_1 \ldots a_n$.



LL(k) left - ... read from left to right.
left - leftmost deriv.
$K \in \mathbb{N}$ - length of the prediction.

for α
first K symbols ($\in \Sigma$) that can be generated from α.

LL(K) Principle.

At any moment, the action is uniquely determined based on:

- closed part: $a_1 \ldots a_i$
- current symbol
- prediction of length K: $a_{i+1} \ldots a_{i+k}$

Def. A cfg is LL(k) if

1) $S \overset{*}{\underset{l}{\Rightarrow}} wA\alpha \overset{A \to \beta}{\underset{l}{\Rightarrow}} w\beta\alpha \overset{*}{\underset{l}{\Rightarrow}} wx$

2). $S \overset{*}{\underset{l}{\Rightarrow}} wA\alpha \overset{A \to \gamma}{\underset{l}{\Rightarrow}} w\gamma\alpha \overset{*}{\underset{l}{\Rightarrow}} wy$

3) $First_K(x) = First_K(y)$

$\Rightarrow \beta = \gamma$

T: A cfg is LL(K) iff $\forall A \in N, \forall A \to \alpha$ and $A \to \gamma$, $\alpha \neq \gamma$ then

$First_K(\alpha\beta) \cap First_K(\gamma\beta) = \emptyset$, $\forall \beta$ st $S \overset{*}{\underset{l}{\Rightarrow}} uA\beta$

$K = 1$ $First_1 = First$

$First : (N \cup \Sigma)^* \to \mathcal{P}(\Sigma)$

$First(\alpha) = \{ u \in \Sigma \mid \alpha \overset{*}{\Rightarrow} u \quad |u| \leq 1$

OR

$\alpha \overset{*}{\Rightarrow} ux \quad |u| = 1 \}$

$A \to \varepsilon \mid a \mid bA$

First $(A) = \{\underline{\varepsilon}, a, b\}$

FOLLOW : $(N \cup \Sigma)^* \to \mathcal{P}(\Sigma)$

FOLLOW $(\beta) = \{ u \in \Sigma \mid S \overset{*}{\underset{l}{\Rightarrow}} w \beta \alpha, u \in$ First (α)

LL(1) Parser : 1) Construct First 2) Construct Follow 3) Constr LL(1) Table
ℓ 4) config + moves

Prop:

1) $L_1 \oplus L_2$ $\oplus_k$ = concat of length K

$L_1 \oplus L_2 = \{ w \mid u \in L_1 \wedge v \in L_2 \quad uv = w$ or $uv = wx, |w| = 1\}$.

Ex1) $L_1 = \{a, ab\}$. L_1 concat $L_2 = \{ab, abb, ac, abc\}$

$L_2 = \{b, c\}$. $L_1 \oplus L_2 = \{a\}$

Ex2 $L_1 = \{\varepsilon, a\}$ L_1 concat $L_2 = \{b, c, ab, ac$

$L_2 = \{b, c\}$ $L_1 \oplus L_2 = \{b, c, a\}$.

2) First $(\alpha \beta)$ = First $(\alpha) \oplus$ First (β)

Alg First

input : G

output First(X), $\forall X \in (N \cup \Sigma)$

For $\forall a \in \Sigma$ do First $(a) = \{a\}$

For $\forall A \in N$ do

$F_0(A) = \{ a \in \Sigma \mid A \to a \in P$ or $A \to a\alpha \in P\}$

$i = 0$

Repeat

$i = i + 1;$

for $\forall A \in N$ do $F_i(A) = F_{i-1}(A)$

for $\forall A \to X_1 \ldots X_m$ do

if $F_{i-1}(X_j) \neq \emptyset$ $\forall j = \overline{1,m}$

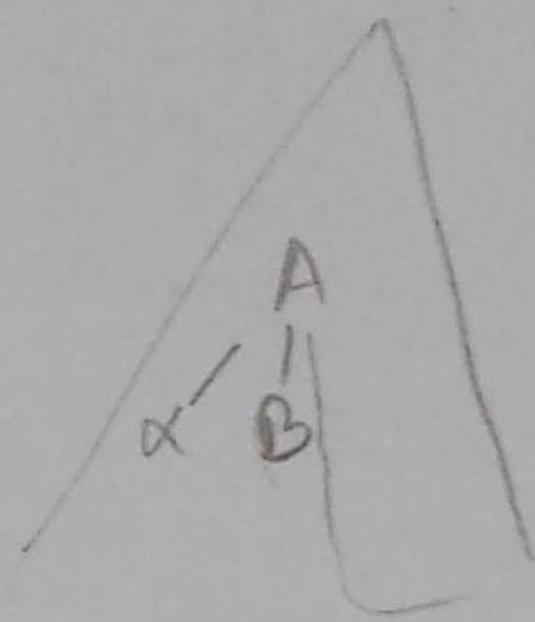
$F_i(A) = F_{i-1}(A) \cup \{ u \in \Sigma \mid u \in F_{i-1}(X_1) \oplus F_{i-1}(X_2) \ldots \oplus F_{i-1}(X_m)\}$

until $F_i(A) = F_{i-1}(A)$ $\forall A \in N$

First $(X) = F_i(X)$ $\forall X \in N \cup \Sigma$.

$FOLLOW(\beta) = \{ u \in \Sigma \mid S \overset{*}{\Rightarrow} w\beta\alpha,\ u \in First(\alpha) \}$.

Alg FOLLOW.



input, G, $First(X)$, $X \in N \cup \Sigma$.

output $FOLLOW(A)$, $A \in N$.

$L_0(S) = \{\varepsilon\}$

for $\forall A \in N \setminus \{S\}$ do $L_0(A) = \emptyset$

$i = 0$.

Repeat

  $i = i + 1$.

  for $\forall B \in N$ do.

    for $\forall A \to \alpha B \beta \in P$ do

      $L_i(B) = L_{i-1}(B) \cup First(\beta)$

      if $\varepsilon \in First(\beta)$ then

        $L_i(B) = L_i(B) \cup L_{i-1}(A)$

  until $L_i(A) = L_{i-1}(A)$, $\forall A \in N$

  for $\forall A \in N$ do $FOLLOW(A) = L_i(A)$

	Σ	$\$$ ← ε
N		
Σ		
$\$$		

"$" $\notin N \cup \Sigma$ $ - marking of stack

Rules for LL(1) Table:

1. $M(A, a) = (\alpha, i)$ if $A \to \alpha \in P$ (i)
   and
   $a \in First(\alpha)$, $a \neq \varepsilon$
2. $M(A, b) = (\alpha, i)$ if $A \to \alpha \in P$ (i)
   $\varepsilon \in First(\alpha)$, $b \in FOLLOW(A)$
3. $M(a, a) = pop$.
4. $M(\$, \$) = acc$
5. Otherwise error

Remarks

1) A cfg is LL(1) if LL(1) table does not contain conflicts → 2 values in the same cell

2) Table stores the information for parsing seq. → computed only once!

4) Config + moves.

LL(1) config (α, β, π)

- α: input stack (input seq)
- β: working stack (syntax tree)
- π: output stack (string of productions)

- initial config $(w\$, S\$, \varepsilon)$.
- final config $(\$, \$, \pi)$ (success)

Moves

1) Push.
$(ux\$, A\beta\$, \pi) \vdash (ux\$, \alpha\beta\$, \pi i)$.
if $M(A, u) = (\alpha, i)$

2) Pop. $(ux\$, u\beta\$, \pi) \vdash (x\$, \beta\$, \pi)$
if $M(u, u) = pop$

3) Accept $(\$, \$, \pi) \vdash acc$.

4) Otherwise error.
$M((a)x\$, X\beta\$, \pi) \vdash error$.
a → location of error.

list-decl → decl | decl ; list-decl   not LL(1)

{ list-decl → decl T.
{ T → ε | ; list decl.

if stmt → if C then S | if C then S else S. ← not LL(1)

{ if stmt → if C then S R.
{ R → ε | else S.

E → E + T | T
T → T * F | F
F → (E) | id | const

} ex LL(1) ! seminary.