Lecture 12

Wednesday February 12

Is the FIRST Set Sufficient?

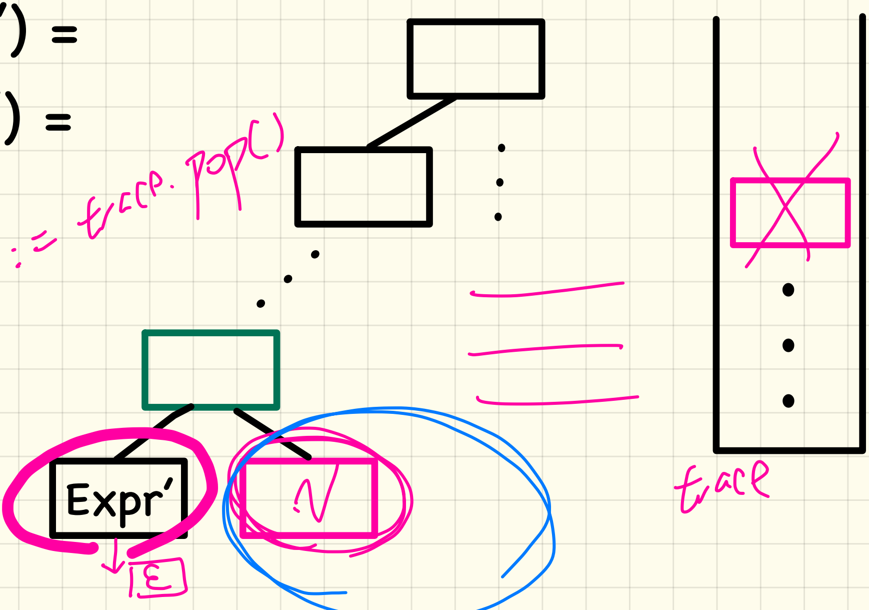
Expr′	$\rightarrow$	+	*Term*	*Term′*	(1)
	|	−	*Term*	*Term′*	(2)
	|	ϵ			(3)

FIRST(+ Term Term′) =

FIRST(- Term Term′) =

FIRST(epsilon) =

focus := trace.pop()



Expr

ε

word:

$$A \rightarrow \boxed{\beta_1}\,\beta_2\,\beta_3$$

$$\varepsilon$$

$$\mathrm{First}(A) = \mathrm{First}(\beta_1) \cup \mathrm{First}(\beta_2)$$

First

↓

rhs

rhs ⟶

Follow

↓

trailer

⟵ trail

FOLLOW Set

$$\textsc{Follow}(v) = \{w \mid w, x, y \in \Sigma^* \land v \overset{*}{\Rightarrow} x \land S \overset{*}{\Rightarrow} xwy\}$$

Right-Recursive CFG:

0	*Goal*	$\rightarrow$	*Expr*	6	*Term'*	$\rightarrow$	x *Factor Term'*
1	*Expr*	$\rightarrow$	*Term Expr'*	7		|	÷ *Factor Term'*
2	*Expr'*	$\rightarrow$	+ *Term Expr'*	8		|	ϵ
3		|	- *Term Expr'*	9	*Factor*	$\rightarrow$	(*Expr*)
4		|	ϵ	10		|	num
5	*Term*	$\rightarrow$	*Factor Term'*	11		|	name

	Expr	***Expr'***	***Term***	***Term'***	***Factor***
FIRST	(, name, num	+, -, ϵ	(, name, num	x, ÷, ϵ	(, name, num

	Expr	***Expr'***	***Term***	***Term'***	***Factor***
FOLLOW	eof,)	eof,)	eof, +, -,)	eof, +, -,)	eof, +, -, x, ÷,)

FOLLOW Set: Algorithm

$$\textsc{Follow}(v) = \{w \mid w, x, y \in \Sigma^* \wedge v \overset{*}{\Rightarrow} x \wedge S \overset{*}{\Rightarrow} xwy\}$$

ALGORITHM: *GetFollow*

  INPUT: *CFG* $G = (V, \Sigma, R, S)$

  OUTPUT: $\textsc{Follow} : V \longrightarrow \mathbb{P}(T \cup \{eof\})$

PROCEDURE:

  for $A \in V$: $\textsc{Follow}(A) := \varnothing$

  $\textsc{Follow}(S) := \{eof\}$

  $lastFollow := \varnothing$

  while ($lastFollow \neq \textsc{Follow}$):

    $lastFollow := \textsc{Follow}$

    for $\boxed{A \to \beta_1 \beta_2 \ldots \beta_k} \in R$:

      $\boxed{trailer} := \textsc{Follow}(A)$

      for $i: k \,..\, 1$: k down to 1

        if $\beta_i \in V$ **then**

          $\textsc{Follow}(\beta_i) := \textsc{Follow}(\beta_i) \cup trailer$

          if $\boxed{\epsilon \in \textsc{First}(\beta_i)}$ → β_i is nullable

            then $trailer := trailer \cup (\textsc{First}(\beta_i) - \epsilon)$

            else $trailer := \textsc{First}(\beta_i)$

        else

          $trailer := \textsc{First}(\beta_i)$

$$A \longrightarrow \beta_1\ \beta_2\ \cdots\ \beta_{k-1}\ \beta_k$$

<u>When</u> i = k

<u>When</u> i = k − 1

$A \rightarrow \beta_1 \beta_2 \beta_3 \beta_4$

ϵ

Assume: $\boxed{\text{First}}$ is computed.

Calculate Follow of β_1, β_2, β_3, and β_4

$$\text{Follow}(\beta_4) = \text{Follow}(A)$$

$\boxed{\text{Term}} \rightarrow \text{Factor } \text{Term}'$

$F(\text{Term}') = F(\text{Term})$

$$\text{Follow}(\beta_3) = \begin{cases} \text{First}(\beta_4) & \text{if } \beta_4 \text{ is not nullable.} \\ \text{First}(\beta_4) \cup \text{Follow}(\beta_4) & \text{if } \beta_4 \text{ is nullable} \end{cases}$$

Follow(A)

$$\boxed{A} \to \beta_1 \mid \beta_2 \mid \beta_3 \mid \beta_4 \mid \quad \text{trailer}$$

nullable : β_3

<u>not</u> nullable: $\beta_1, \beta_2, \beta_4$

trailer

$First(\beta_4) \cup First(\beta_3)$

$First(\beta_4)$

$Follow(\beta_1) = First(\beta_2) \cup \cancel{Follow(\beta_2)}$

$\because \beta_4$ not nullable

$Follow(\beta_4) = Follow(A)$

$Follow(\beta_3) = First(\beta_4) \cup \cancel{Follow(\beta_4)}$

$Follow(\beta_2) = First(\beta_3) \cup \underline{Follow(\beta_3)}$ $\leftarrow \because \beta_3$ nullable.

Right-Recursive CFG:

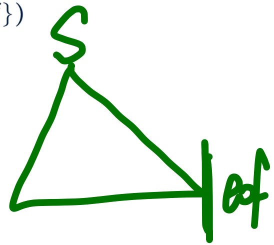
0	*Goal*	→	*Expr*	6	*Term′*	→	× *Factor Term′*
1	*Expr*	→	*Term Expr′*	7		|	÷ *Factor Term′*
2	*Expr′*	→	+ *Term Expr′*	8		|	ϵ
3		|	- *Term Expr′*	9	*Factor*	→	(*Expr*)
4		|	ϵ	10		|	num
5	*Term*	→	*Factor Term′*	11		|	name

~~G~~, ~~E~~, ~~E′~~, ~~T~~, T′

FOLLOW Set: Tracing

First choose rules whose LHS is processed.
Then rules whose RHS ends with a terminal.

ALGORITHM: *GetFollow*
INPUT: *CFG* $G = (V, \Sigma, R, S)$
OUTPUT: $\text{FOLLOW} : V \longrightarrow \mathbb{P}(T \cup \{eof\})$
PROCEDURE:
for $A \in V$: $\text{FOLLOW}(A) := \varnothing$
$\text{FOLLOW}(S) := \{eof\}$
$lastFollow := \varnothing$
while ($lastFollow \neq \text{FOLLOW}$):
  $lastFollow := \text{FOLLOW}$
  for $A \to \beta_1 \beta_2 \dots \beta_k \in R$:
    trailer $:= \text{FOLLOW}(A)$
    for $i : k \dots 1$:
      if $\beta_i \in V$ then
        $\text{FOLLOW}(\beta_i) := \text{FOLLOW}(\beta_i) \cup$ *trailer*
        if $\epsilon \in \text{FIRST}(\beta_i)$
          then *trailer* $:=$ *trailer* $\cup (\text{FIRST}(\beta_i) - \epsilon)$
          else *trailer* $:= \text{FIRST}(\beta_i)$
      else
        trailer $:= \text{FIRST}(\beta_i)$



	Expr	Expr′	Term	Term′	Factor
FIRST	(, name, num	+, -, ϵ	(, name, num	x, ÷, ϵ	(, name, num

Goal: {eof}

Expr	Expr′	Term	Term′	Factor
eof	eof	+		*
)	)	-		÷
		ϵ		ϵ
		eof		
		)		

$Term' \to * \boxed{Factor}\ Term'$

Follow(T') = ϕ

Follow(Factor) = First(Term')

= * , ε

trailer ::= trailer ∪ First(Expr')

t = Follow(E')

Expr → Term Expr'

Follow(Expr) = {eof,)}

Follow(Expr') = {eof,)}

Follow(Term) = First(Expr') ∪ Follow(Expr')

First(Expr') = {+, -, ε}

¬(ε ∈ First(Expr')

Follow(β_3) = Follow(Factor)

β_1 β_2 β_3 ε

Factor → (Expr)

Follow(β_2) = ① First(β_3) ≠ if β_3 not nullable,

② First(β_3) ∪ Follow(β_3) ≠ if β_3 nullable

trailer

Follow(Expr) = First( ) = {) }

Follow(Expr) = {) }

Follow(Expr) = (First( ) ∪ ~~Follow(Factor)~~)

{) } " ( ) " if not nullable

$\underline{Factor}_{5} \rightarrow \boxed{(\boxed{Expr})}$

assume:
Follow(Factor)
already computed
previously.

Compute now:
Follow(Expr)

First(___) → terminal / non-terminal

$First^t$( )

$\cancel{F}$(___)

First(___) → RHS of some production

Follow(___) → variable

Backtrack-free Grammar

$\{+, -\}$ $\quad$ $\{*, +\}$

$$A \to \gamma_1 \mid \gamma_2 \mid \gamma_3 \mid \cdots \gamma_n$$

Con. $\quad$ Con. $\quad$ Con.

Backtrack-Free Grammar

$$\textsc{First}^+(A \rightarrow \beta) = \begin{cases} \textsc{First}(\beta) & \text{if } \epsilon \notin \textsc{First}(\beta) \\ \textsc{First}(\beta) \cup \textsc{Follow}(A) & \text{otherwise} \end{cases}$$

$\textsc{First}(\beta)$ is the extended version where β may be $\beta_1 \beta_2 \dots \beta_n$

$A \rightarrow \gamma_1 \mid \gamma_2 \mid \dots \mid \gamma_n$ satisfying:

$$\forall i, j : 1 \leq i, j \leq n \wedge i \neq j \bullet \textsc{First}^+(\gamma_i) \cap \textsc{First}^+(\gamma_j) = \varnothing$$