Predictive Parser Transforms

Left Recursion Elimination

Replace $A \longrightarrow A\ \alpha \mid \beta$

With $A \longrightarrow \beta\ A'$

$A' \longrightarrow \alpha\ A' \mid \varepsilon$

Where β does not start with *A* and may not be present

Left Factoring

Replace $A \longrightarrow \alpha\ \beta_1 \mid \dots \mid \alpha\ \beta_m \mid y_1 \mid \dots \mid y_n$

With $A \longrightarrow \alpha\ A' \mid y_1 \mid \dots \mid y_n$

$A' \longrightarrow \beta_1 \mid \dots \mid \beta_m$

Where β_i and y_i are sequence of symbols with no common prefix

y_i May not be present, one of the β may be ε

Building the Parse Table

FIRST(α) for $\alpha = Y_1 Y_2 \ldots Y_k$
Add FIRST(Y_1) - {ε}
If ε is in FIRST($Y_{1 \text{ to } i-1}$): add FIRST(Y_i) – {ε}
If ε is in all RHS symbols, add ε

FOLLOW(A) for $X \longrightarrow \alpha\ A\ \beta$
If A is the start, add **eof**
Add FIRST(β) – {ε}
Add FOLLOW(*X*) if ε in FIRST(β) or β empty

Table[X][t]

```
for each production X ⟶ α
 for each terminal t in FIRST(α)
   put α in Table[X][t]
 if ε is in FIRST(α){
   for each terminal t in FOLLOW(X){
     put α in Table[X][t]
```

Left Recursion Elimination Examples

$A \longrightarrow A\ \alpha \mid \beta$ ⇒ $A \longrightarrow \beta\ A'$
$A' \longrightarrow \alpha\ A' \mid \varepsilon$

$E \longrightarrow E$ **cross id** | **id**

$E \longrightarrow E + T \mid T$
$T \longrightarrow \mathrm{T} * \mathrm{F} \mid \mathrm{F}$
$F \longrightarrow (\ \mathrm{E}\ ) \mid \mathrm{id}$

SList ⟶ *SList D* | ε

D ⟶ *Type* **id semi**

Type ⟶ **bool** | **int**

Left Factoring Examples

$A \longrightarrow \alpha\, \beta_1 \mid \ldots \mid \alpha\, \beta_m \mid y_1 \mid \ldots \mid y_n \quad \Rightarrow \quad \begin{array}{l} A \longrightarrow \alpha\, A' \mid y_1 \mid \ldots \mid y_n \\ A' \longrightarrow \beta_1 \mid \ldots \mid \beta_m \end{array}$

X ⟶ **< a > | < b > | < c > | d**

Stmt ⟶ **id assign** *E* | **id (** *EList* **)** | **return**
E ⟶ **intlit** | **id**
Elist ⟶ *E* | *E* **comma** *EList*

S ⟶ **if** *E* **then** *S* | **if** E **then** *S* **else** *S* | **semi**
E ⟶ **boollit**

FIRST(α) for $\alpha = Y_1\ Y_2 \ldots Y_k$

- Add FIRST(Y_1) - {ε}
- If ε is in FIRST($Y_{1 \text{ to } i-1}$): add FIRST(Y_i) – {ε}
- If ε is in all RHS symbols, add ε

Exp	⟶	*Term Exp'*
Exp'	⟶	**minus** *Term Exp'* | ε
Term	⟶	*Factor Term'*
Term'	⟶	**divide** *Factor Term'* | ε
Factor	⟶	**intlit** | **lparens** *Exp* **rparens**

E → TX
X → +TX | ε
T → FY
Y → *FY | ε
F → (E) | id

FOLLOW(A) for $X \longrightarrow \alpha\ A\ \beta$

If A is the start, add **eof**

Add FIRST(β) – {ε}

Add FOLLOW(X) if ε in FIRST(β) or β is empty

S	⟶	B **c** **|** D B
B	⟶	**a b** **|** **c** S
D	⟶	**d** **|** ε

Parse Table

```
for each production X ⟶ α
 for each terminal t in FIRST(α)
   put α in Table[X][t]
 if ε is in FIRST(α){
   for each terminal t in FOLLOW(X){
     put α in Table[X][t]
```

S ⟶ B **c** | D B

B ⟶ **a b** | **c** S

D ⟶ **d** | ε

	a	**b**	**c**	**d**	**eof**
S					
B					
D					

FIRST (S)

FIRST (B)

FIRST (D)

FIRST (B **c**)

FIRST (D B)

FIRST (**a b**)

FIRST (**c** *S*)

FOLLOW (S)

FOLLOW (B)

FOLLOW (D)

Parse Table

```
for each production X ⟶ α
 for each terminal t in FIRST(α)
   put α in Table[X][t]
 if ε is in FIRST(α){
   for each terminal t in FOLLOW(X){
     put α in Table[X][t]
```

$S \rightarrow$ **(** S **)** | **{** S **}** | **ε**

	(	**)**	**{**	**}**	**eof**
S					

FIRST (S)

FIRST (**(** S **)**)

FIRST (**{** S **}**)

FIRST (ε)

FOLLOW (S)

$S \rightarrow$ **+** S | **ε**

	+	**eof**
S		

FIRST (S)

FIRST (**+** S)

FIRST (ε)

FOLLOW (S)