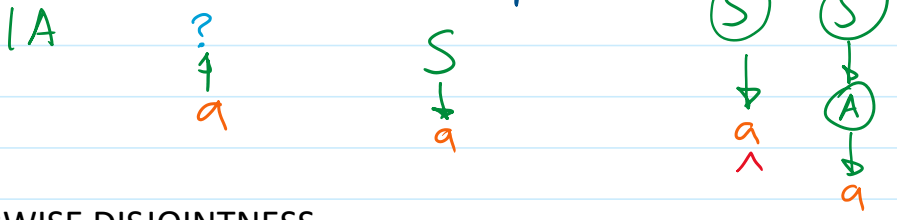
11 Grammar Issues with Top-Down parsers

Monday, October 14, 2024 12:26 PM

- AMBIGUITY

Not an issue for top-down parsers

e.g. $S \to a \mid A$, $A \to a$



- LACK OF PAIRWISE DISJOINTNESS

for a non-terminal A
more than one body of A begins with the same prefix

Eg.

$A \to aD \mid aC \mid aaB \mid b$

```
FUNCTION parse_A()
  IF token = 'a' THEN
    getToken()
```

The token does not gives enough information about which body to apply

Solution 1:- read more tokens.

Solution 2:- hack the parser, Choose bodies in order.

Solution 3:- Change the grammar.

Remedy: Left-Factoring

Replace: $A \to \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3 \mid \ldots \mid \Gamma_1 \mid \Gamma_2 \mid \ldots$

α is a common prefix

with: $A \to \alpha A' \mid \Gamma_1 \mid \Gamma_2 \mid \ldots$

$A' \to \beta_1 \mid \beta_2 \mid \beta_3 \mid \ldots$

E.g.

E.g.

$A \to aD \mid aC \mid aaB \mid b$

($\alpha\beta_1$, $\alpha\beta_2$, $\alpha\beta_3$, Γ_1)

Left factor.

$A \to aA' \mid b$

$A' \to D \mid C \mid aB$

Exercise:

$S \to zzAy \mid zzyz \mid zzy \mid zzx \mid yyA \mid yyy$

$A \to xxAy \mid xzyA \mid yzx$

Rewrite: S

$S \to zzS' \mid yyA \mid yyy$

$S' \to yAy \mid yz \mid y \mid x$

Rewrite A

$A \to xA' \mid yzx$

$A' \to xAy \mid zyA$

Rewrite S

$S \to zzS' \mid yyS''$

$S'' \to A \mid y$

Rewrite S'

$S' \to yS''' \mid x$

$S''' \to Ay \mid z \mid \lambda$

- LEFT RECURSION

• Direct

$A \to Aa \mid Ab \mid c$

$\{c, ca, caaaaa, cababb, \ldots\}$

```
FUNCTION parse_A()
  parse_A()
  ...
  ...
```

$A \to c \mid Aa \mid Ab$

```
FUNCTION parse_A()
```

• Indirect

$A \to Bx$

$B \to Cy$

$C \to Az \mid w$

parse_A()
↳ parse_B()
↳ parse_C()
↳ parse_A()
↳ parse_B()
...

$A \to c \mid Aa \mid Ab$

```
FUNCTION parse_A()
  IF token = 'c' THEN
    getToken()
  ELSE
    parse_A()
  …
  …
```

Remedy: Direct Left Recursion Elimination

Replace: $A \to A\alpha_1 \mid A\alpha_2 \mid A\alpha_3 \mid \ldots \mid \beta_1 \mid \beta_2 \mid \beta_3 \mid \ldots$

by: $A \to \beta_1 A' \mid \beta_2 A' \mid \beta_3 A' \mid \ldots$

$A' \to \alpha_1 A' \mid \alpha_2 A' \mid \alpha_3 A' \mid \ldots \mid \lambda$

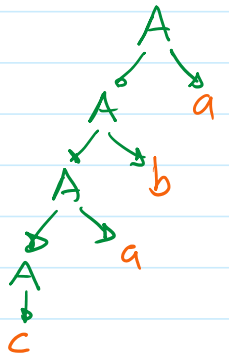
e.g.1

$A \to Aa \mid Ab \mid c \Rightarrow A \to cA'$

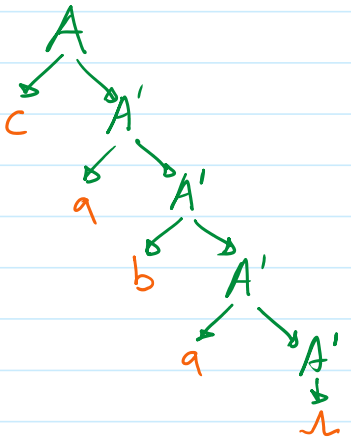
$A' \to aA' \mid bA' \mid \lambda$

($\alpha_1 = a$, $\alpha_2 = b$, $\beta_1 = c$)

Intuition.



caba



E.G #2

$E \to E + T \mid T$ ($\alpha_1 = +T$, $\beta = T$)

$T \to T * F \mid F$

$F \to int \mid (E)$

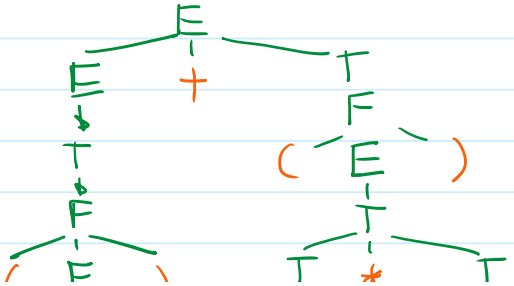
Reunte.

$E \to TE'$

$E' \to +TE' \mid \lambda$

$T \to FT'$

$(5+3) + (8 * (1+2))$

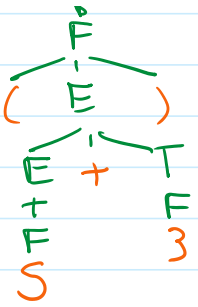


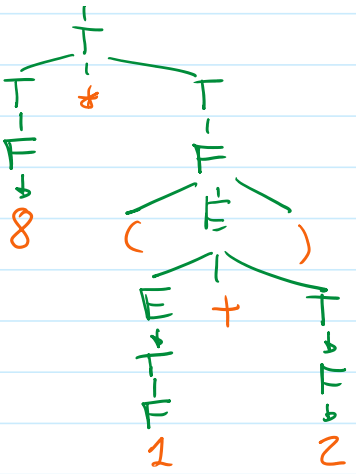
$E' \rightarrow + T E' \mid \lambda$

$T \rightarrow F T'$

$T' \rightarrow * F T' \mid \lambda$

$F \rightarrow \underline{int} \mid (E)$





T'

* F T'

* F * F T'

* F * F * F T'

* F * F * F * F T'

0 or more repetitions of * F

In EBNF

$E \rightarrow T \{ + T \}$

$T \rightarrow F \{ * F \}$

$F \rightarrow \underline{int} \mid (E)$

```
FUNCTION parse_E()
  parse_T()
  WHILE token = '+' DO
    getToken()
    parse_T()
  END
END.
```

```
FUNCTION parse_T()
  parse_F()
  WHILE token = '*' DO
    getToken()
    parse_F()
  END
END.
```

```
FUNCTION parse_F()
  IF is_integer(token) THEN
    getToken()
  ELSIF token = '(' THEN
    getToken()
    parse_E()
    IF token = ')' THEN
      getToken()
    ELSE
      error(") expected")
    END
  ELSE
    error()
  END
END.
```