

11 Grammar Issues with Top-Down Parsers

Monday, March 17, 2025

12:38 PM

- AMBIGUITY

Not an issue for top-down parsers

eg.

$S \rightarrow a \mid A$
 $A \rightarrow a$



- LACK OF PAIRWISE DISJOINTNESS

- for a non-terminal A

more than one body of A begins with the same prefix.

E.g.

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

```
FUNCTION Parse_A ( )  
  IF token = 'a' THEN  
    getToken()  
    ?????
```

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

Solution 1:- Implement Backtracking.

- computationally expensive

Solution 2:- read more tokens.

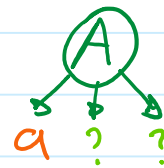
- complicates parser

Solution 3:- hack the parser.

Arbitrarily choose a body.

- makes parser incomplete.

Solution 4:- Change the grammar



- Remedy: Left-Factoring

Replace $A \rightarrow \alpha \beta_1 \mid \alpha \beta_2 \mid \alpha \beta_3 \mid \dots \mid \Gamma_1 \mid \Gamma_2 \mid \dots$
 α is a common prefix

With: $A \rightarrow \alpha A' \mid \Gamma_1 \mid \Gamma_2 \mid \Gamma_3 \mid \dots$
 $A' \rightarrow \beta_1 \mid \beta_2 \mid \beta_3 \mid \dots$

E.g.

$A \rightarrow \underbrace{a}_{\alpha} \underbrace{D}_{\beta_1} \mid \underbrace{a}_{\alpha} \underbrace{C}_{\beta_2} \mid \underbrace{aa}_{\alpha} \underbrace{B}_{\beta_3} \mid \underbrace{b}_{\Gamma_1}$

Left-factor:

$A \rightarrow a A' \mid b$

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

• ;

• ;

Exercise:

$S \rightarrow \underline{zz} y A_y \mid \underline{zz} y z \mid \underline{zz} y \mid \underline{zz} x \mid y y A \mid y y y$

$A \rightarrow \underline{x} x A_y \mid \underline{x} z y A \mid y z x$

rewrite:

$S \rightarrow \underline{zz} S' \mid \underline{y y} A \mid \underline{y y y}$

$S' \rightarrow \underline{x} A_y \mid \underline{y z} \mid \underline{y} \mid x$

rewrite

$A \rightarrow x A' \mid y z x$

$A' \rightarrow x A_y \mid z y A$

$S' \rightarrow y S'' \mid x$

$S'' \rightarrow A_y \mid z \mid \Lambda$

$S \rightarrow y y S''' \mid z z S'$

$$S \rightarrow yyS''' \mid zzS'$$

$$S''' \rightarrow A \mid y$$

• LEFT RECURSION

• Direct

$$A \rightarrow Aa \mid Ab \mid c$$

$$\{c, cb, cab, caabb, \dots\}$$

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

• Indirect.

$$A \rightarrow Bx$$

$$B \rightarrow Cy$$

$$C \rightarrow Az \mid w$$

```
Parse-A()
  ↳ Parse-B()
    ↳ Parse-C()
      ↳ Parse-A()
        ↳ Parse-B()
          ↳ Parse-C()
            ...
```

• Remedy: Direct Left Recursion Elimination

Replace:

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid A\alpha_3 \mid \dots \mid \beta_1 \mid \beta_2 \mid \beta_3 \mid \dots$$

by

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \beta_3 A' \mid \dots$$

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \alpha_3 A' \mid \dots \mid \epsilon$$

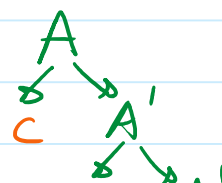
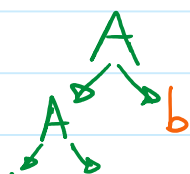
e.g.1

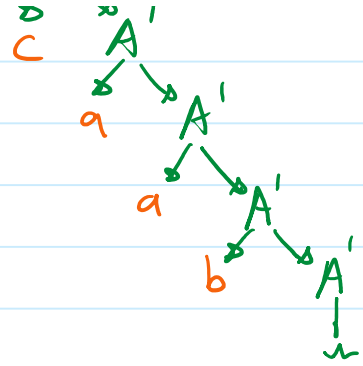
$$A \rightarrow Aa \mid Ab \mid c \quad \xRightarrow{\text{rewrite}} \quad A \rightarrow cA'$$

$$A' \rightarrow aA' \mid bA' \mid \epsilon$$

Intuition.

caab





$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

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

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid \wedge \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid \wedge \\ F &\rightarrow \underline{\text{int}} \mid (E) \end{aligned}$$
[illegible]

T' ?

* FT'

* F FT'

* F F FT'

* F F F FT'

In EBNF

$$E \rightarrow T \{ + T \}$$
$$T \rightarrow F \{ * F \}$$
$$F \rightarrow \underline{\text{int}} \mid (E)$$

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

Ex:2

Ex:1

$$\begin{aligned}
 S &\rightarrow aS \mid \underline{Sa} \mid \underline{Sb} \mid x \\
 A &\rightarrow \underline{Ab} \mid \underline{Ac} \mid a \mid \lambda \\
 A &\rightarrow aA' \mid A' \\
 A' &\rightarrow bA' \mid cA' \mid \lambda \\
 S &\rightarrow aSS' \mid xS' \\
 S' &\rightarrow aS' \mid bS' \mid \lambda
 \end{aligned}$$

— 0 — EOF