

lexical analyzer \rightarrow parser.

Top-down.

(Start symbol \rightarrow root)

(left to right scan.

Preorder: \textcircled{A}

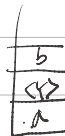
Left most derivation.

Nonterminal: $\boxed{A \rightarrow B}$ 3 분.

$\langle A \rangle \rightarrow a \langle X \rangle b.$

Bottom up:

$\langle X \rangle \rightarrow a \langle X \rangle b$ handle



reduce.

Start symbol 은 stack에 들어갈 수 없.



1. Goto (Nonterminal of Look): reduce가 될까?

2. input에 token이 들어갈까?

3. Shift: stack에 들어갈까?

4. reduce: rule 4번을 R4. Handle가 생김.

<primitive function>

Top

(MINUS $t_1 t_2$) : $t_1 - t_2$.

otherwise True
false, 0 \rightarrow false.

(IF $t_1 t_2$) : t_2 , if t_1 is true.

otherwise.

prefix

what affect.



{MINUS, IF}

primitive function

$FUN: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$

Domain이 정수만 가능

ADD, MULTI, ...

Term

$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$
3. → $(M(IF, 1, 2), 3)$, $(M, 1, 2)$

Interpreter:

→ source 코드를
정리하는 것.

문법: 코드를 해석하는 것.

VALUE: TOK → Z

ex) % 3 \downarrow
Z는 3.

ex) sum \downarrow
Z는 undefined.

ex) % -2 \downarrow
Z는 -2.

% (MINUS (-2))

ex) % (M (IF (2) 3))
 $\Rightarrow -1$.

ex) % (M (M 0 2))
 $\Rightarrow 3$.

TOK → BUF.

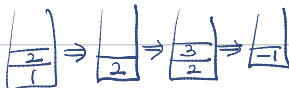
$\langle \text{term} \rangle \rightarrow \text{var} \mid \text{int} \mid (\text{MINUS } \langle \text{term} \rangle \langle \text{term} \rangle) \mid (\text{IF } \langle \text{term} \rangle \langle \text{term} \rangle)$

VALUE() = 현재 해석하는 것.

postfix

① postfix 해석. $(M(IF\ 1\ 2)\ 3) \Rightarrow 1\ 2\ IF\ 3\ M$.

②. stack을 이용한 것.



① Syntax Analyzer: 문법 검사.

② prefix \Rightarrow postfix로 바꿈.

③. 실행

이런 것은 prefix로 작성함.

expression of the algebra = For Language of the term of expression

$$X = (f), \quad X = (g), \quad X = (f - z)$$