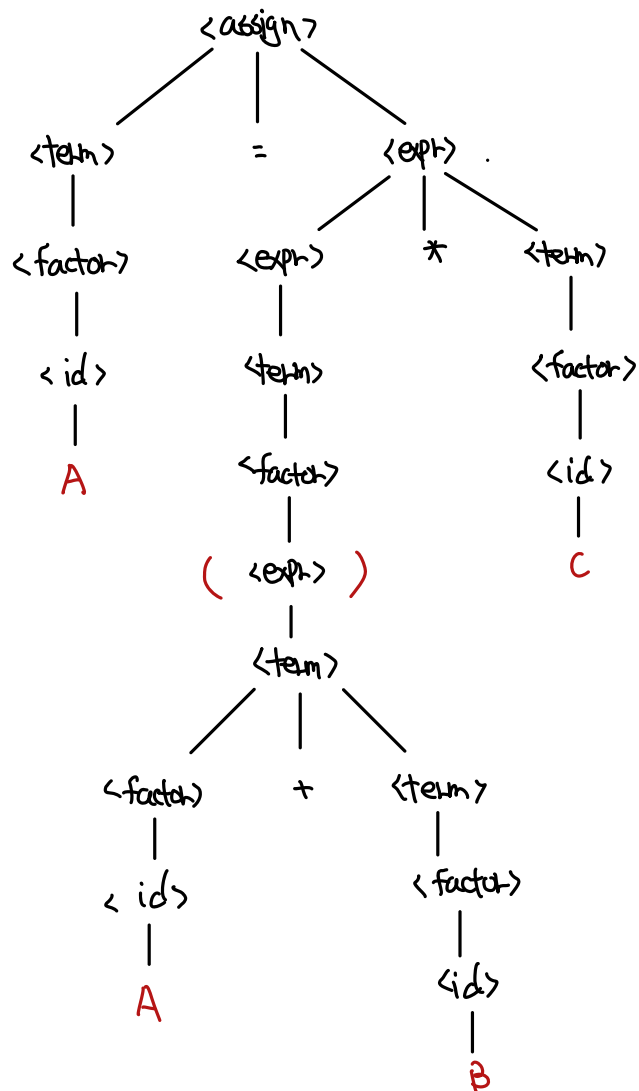


1. $A = (A + B) * C$



-
- 2.
- $\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 - $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle * \langle \text{term} \rangle \mid \langle \text{term} \rangle$
 - $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle + \langle \text{term} \rangle \mid \langle \text{factor} \rangle$
 - $\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) \mid \langle \text{id} \rangle \mid (+ \langle \text{id} \rangle) \mid (- \langle \text{id} \rangle)$
 - $\langle \text{id} \rangle \rightarrow A \mid B \mid C$

3.

Input : 0 OR 1 AND 1 == True

$\langle \text{compare} \rangle \rightarrow \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool expr} \rangle \text{ AND } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool expr} \rangle \text{ OR } \langle \text{bool expr} \rangle \text{ AND } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool} \rangle \text{ OR } \langle \text{bool expr} \rangle \text{ AND } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool value} \rangle \text{ OR } \langle \text{bool expr} \rangle \text{ AND } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool expr} \rangle$ AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool} \rangle$ AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool value} \rangle$ AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND $\langle \text{bool} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND $\langle \text{bool value} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND 1 == $\langle \text{bool} \rangle$

0 OR 1 AND 1 == $\langle \text{bool value} \rangle$

0 OR 1 AND 1 == True

$\langle \text{compare} \rangle \rightarrow \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool expr} \rangle \text{ OR } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool} \rangle \text{ OR } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

$\langle \text{bool value} \rangle \text{ OR } \langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool expr} \rangle$ AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool} \rangle$ AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR $\langle \text{bool value} \rangle$ AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND $\langle \text{bool expr} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND $\langle \text{bool} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND $\langle \text{bool value} \rangle == \langle \text{bool expr} \rangle$

0 OR 1 AND 1 == $\langle \text{bool} \rangle$

0 OR 1 AND 1 == $\langle \text{bool value} \rangle$

0 OR 1 AND 1 == True

4. a) A Java class definition header statement :

Java example : Public class main {int x=5}
 $\langle \text{class declaration} \rangle \rightarrow \langle \text{class modifiers} \rangle \text{ class } \langle \text{id} \rangle \{ \text{class body} \}$

$\langle \text{class declaration} \rangle \rightarrow \langle \text{class modifiers} \rangle \text{ class } \langle \text{identifier} \rangle \langle \text{class body} \rangle$

$\langle \text{class modifiers} \rangle \rightarrow \langle \text{class modifier} \rangle | \langle \text{class modifiers} \rangle \langle \text{class modifier} \rangle$

$\langle \text{class modifier} \rangle \rightarrow \text{public} | \text{abstract} | \text{final}$

$\langle \text{identifier} \rangle \rightarrow \langle \text{Letter} \rangle | \langle \text{digit} \rangle$

$\langle \text{Letter} \rangle \rightarrow a \dots | z | A \dots | Z$

$\langle \text{digit} \rangle \rightarrow 0 \dots | 9$

$\langle \text{class body} \rangle \rightarrow \langle \text{Type} \rangle \langle \text{assign} \rangle$

$\langle \text{Type} \rangle \rightarrow \text{int} | \text{string} | \text{char} \dots$

$\langle \text{assign} \rangle \rightarrow \langle \text{identifier} \rangle = \langle \text{expr} \rangle$

$\langle \text{expr} \rangle \rightarrow \langle \text{identifier} \rangle$

b) A C switch statement

C sample : `switch (grade) {case (grade >= 90): printf ('A'); break;}`
`<selection-statement> → switch (<expr>) {<labeled-statement>}`

`<selection-statement> → switch (<expr>){<labeled-statement>}`

`<expr> → <type> <id>`

`<type> → <char> | <string> | <int> ...`

`<id> → <letter> | <digit> .`

`<letter> → a... | z | A... | Z .`

`<digit> → 0... | 9`

`<labeled-statement> → case <constant-expr> : <statement> .`

`<constant-expr> → <relational-expr>`

`<relational-expr> → <relational-expr> >= <digit>`

`<statement> → <expr> | <jump-statement>`

`<jump-statement> → break ;`

c) A C union definition

union definition : `union union_name { datatype field_name; datatype field_name; } ;`

`<struct-or-union-specifier> → <struct-or-union> <letter> {<expr>}`

`<struct-or-union-specifier> → <struct-or-union> <id> {<expr>}`

`<struct-or-union> → struct | union .`

`<id> → <letter> | <digit> .`

`<letter> → a... | z | A... | Z .`

`<digit> → 0... | 9`

`<expr> → <type> <letter>`