CS 4600 - HW2

Ekene Onyekachi

1) a) <program> ::= <stmt> {stmt}

<stmt> ::= <assign> | if-stmt> | <while-stmt>

<assign> ::= <id> = <expr>;

<if-stmt> ::= if (<expr>) <stmt> [else < stmt >]

<while-stmt> ::= while (<expr>) <stmt>

<expr> ::= <term> {(+|-) <term>}

<term> ::= <factor> {(*|/) <factor>}

<factor> ::= <id> | <number> | (<expr>)

b) <function-call> ::= <id (<opt-arg-list>)

<opt-arg-list> ::= <arg-list> | ε

<arg-list> ::= <expr> <arg-list-tail>

<arg-list-tail> ::= <expr> <arg-list-tail> | ε

<declaration> ::= <type> <id> | <type> <id> = <expr>

c) <or-expr> ::= <or_ expr> or <and-expr>

<and-expr> ::= <and-expr> and <not-expr> | <not-expr>

<not-expr> ::= not <not-expr> | <primary>

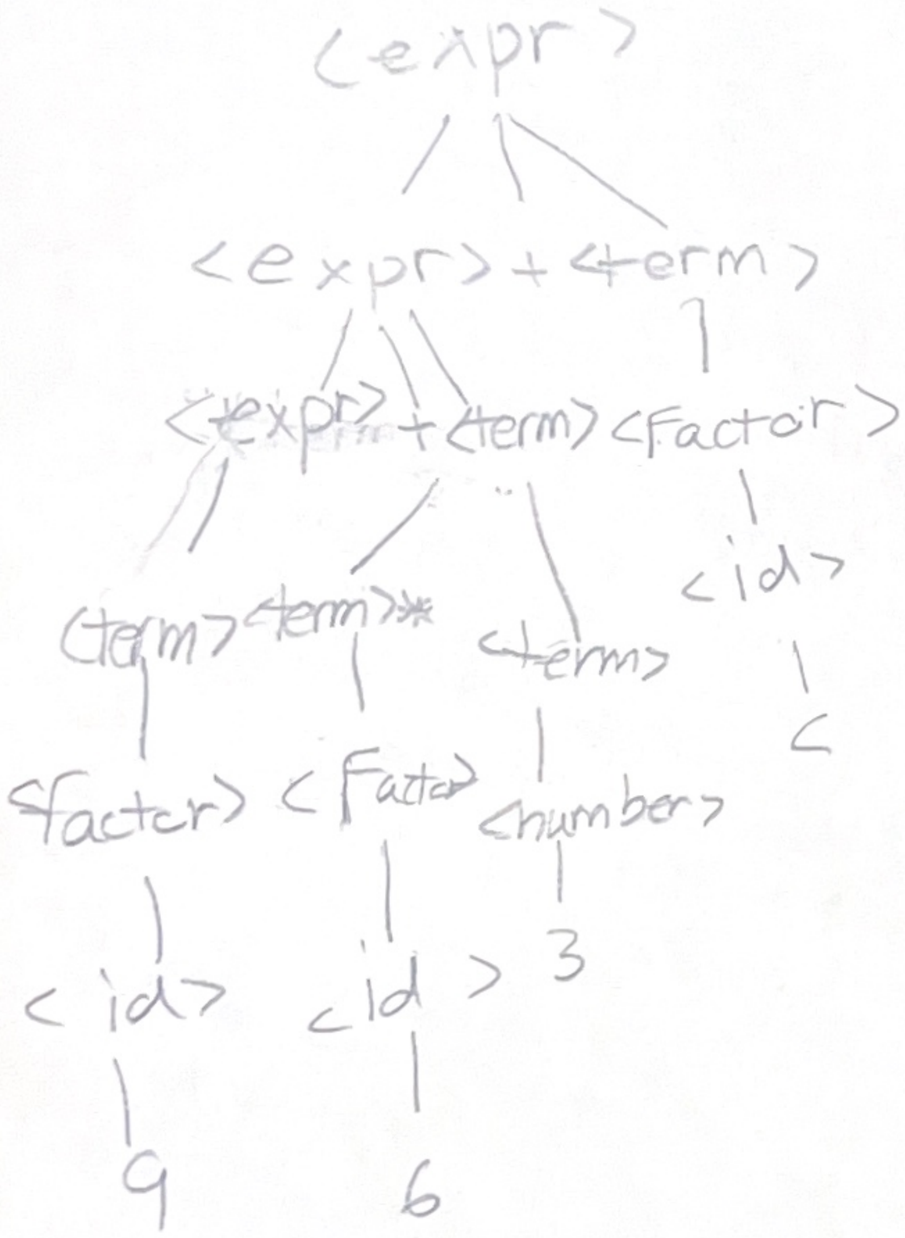
<primary> ::= true | false | (<or-expr>)

2a) <S>

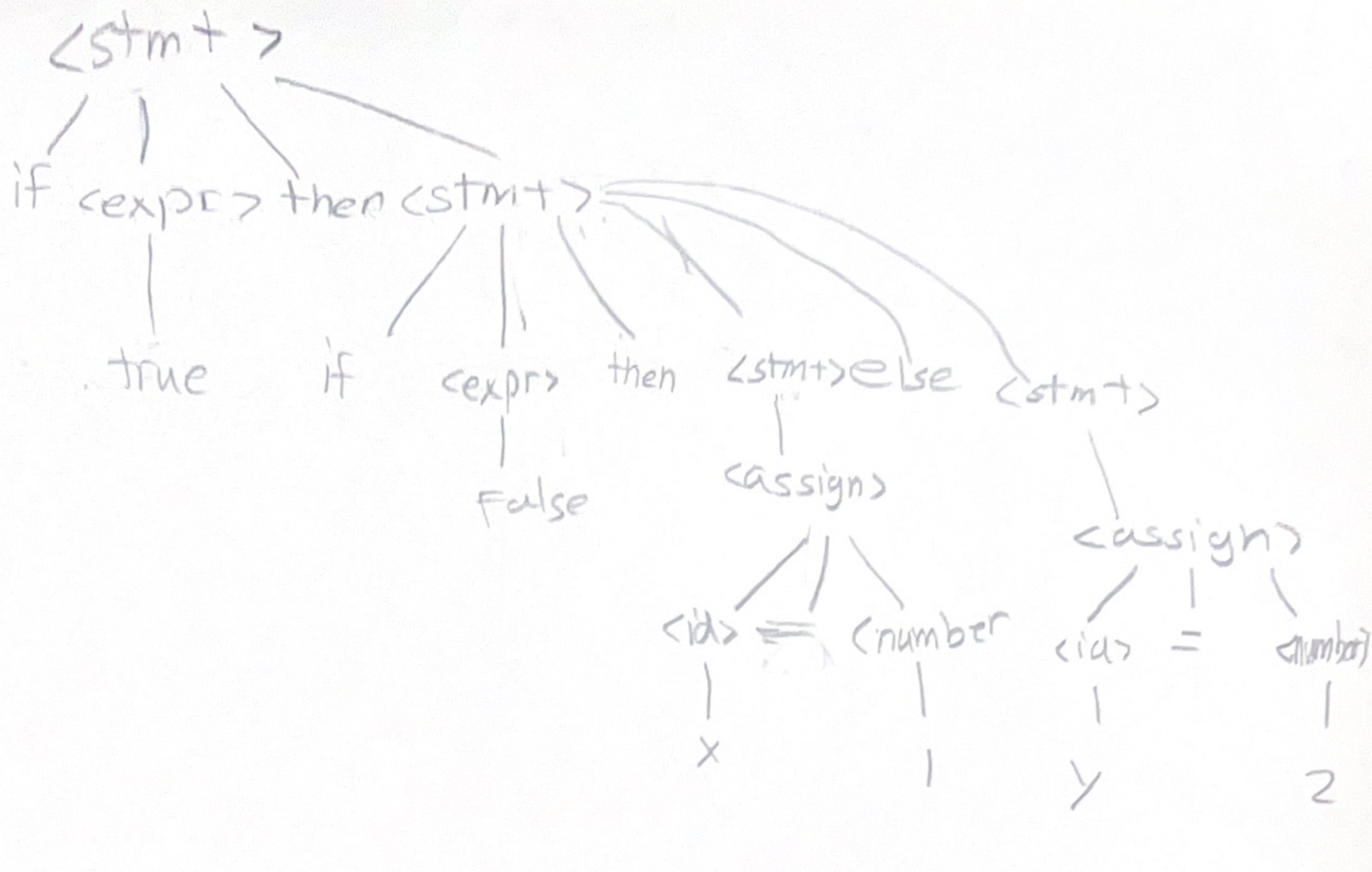
<S> ⇒ <A><B><C>
⇒ a <A><B><E>
⇒ a a <B><C>
⇒ a a b <B><C>
⇒ a a b b <B><C>
⇒ a a b b b <C>
⇒ a a b b b c <C>
⇒ a a b b b c c

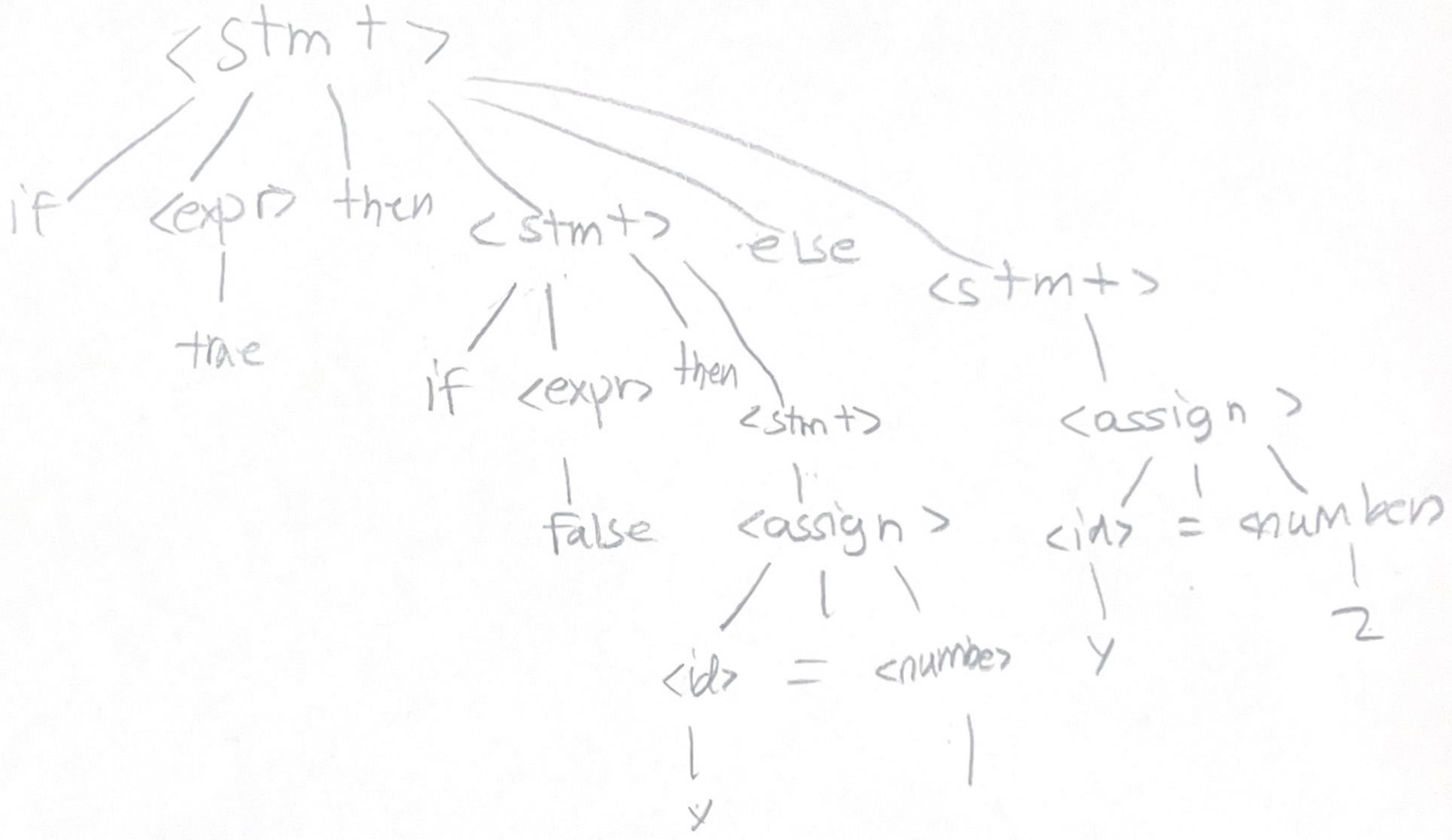
final string: a a b b b c c

b) a + b * 3 + c

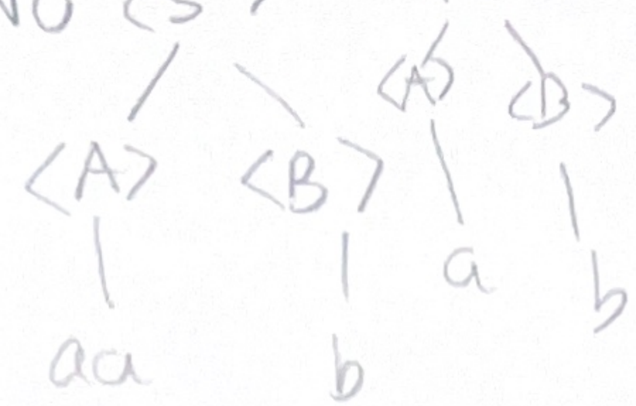


2c)

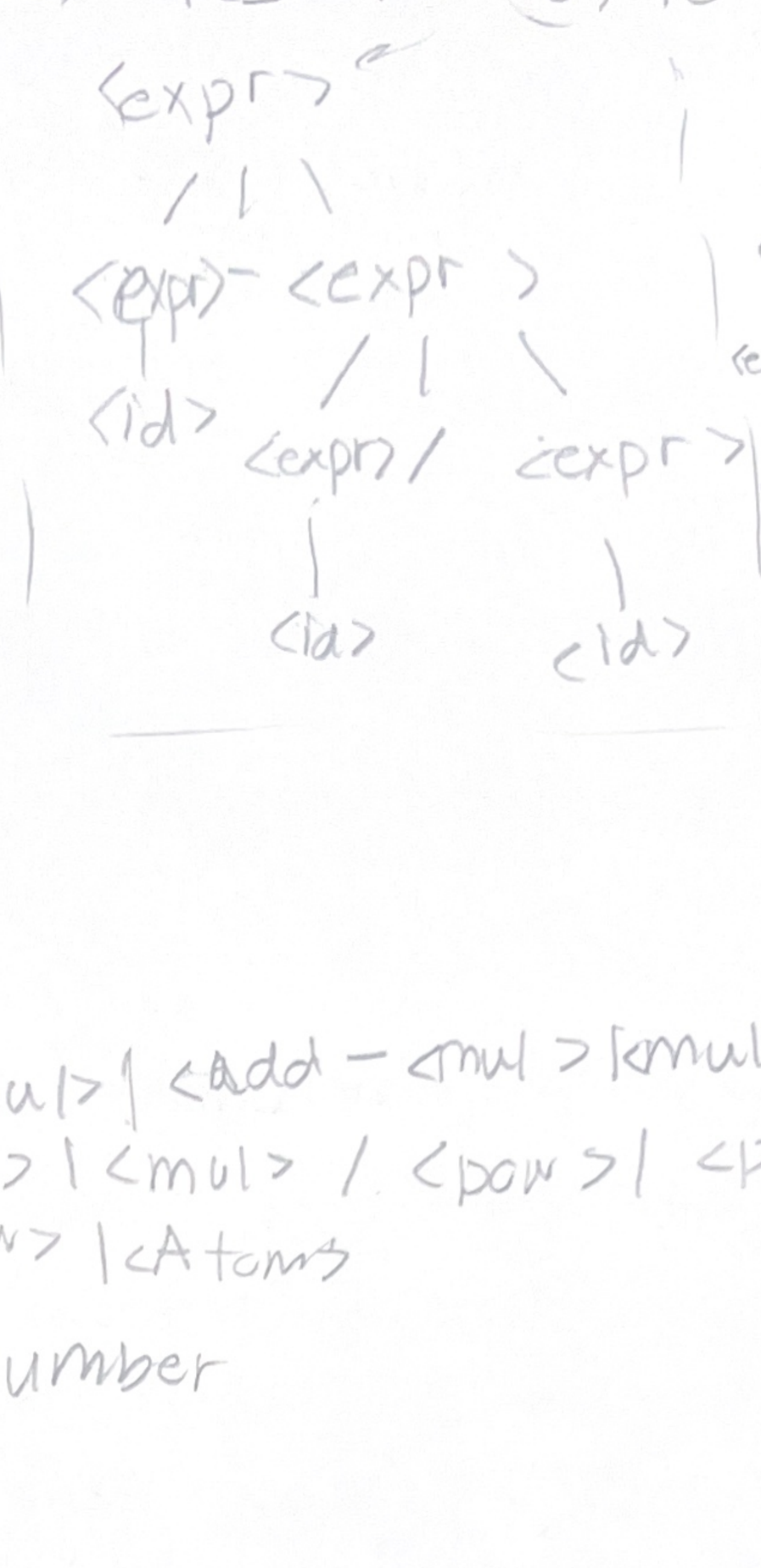

<stm+>
if <expr> then <stm+>
true
if <expr> then <stm+> else <stm+>
False
<assign>
<assign>
<id> = <number
<id> = <number
x
1
y
2

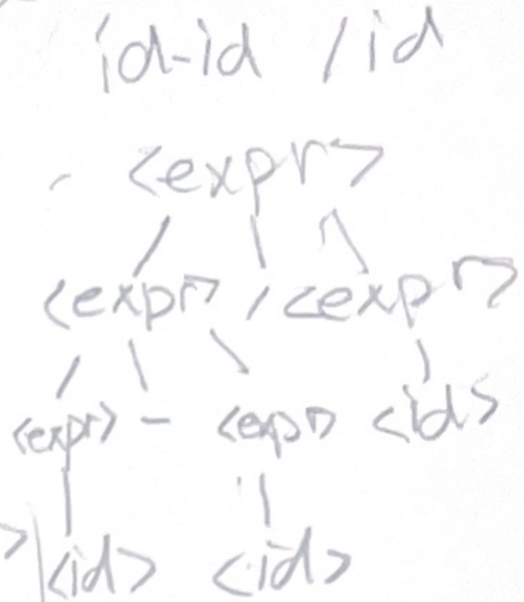

<stm+>
if <expr> then <stm+> else <stm+>
true
if <expr> then <stm+>
<assign>
False
<assign>
<id> = <number>
<id> = <number>
y
2
<id> = <number>
y

3 a) 1) No



2) Yes id - id / id





b) <expr> ::= <add>

<add> ::= <add> + <mul> | <add - <mul> | <mul>

<mul> ::= <mul> * <pow> | <mul> / <pow> | <pow>

<pow> ::= <Atom> ^ <pow> | <Atom>

<Atom> ::= (<expr>) | <number