

# 1. Token 정의하기

TYPE int|char|boolean|string

DIGIT 0|1|2|...|9

ALPHABET\_T a|b|c|...|z

ALPHABET\_NT A|B|C|...|Z

KEYWORD if|else|while|class|return|public  
|static|void|main|args|package

ARITHMETIC\_OPERATOR +|-|\*|/

ASSIGNMENT\_OPERATOR =

COMPARISON\_OPERATOR <|>|==|!=|<=|>=

SEMI ;

LBRACE {

RBRACE }

LPAREN (

RPAREN )

LBRACK [

RBRACK ]

COMMA ,

WHITESPACE \t| \n| ' |

## 2. 토큰 Token Regular Expression 으로 나타내기

Integer  $(-)^*[0-9]^+ | 0$

Literal-String  $"([A-Z]^+ | [a-z]^+ | [" "]^+)^*"$

Identifier  $[a-z]^+ [0-9, a-z]^* | [-]^+ [0-9, a-z]^*$

Type  $int | char | boolean | String$

Arithmetic\_Operator  $+ | - | * | /$

Assignment\_Operator  $=$

Comparison\_Operator  $< | > | == | != | <= | >=$

Semi  $;$

Lbrace  $\{$

Rbrace  $\}$

Lparen  $($

Rparen  $)$

Lbrack  $[$

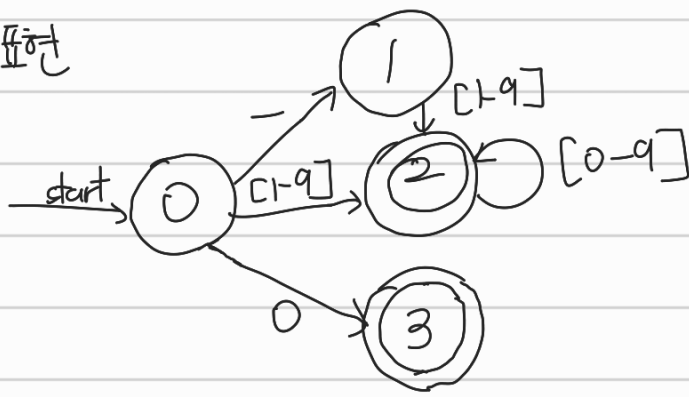
Rbrack  $]$

Comma  $,$

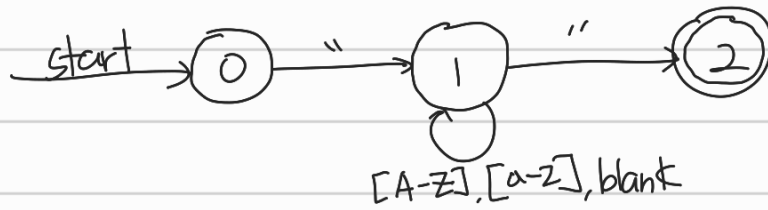
Whitespace  $\backslash t / \backslash n / ' '$

### 3. DFA 設計

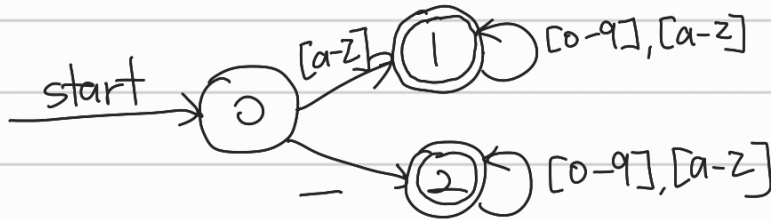
#### • Integer



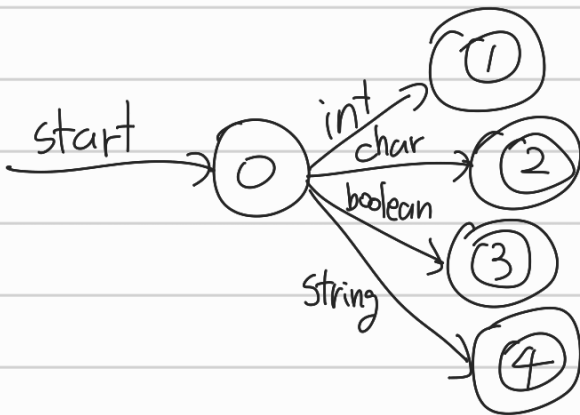
#### • Literal-String



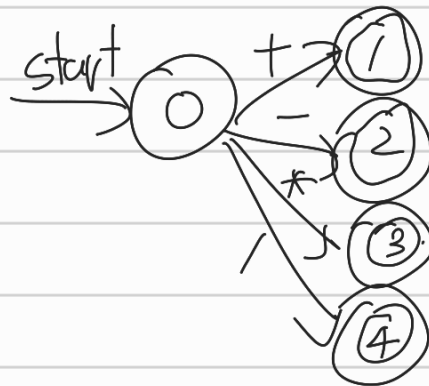
#### • Identifier



#### • Type

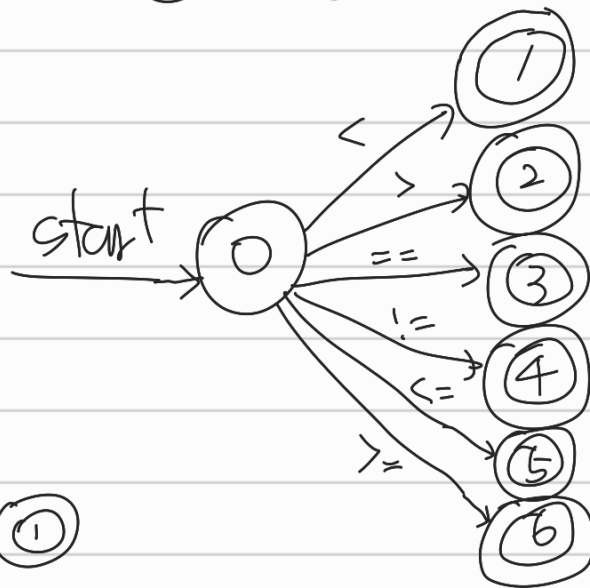


#### • Arithmetic-Operator



• Assignment - Operator  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{=}$  (1)

• Comparison - Operator



• Semi  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{;}$  (1)

• Lbrace  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{\{}$  (1)

• Rbrace  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{\{}$  (1)

• Lparen  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{(}$  (1)

• Rparen  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{)}$  (1)

• Lbrack  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{[}$  (1)

• Rbrack  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{]}$  (1)

• Commma  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{,}$  (1)

• Whitespace  $\xrightarrow{\text{start}}$  (0)  $\xrightarrow{\text{wt}}$  (1)  
 $\xrightarrow{\text{wn}}$  (2)  
 $\xrightarrow{\text{blank}}$  (3)