**Implementation of**

**a lexical analyzer**

|  |  |
| --- | --- |
| **분반** | 02분반 |
| **학과** | 컴퓨터공학부 |
| **팀** | TEAM 10 |
| **팀원** | 임창환, 방성원 |

# Define tokens

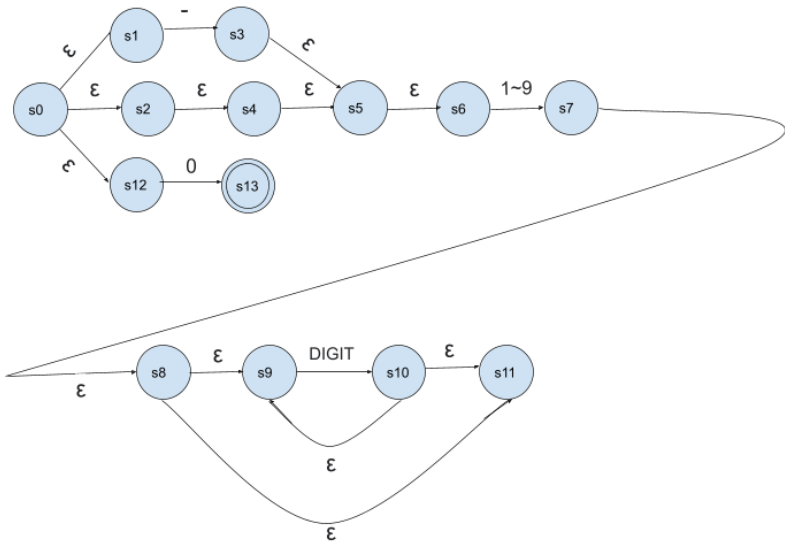
|  |  |
| --- | --- |
| int, char, bool, string | VTYPE |
| Identifier of variables and functions | ID |
| Signed integer, Literal string, Single character, Boolean string | VALUE |
| Comparison operators | COMP |
| Arithmetic operators | ARITH |
| Assignment operator | ASSIGN |
| Keywords for special s tatement | KEYWORD |
| Left Parantheses | LPARAN |
| Right Parantheses | RPARAN |
| Left Brace | LBRACE |
| Right Brace | RBRACE |
| Left Bracket | LBRACKET |
| Right Bracket | RBRACKET |
| Semicolon | SEMI |
| Symbol for separating | SEPARATE |

# Regular expressions

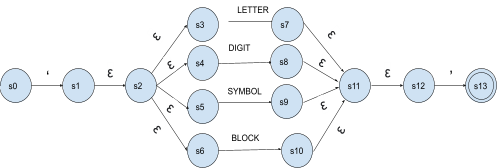


3. construct a NFA

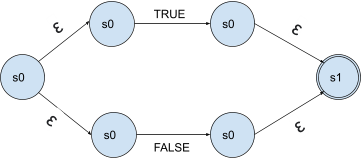
Signed integer



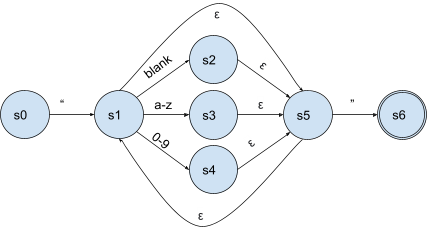
Single character

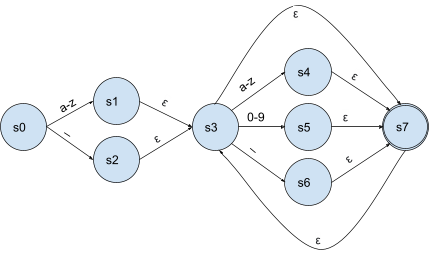


Boolean string

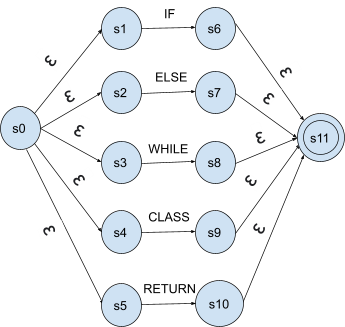


Literal String

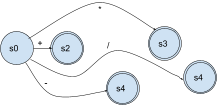
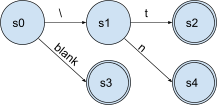


An Identifier of variables and functions 

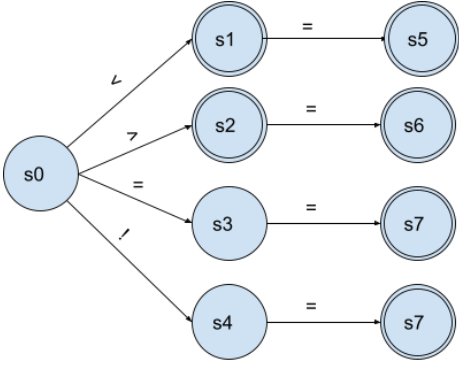
Keywords for special s tatement



Arithmetic operators WHITESPACE

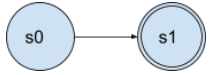


Comparison operators



A terminating symbol of statements, Assignment operators,

left / right parentheses, left / right braces, left / right brackets, separate, zero



|  |  |  |  |
| --- | --- | --- | --- |
|  | - | 1~9 | 0 |
| T0 | T1 | T2 | T3 |
| T1 | ∅ | T2 | ∅ |
| T2 | ∅ | T4 | T4 |
| T3 | ∅ | ∅ | ∅ |
| T4 | ∅ | T4 | T4 |

4. Construct a DFA from NFA

**Signed integer**

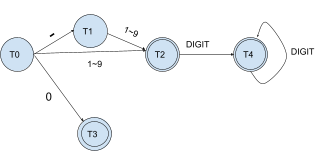
T0 = {s0,s1,s2,s4,s5,s6,s12}

T1 = {s3,s5,s6}

T2 = {s7,s8,s9,s11} final

T3 = {s13} final

T4 = {s9, s10, s11} final



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ‘ | Letter | Digit | Symbol | block | ’ |
| T0 | T1 |  |  |  |  |  |
| T1 |  | T2 | T3 | T4 | T5 |  |
| T2 |  |  |  |  |  | T6 |
| T3 |  |  |  |  |  | T6 |
| T4 |  |  |  |  |  | T6 |
| T5 |  |  |  |  |  | T6 |
| T6 |  |  |  |  |  |  |

**Single character**

T0 = {s0}

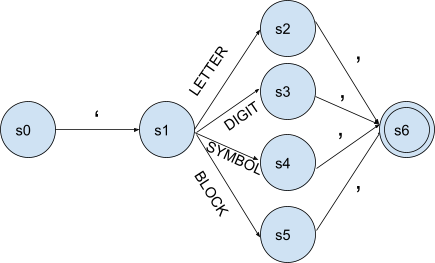
T1 = {s1, s2, s3, s4, s5, s6}

T2 = {s7,s11,s12}

T3 = {s8, s11, s12}

T4 = {s9, s11, s12}

T5 = {s10, s11, s12}

T6 = {s13} final

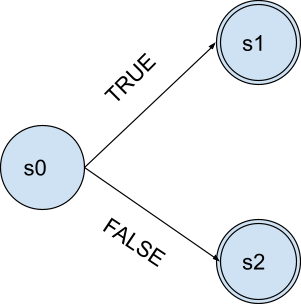
**Boolean string**

|  |  |  |
| --- | --- | --- |
|  | TRUE | FALSE |
| T0 | T1 | T2 |
| T1 |  |  |
| T2 |  |  |

T0 = {s0, s1, s2}

T1 = {s3, s5} final

T2 = {s4, s5} final



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | “ | blank | a-z | 0-9 | ” |
| T0 | T1 |  |  |  |  |
| T1 |  | T2 | T3 | T4 | T5 |
| T2 |  | T2 | T3 | T4 | T5 |
| T3 |  | T2 | T3 | T4 | T5 |
| T4 |  | T2 | T3 | T4 | T5 |
| T5 |  |  |  |  |  |

**Literal String**

T0 = {s0}

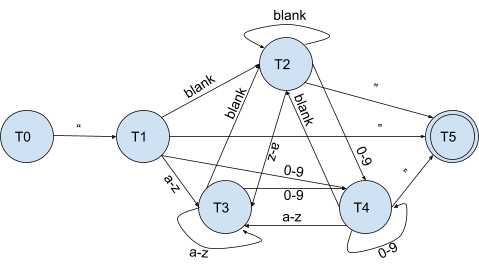
T1 = {s1,s5}

T2 = {s2,s5,s1}

T3 = {s3,s5,s1}

T4 = {s4,s5,s1}

T5 = {s6} final



|  |  |  |  |
| --- | --- | --- | --- |
|  | a-z | \_ | 0-9 |
| T0 | T1 | T2 |  |
| T1 | T3 | T5 | T4 |
| T2 | T3 | T5 | T4 |
| T3 | T3 | T5 | T4 |
| T4 | T3 | T5 | T4 |
| T5 | T3 | T5 | T4 |

**An Identifier of variables and functions**

T0 = {s0}

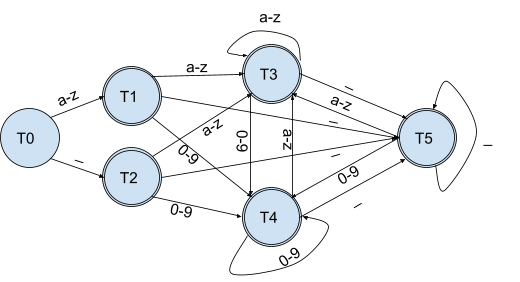
T1 = {s1,s3,s7} final

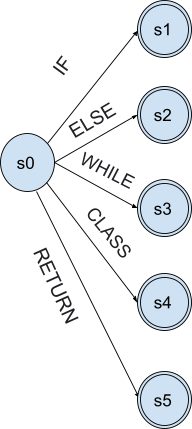
T2 = {s2,s3,s7} final

T3 = {s4,s7,s3} final

T4 = {s5,s7,s3} final

T5 = {s6,s7,s3} final



**Keywords for special statement**

T0 = {s0, s1, s2, s3, s4, s5}

T1 = {s6, s11} final

T2 = {s7, s11} final

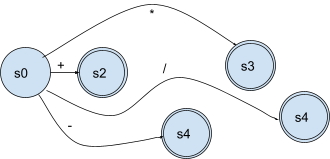
T3 = {s8, s11} final

T4 = {s9, s11} final

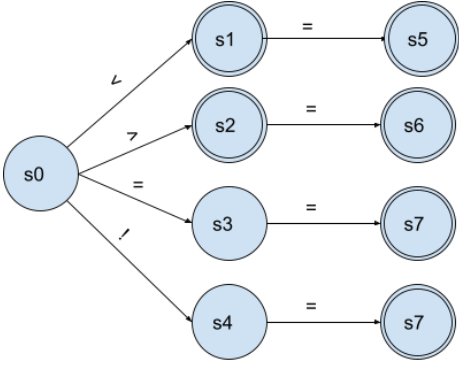
T5 = {s10, s11} final

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | IF | ELSE | WHILE | CLASS | RETURN |
| T0 | T1 | T2 | T3 | T4 | T5 |
| T1 |  |  |  |  |  |
| T2 |  |  |  |  |  |
| T3 |  |  |  |  |  |
| T4 |  |  |  |  |  |
| T5 |  |  |  |  |  |

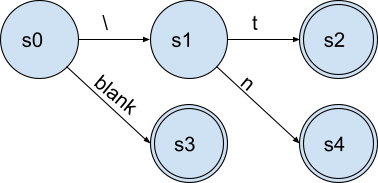
**Arithmetic operators**



**Comparison operators**

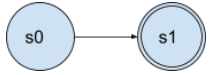


**WHITESPACE**



A terminating symbol of statements, Assignment operators,

left / right parentheses, left / right braces, left / right brackets, separate, zero

**