컴파일러 과제 1 20186889 전용환

Regular expression

- Variable type

int | char | boolean | string

- Signed integer

Digits = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Digits-nonzero = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

0 | (ε | -) Digit_nonzero Digit*

- Single character

Alphabet = a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

special symbol = ! | ? | # | % | / | @ | $ | ^ | * | + | - | _ | ~ | / | ∞ | . | ; | < | > | = | ' | " | ;

' Digits | Alphabet | special symbol | _ '
그외에

- Literal string

" (Digits | Alphabet |) * "

- Boolean string

true | false

- Identifier

_ | Alphabet (_ | Alphabet | Digits)*

- keyword

if | else | while | class | return

- Arithmetic operators

+ | - | * | /

- Assignment operator

=

- Comparison operators

  < | > | == | != | <= | >=

- terminating symbol of statement

  ;

- A pair of symbols for defining area / scope of variables and function

  { | }

- A pair of symbol for using an array

  [|]

- A pair of symbol for indicating a function / statement

  (|)

- A symbol for separating input arguments in function

  ,

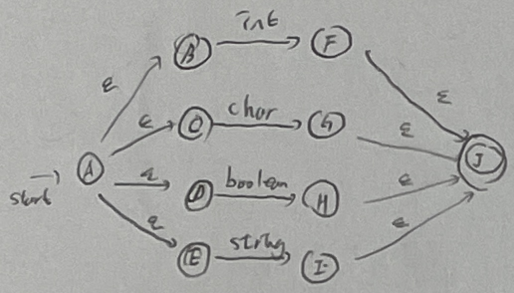
- white space

  _ | \t | \n

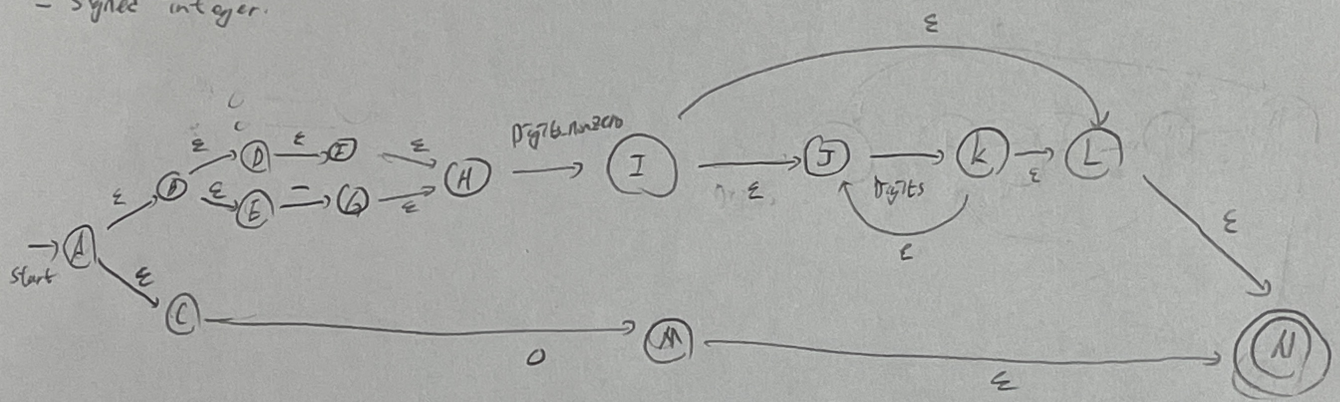
컴파일러 과제 1 20186889 권용찬

NFA

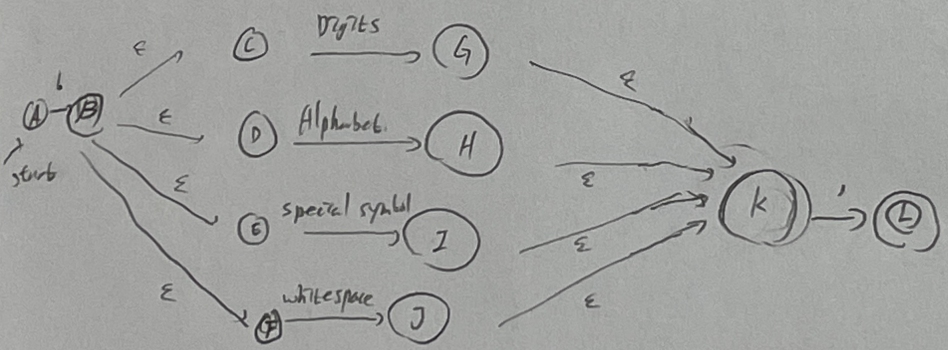
- variable type.



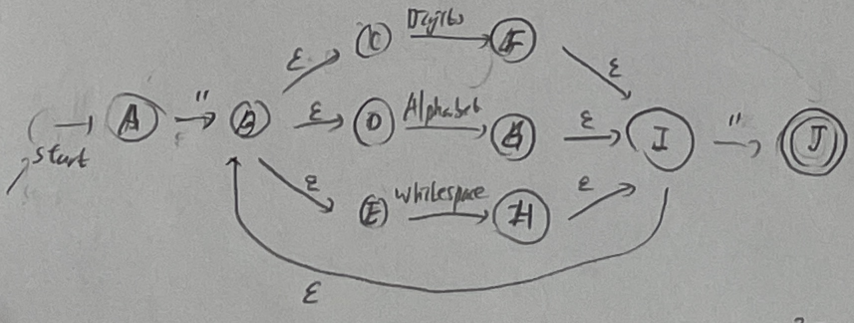
- signed integer.



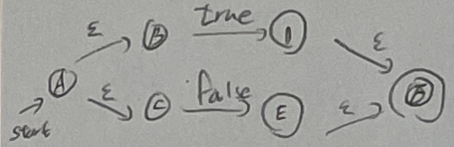
- single character



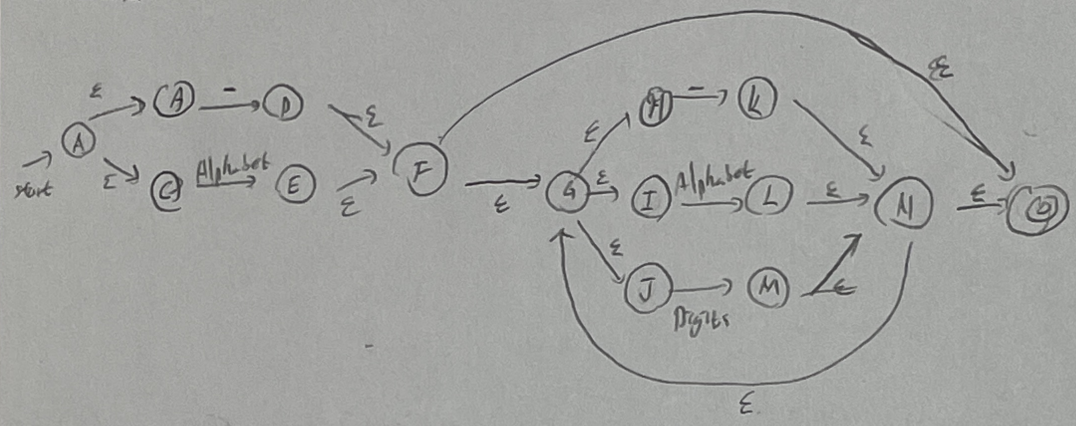
- Literal string



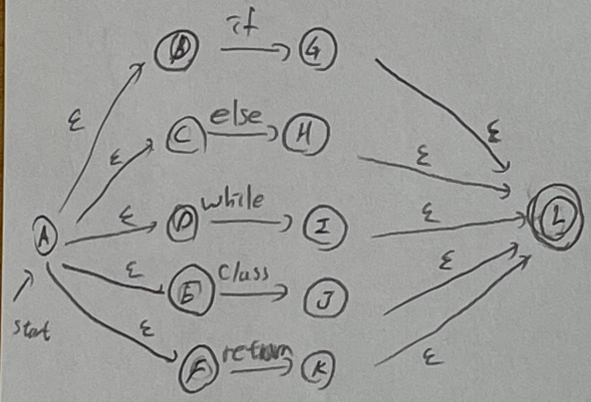
- Boolean string



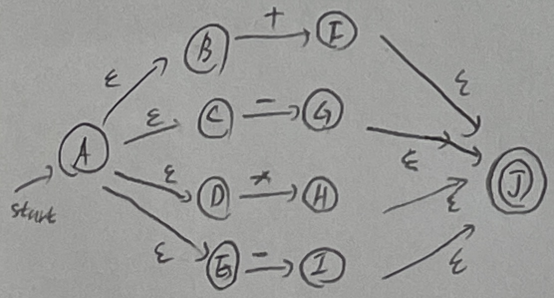
- Identifier.



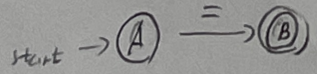
- keyword



- Arithmetic operators

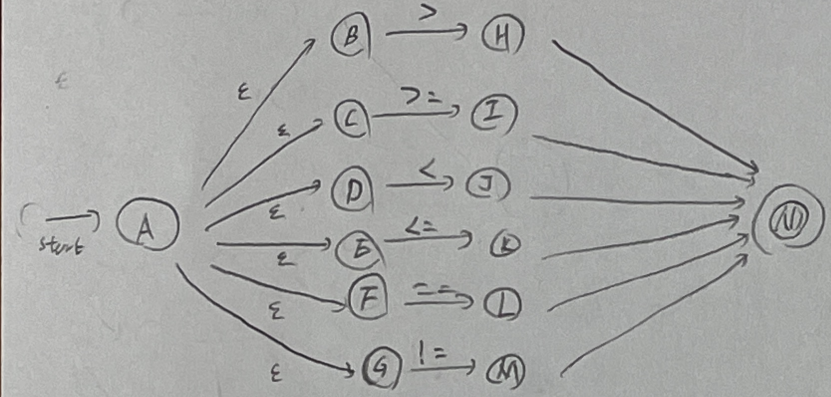


- Assignment operator

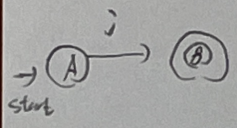


컴파일러 과제 1. 2018 6889 권용찬

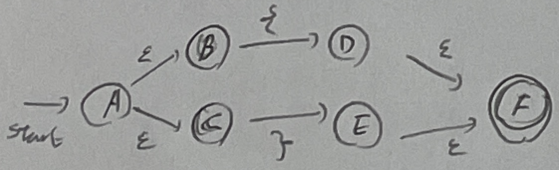
Comparison operator



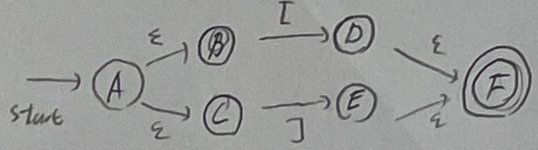
terminating symbol of statement



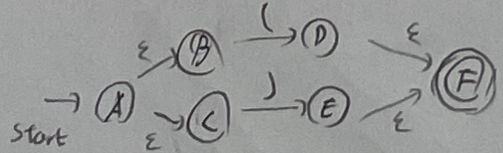
Symbol for defining area / scope.



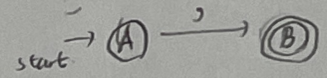
Symbol for using an array



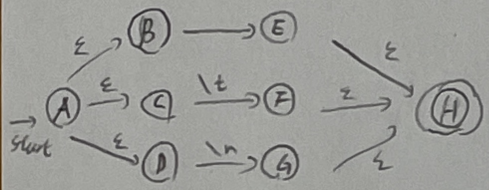
Symbol for indicating a function / statement.



Symbol for separating input arguments in function



White space



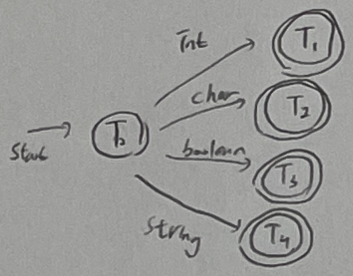
컴파일러 과제 1 20186889 권용찬

DFA

- variable type

	int	char	boolean	string
T_0	T_1	T_2	T_3	T_4
T_1	∅	∅	∅	∅
T_2	∅	∅	∅	∅
T_3	∅	∅	∅	∅
T_4	∅	∅	∅	∅

- T_0 = ε-closure (A) = {A, B, C, D}
- T_1 = ε-closure (δ(T_0, int)) = ε-closure (F) = {F, J}
- T_2 = ε-closure (δ(T_0, char)) = ε-closure (G) = {G, J}
- T_3 = ε-closure (δ(T_0, boolean)) = ε-closure (H) = {H, J}
- T_4 = ε-closure (δ(T_0, string)) = ε-closure (I) = {I, J}

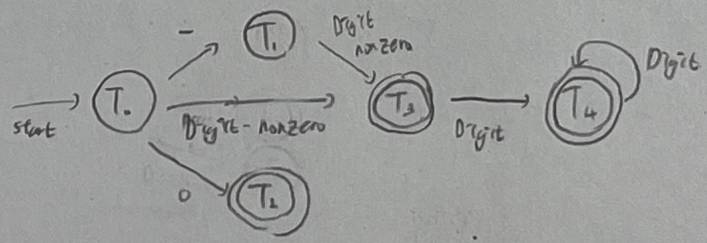


- Signed integer

- T_0 = ε-closure (A) = {A, B, C, D, E, F, H}
- T_1 = ε-closure (δ(T_0, −)) = ε-closure (G) = {G, H}
- T_2 = ε-closure (δ(T_0, 0)) = ε-closure (M) = {M, N}
- T_3 = ε-closure (δ(T_0, Digit-nonzero)) = ε-closure (I) = {I, J, L, N}

  ε-closure (δ(T_1, Digit-nonzero)) = ε-closure (I) = {I, J, L, N} = T_3
- T_4 = ε-closure (δ(T_3, Digits)) = ε-closure (K) = {K, L, N}

	−	0	Digits-nonzero	Digit
T_0	T_1	T_2	T_3	∅
T_1	∅	∅	T_3	∅
T_2	∅	∅	∅	∅
T_3	∅	∅	∅	T_4
T_4	∅	∅	∅	T_4



-Single character.

- T_0 = ε-closure (A) = {A}

T_1 = ε-closure (δ(T_0, ')) = ε-closure (B) = {B, C, D, E, F}

T_2 = ε-closure (δ(T_1, Digits)) = ε-closure (G) = {G, K}

T_3 = ε-closure (δ(T_1, Alphabet)) = ε-closure (H) = {H, K}

T_4 = ε-closure (δ(T_1, special symbol)) = ε-closure (I) = {I, K}

T_5 = ε-closure (δ(T_1, whitespace)) = ε-closure (J) = {J, K}

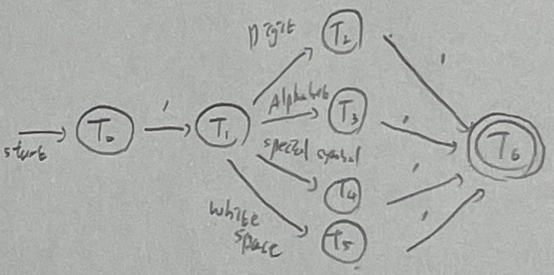
T_6 = ε-closure (δ(T_2, ')) = ε-closure (L) = {L}

ε-closure (δ(T_3, ')) = ε-closure (L) = {L} = T_6

ε-closure (δ(T_4, ')) = ε-closure (L) = {L} = T_6

ε-closure (δ(T_5, ')) = ε-closure (L) = {L} = T_6

	'	Digits	Alphabet	special symbol	whitespace
T_0	T_1	∅	∅	∅	∅
T_1	∅	T_2	T_3	T_4	T_5
T_2	T_6	∅	∅	∅	∅
T_3	T_6	∅	∅	∅	∅
T_4	T_6	∅	∅	∅	∅
T_5	T_6	∅	∅	∅	∅
T_6	∅	∅	∅	∅	∅



-Literal string

- T_0 = ε-closure (A) = {A}
- T_1 = ε-closure (δ(T_0, ")) = ε-closure (B) = {B, L, D, E}
- T_2 = ε-closure (δ(T_1, Digits)) = ε-closure (F) = {B, C, D, E, F, I}
- T_3 = ε-closure (δ(T_1, Alphabet)) = ε-closure (G) = {B, C, D, E, G, I}
- T_4 = ε-closure (δ(T_1, whitespace)) = ε-closure (H) = {B, C, D, E, H, I}
- ε-closure (δ(T_2, Digits)) = T_2
  ε-closure (δ(T_2, Alphabet)) = T_3
  ε-closure (δ(T_3, Alphabet)) = T_3
  ε-closure (δ(T_4, Digits)) = T_2
  ε-closure (δ(T_4, whitespace)) = T_4
  ε-closure (δ(T_3, whitespace)) = T_4
  ε-closure (δ(T_3, Digits)) = T_2
  ε-closure (δ(T_4, whitespace)) = T_4
  ε-closure (δ(T_4, Alphabet)) = T_3

T_5 = ε-closure (δ(T_1, ")) = ε-closure (J) = {J}

ε-closure (δ(T_3, ")) = T_7

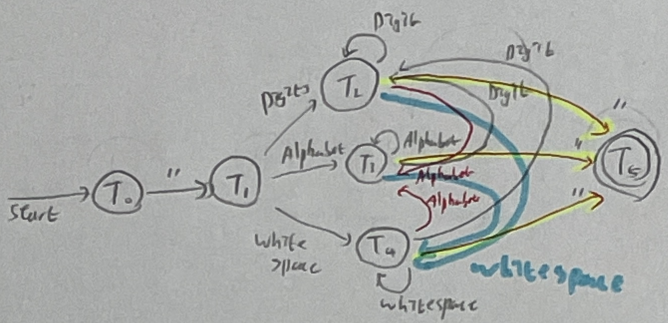
ε-closure (δ(T_4, ")) = T_5

컴파일러 과제1 2018664069 권용빈

DFA

-Literal string

	"	Digits	Alphabet	whitespace
T_0	T_1	∅	∅	∅
T_1	∅	T_2	T_3	T_4
T_2	T_5	T_2	T_3	T_4
T_3	T_5	T_2	T_3	T_4
T_4	T_5	∅	∅	∅
T_5	∅	∅	∅	∅



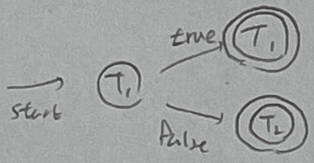
-Bolean string

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, true)) = \varepsilon\text{-closure}(D) = \{D, F\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0, False)) = \varepsilon\text{-closure}(E) = \{E, F\}$

	true	false
T_0	T_1	T_2
T_1	∅	∅
T_2	∅	∅



-Identifier

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, -)) = \varepsilon\text{-closure}(D) = \{D, F, G, H, I, J, O\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0, Alphabet)) = \varepsilon\text{-closure}(E) = \{E, F, G, H, I, J, O\}$

$T_3 = \varepsilon\text{-closure}(\delta(T_1, -)) = \varepsilon\text{-closure}(K) = \{G, H, I, J, K, N, O\}$

$T_4 = \varepsilon\text{-closure}(\delta(T_1, Alphabet)) = \varepsilon\text{-closure}(L) = \{G, H, I, J, L, N, O\}$

$T_5 = \varepsilon\text{-closure}(\delta(T_1, Digits)) = \varepsilon\text{-closure}(M) = \{G, H, I, J, M, N, O\}$

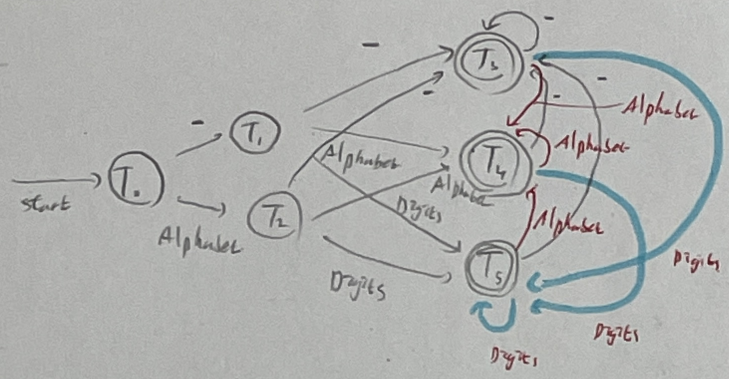
$\varepsilon\text{-closure}(\delta(T_2, -)) = T_3$ $\quad \varepsilon\text{-closure}(\delta(T_2, Digits)) = T_5$ $\quad \varepsilon\text{-closure}(\delta(T_3, Alphabet)) = T_4$

$\varepsilon\text{-closure}(\delta(T_2, Alphabet)) = T_4$ $\quad \varepsilon\text{-closure}(\delta(T_3, -)) = T_3$ $\quad \varepsilon\text{-closure}(\delta(T_3, Digits)) = T_5$

$\varepsilon\text{-closure}(\delta(T_4, -)) = T_3$ $\quad \varepsilon\text{-closure}(\delta(T_4, Alphabet)) = T_4$ $\quad \varepsilon\text{-closure}(\delta(T_4, Digits)) = T_5$

$\varepsilon\text{-closure}(\delta(T_5, -)) = T_3$ $\quad \varepsilon\text{-closure}(\delta(T_5, Alphabet)) = T_4$ $\quad \varepsilon\text{-closure}(\delta(T_5, Digits)) = T_5$

	-	Alphabet	Digits
T_0	T_1	T_2	$\emptyset$
T_1	T_3	T_4	T_5
T_2	T_3	T_4	T_5
T_3	T_3	T_4	T_5
T_4	T_3	T_4	T_5
T_5	T_3	T_4	T_5



- keyword

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C, D, E, F\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, if)) = \varepsilon\text{-closure}(G) = \{G, L\}$

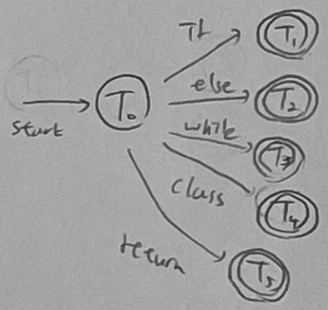
$T_2 = \varepsilon\text{-closure}(\delta(T_0, else)) = \varepsilon\text{-closure}(H) = \{H, L\}$

$T_3 = \varepsilon\text{-closure}(\delta(T_0, while)) = \varepsilon\text{-closure}(I) = \{I, L\}$

$T_4 = \varepsilon\text{-closure}(\delta(T_0, class)) = \varepsilon\text{-closure}(J) = \{J, L\}$

$T_5 = \varepsilon\text{-closure}(\delta(T_0, return)) = \varepsilon\text{-closure}(K) = \{K, L\}$

	if	else	while	class	return
T_0	T_1	T_2	T_3	T_4	T_5
T_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$



- Arithmetic operators

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C, D, E\}$

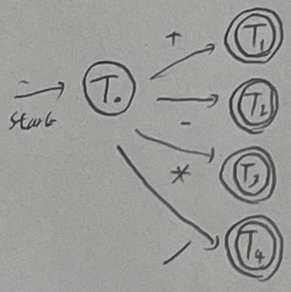
$T_1 = \varepsilon\text{-closure}(\delta(T_0, +)) = \varepsilon\text{-closure}(F) = \{F, J\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0, -)) = \varepsilon\text{-closure}(G) = \{G, J\}$

$T_3 = \varepsilon\text{-closure}(\delta(T_0, *)) = \varepsilon\text{-closure}(H) = \{H, J\}$

$T_4 = \varepsilon\text{-closure}(\delta(T_0, /)) = \varepsilon\text{-closure}(I) = \{I, J\}$

	+	-	*	/
T_0	T_1	T_2	T_3	T_4
T_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$



컴파일러 과제 1 20186889 권용한

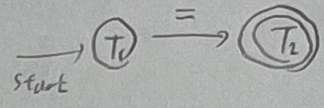
DFA

- Assignment operator

$T_0 = \varepsilon\text{-closure}(A) = \{A\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, =)) = \varepsilon\text{-closure}(B) = \{B\}$

	=
T_1	T_2
T_2	$\emptyset$



- Comparison operator

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C, D, E, F, G\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, >)) = \varepsilon\text{-closure}(H) = \{H, N\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0, >=)) = \varepsilon\text{-closure}(I) = \{I, N\}$

$T_3 = \varepsilon\text{-closure}(\delta(T_0, <)) = \varepsilon\text{-closure}(J) = \{J, N\}$

$T_4 = \varepsilon\text{-closure}(\delta(T_0, <=)) = \varepsilon\text{-closure}(K) = \{K, N\}$

$T_5 = \varepsilon\text{-closure}(\delta(T_0, ==)) = \varepsilon\text{-closure}(L) = \{L, N\}$

$T_6 = \varepsilon\text{-closure}(\delta(T_0, !=)) = \varepsilon\text{-closure}(M) = \{M, N\}$

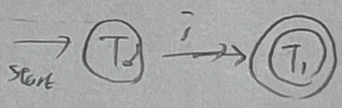
	>	>=	<	<=	==	!=
T_0	T_1	T_2	T_3	T_4	T_5	T_6
T_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_6	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

- Terminating symbol

$T_0 = \varepsilon\text{-closure}(A) = \{A\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, ;)) = \varepsilon\text{-closure}(B) = \{B\}$

	;
T_0	T_1
T_1	$\emptyset$



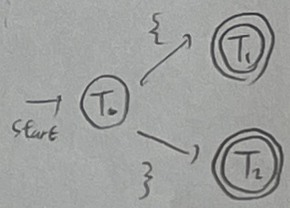
- Symbol for defining area/scope

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, \{)) = \varepsilon\text{-closure}(D) = \{D, F\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0, \})) = \varepsilon\text{-closure}(E) = \{E, F\}$

	{	}
T_0	T_1	T_2
T_1	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$



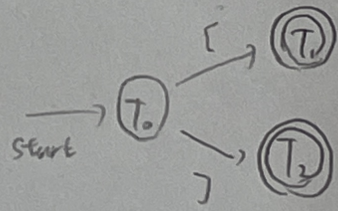
- Symbol for using an array

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, [)) = \varepsilon\text{-closure}(D) = \{D, F\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0,])) = \varepsilon\text{-closure}(E) = \{E, F\}$

	[	]
T_0	T_1	T_2
T_1	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$



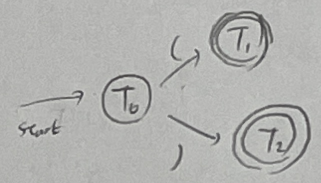
- Symbol for indicating a function / statement.

$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, ())) = \varepsilon\text{-closure}(D) = \{D, F\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0,))) = \varepsilon\text{-closure}(E) = \{E, F\}$

	(	)
T_0	T_1	T_2
T_1	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$

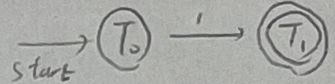


- Symbol for separating input arguments in function

$T_0 = \varepsilon\text{-closure}(A) = \{A\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, ,)) = \varepsilon\text{-closure}(B) = \{B\}$

	,
T_0	T_1
T_1	$\emptyset$



- while space

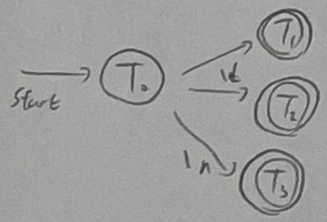
$T_0 = \varepsilon\text{-closure}(A) = \{A, B, C, D\}$

$T_1 = \varepsilon\text{-closure}(\delta(T_0, \_)) = \varepsilon\text{-closure}(E) = \{E, H\}$

$T_2 = \varepsilon\text{-closure}(\delta(T_0, \backslash t)) = \varepsilon\text{-closure}(F) = \{F, H\}$

$T_3 = \varepsilon\text{-closure}(\delta(T_0, \backslash n)) = \varepsilon\text{-closure}(G) = \{G, H\}$

		\t	\n
T_0	T_1	T_2	T_3
T_1	$\emptyset$	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$	$\emptyset$
T_3	$\emptyset$	$\emptyset$	$\emptyset$



Document

20186889 권용한

c언어를 사용하여 과제를 수행하였으며, 우선 정수, 변수나 함수 명, 문자와 문자열은 다른 분류체계에 비해 변수가 많고 서로 겹치는 경우가 많아서 우선 다른 분류사항들을 우선 1대 1로 비교하여 맞다면 그에 해당하는 분류체계에 포함하도록 조정하였습니다.

적절한 동작을 하기 위해서 우선 다양한 심볼들에 따라 우선 구분하여 temp.txt파일에 저장하여 토큰화를 하였습니다. 이때 토큰 구분을 하면서 line에 대한 정도(몇번째 줄인지에 대한 정보)가 사라지기 때문에, 우선 몇번째 line인지 우선 저장하며, 각 line별로 토큰화를 진행하였습니다.

토큰화가 진행된 이후에는 각 토큰에 해당하는 분류를 찾아 output.txt파일에 저장하도록 하였으며, 이때 각각의 토큰들에 해당하는 분류체계 중 적절하지 못한 문자가 섞여있다면 이를 오류로 판단하여 파일에 해당 line과 error 원인을 적어주었습니다.

Single character와 literal string의 경우 digits를 signed integer와 공유하게 되며, 또한 Identifier와 alphabet이나 다른 symbol도 함께 사용 가능하기 때문에 분류에 어려움이 있을 수 있는데, 이러한 것을 방지하고자 '와 "가 홀수번째 등장한다면 그 이후에 등장하는 input은 single character이나 literal sting으로 인식하게 하였고, 다음번 '나 "가 등장하기까지의 영역까지를 그 범위로 인식하도록 하였습니다. 그리고 나머지 경우에 대해 분류를 하게 되며, 이러한 순서로 분류하며 integer나 ID와 같은 복잡한 경우를 제외하곤 1대 1로 매칭하며 분류하고, integer나 ID같은 경우 처음 등장하는 문자가 다르기 때문에 이를 이용해 처음 이 둘을 구분하고, integer나 ID에서 허용되지 않는 문자열이 등장하는 경우에 대해서는 오류로 간주하였습니다.