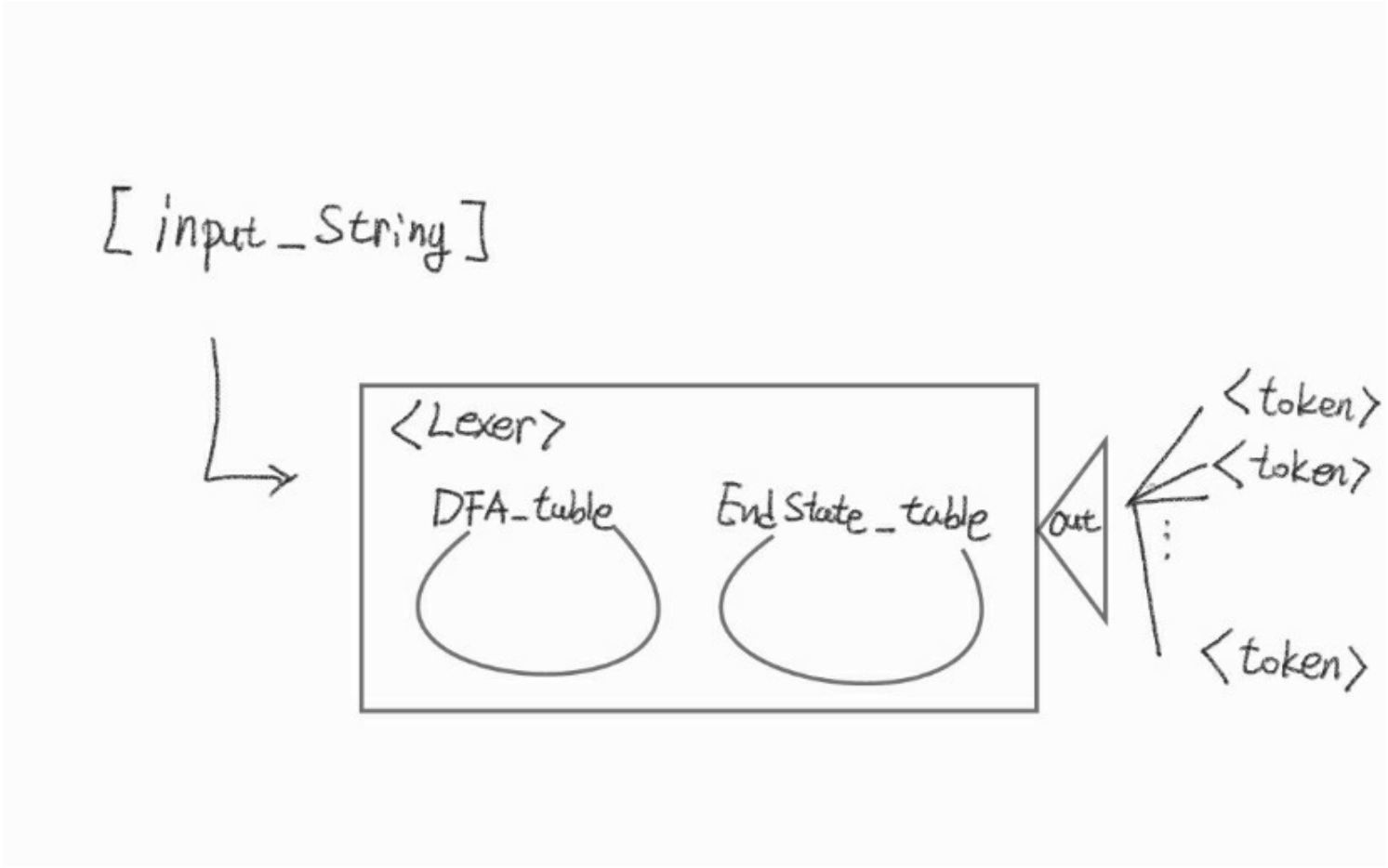
Compiler Team Project #1
-Lexical Analyzer

02 분반 26 조
20175429 유승훈
20170807 박민기

- **overall procedure**

일괄된 흐름을 가진 하나의 큰 오토마타로 lexical analyzer 를 구현하였습니다.
초기 구상은 nfa 상의 node 들을 전부 하나의 오토마타에 표기하려 했으나, 해당 방식으로 프로그램을 작성하고 dfa table 을 작성하는 것에 어려움을 느껴 여러 개의 오토마타로 나누어 동작하도록 하였습니다.
처음 input 으로부터 공통된 위치(오토마타: 1, 상태: T0)에서 시작하여, 이후 각각의 오토마타로 process 를 전달시켜 구문 분석을 진행합니다.
현재 위치에서 입력된 input 을 통해 다음 이동할 수 있는 위치를 dfa_table 이라는 3 차원 pair<int,int> 배열에 저장해 두었습니다. 다음위치로 이동이 제한될 경우를 dfa_table[automata][state] == {0,0}으로 특정하였고, 이 경우에 EndState_table 을 확인하여 정상종료와 비정상 종료를 확인하게 됩니다.
EndState_table 속에 현재 위치가 정상 종료라고 표시되어 있을경우(EndState_table[automata][state] != "") 지금까지 받은 문자열을 하나의 token 으로 구분하여 answer 벡터에 저장합니다.
EndState_table 속에 현재위치가 비정상종료라고 표시되어 있을경우(EndState_table[automata][state] == "") 분석을 종료하고 에러 메세지를 출력하여 어느 문자열에 문제가 있었는지 출력합니다.
만약 분석이 정상적으로 종료된다면, answer 벡터 속 인자들을 입력을 받은 순서대로 토큰의 타입과 종류를 출력하여 줍니다.



- **Data structure**

typedef pair<int, int> ii;

오토마타와 상태 변수를 짝으로 저장하기 위해 새롭게 정의한 타입

typedef pair<string, string> ss;

토큰 저장시, 토큰의 타입과 Lexeme 을 짝으로 저장하기 위해 새롭게 저장한 타입

ii DFA_table[15][39][127];

현재 오토마타와 상태, 입력 문자를 통해 다음 오토마타와 상태를 저장하는 3 차원 배열

string EndState_table[15][39];

문자열 입력 종료시, 현재 오토마타와 상태를 통해 토큰의 타입을 특정할 수 있도록 만든 2 차원 배열

- **Algorithms**

****symbol '-' 의 처리방법***

'-'가 입력 됐을 시 오토마타 4 번으로 보내어 분석을진행합니다. 오토마타를 통해 분석 도중 '-'가 홀로 입력 되었을 경우엔 operator 로, 그렇지 않고 위에 '0'를 제외한 digit 등과 합쳐져서 나왔을 경우에는 이전에 입력된 token 과 연결 지어 그 의미가 맞는지 분석합니다.

identifier, signed_integer, RSparen 등의 뒤에 '-'가 올 경우, 이는 operator - 로 동작할 가능성이 있으므로 이경우 해당 입력을 operator '-'와 positive integer 로 구분하여 입력 받습니다.

**** identifier + special statement + boolean + Vtype***

위의 입력들은 알파벳 소문자나 대문자가 처음 입력으로 들어오는 token 이라는 공통점을 가지고 있습니다. 따라서 위의 입력들을 받았을 때 token table 을 정교하게 설계하여 특정한 의미를 갖는 문자열(special statement , boolean , Vtype)과 정확히 일치하며 끝난다면 해당하는 type 으로 인식합니다.
하지만 그렇지 않고 아무런 의미를 갖지 못하는 입력으로 인식된다면 그것을 identifier 로 인식합니다.

- **Example**

input:
0123456789
output:
<Signed_integer,0>
<Signed_integer,123456789>

input:
-1- was-1
output:
<Signed_integer,-1>
<Arithmetic_op,->
<Identifier,was>
<Arithmetic_op,->
<Signed_integer,1>

input:
it was if else white while"if" clase 1-+2 = (5)

output:
<Identifier,it>
<Identifier,was>
<special_if,if>
<special_else,else>
<Identifier,white>
<special_while,while>
<String,"if">
<Identifier,clase>
<Signed_integer,1>
<Arithmetic_op,->
<Arithmetic_op,+>
<Signed_integer,2>
<Assign_op,=>
< LSparen,(>
< Signed_integer,5>
<RSparen,)>

error case

input:
"it is a apple
output:
there can't be token like "it is a apple

input:
'ab'
output:
there can't be token like 'ab'

input:
"hello -=-! on
output:
there can't be token like " hello -
(find the first error and exit)

[Regular Expression]

ZERO = 0

DIGIT = 1|2|3|--- |8|9

LETTER = a|b|C--- |z|A|B|C|--- |Z

BLANK = /t | /n | blank

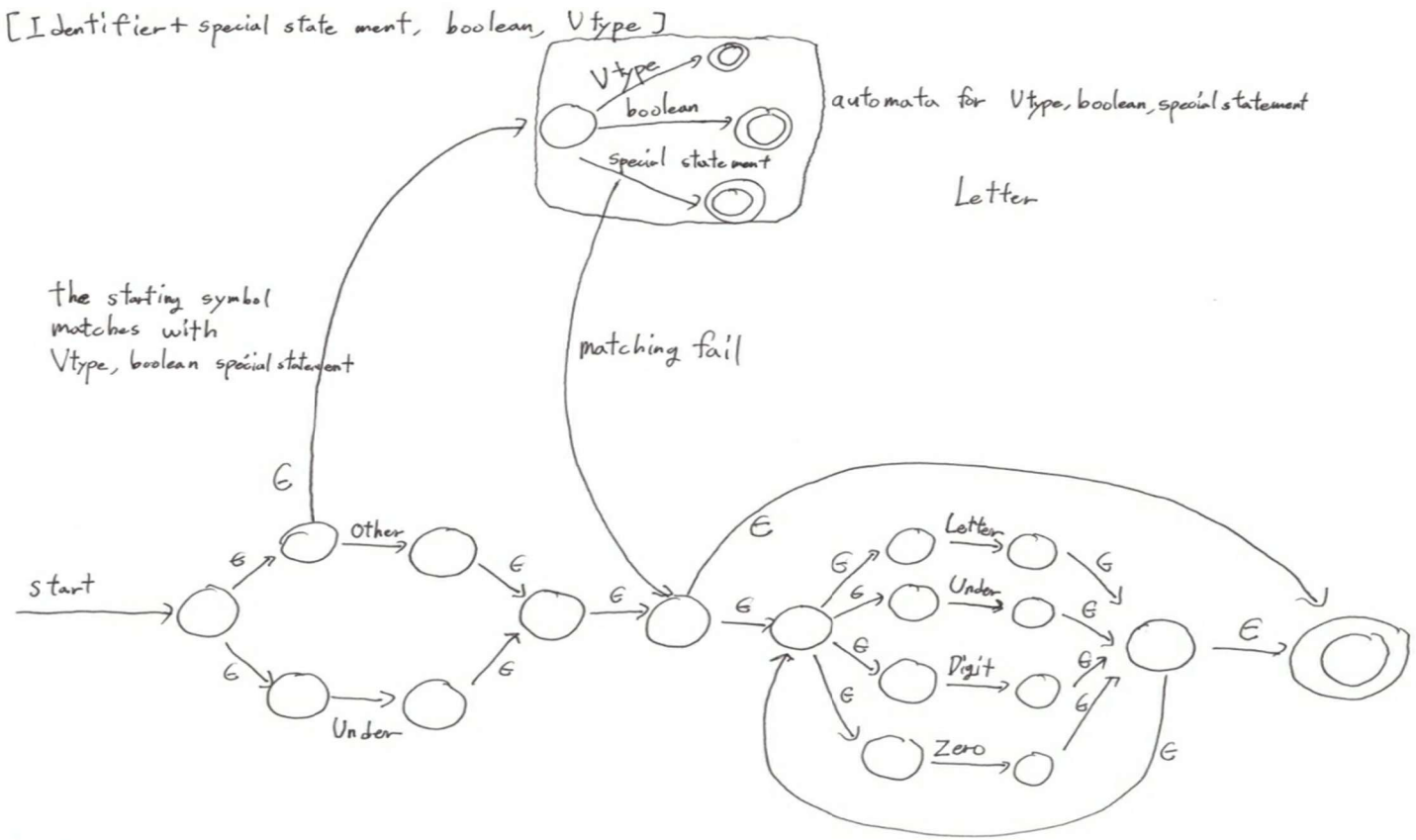
Other = def) Set of characters that are not Classified in Certain regular expression

Single_Character

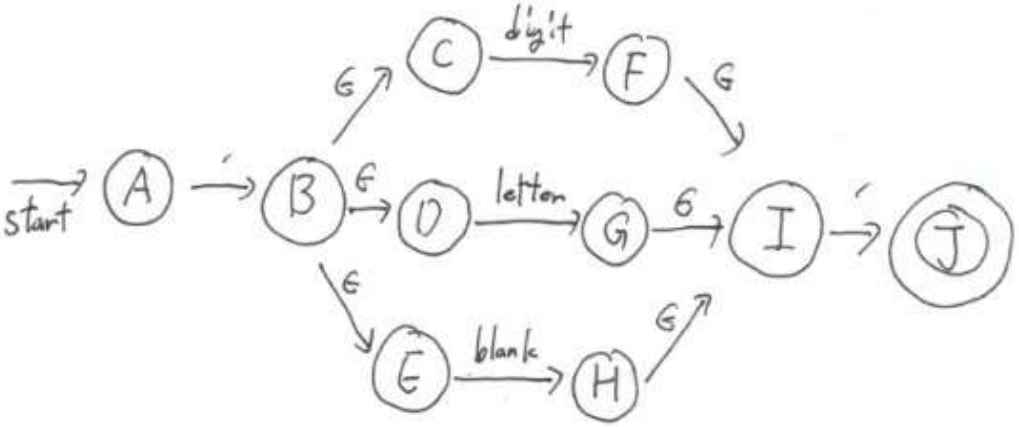
= '(DIGIT | ZERO | BLANK | LETTER)'

Literal_String

="(DIGIT | ZERO | BLANK | LETTER) (DIGIT | ZERO | BLANK | LETTER)*"


[Identifier+ special statement, boolean, Vtype]
Vtype
boolean
special statement
automata for Vtype, boolean, special statement
Letter
the starting symbol
matches with
Vtype, boolean special statement
matching fail
ϵ
start
Other
Under
Letter
Under
Digit
Zero

[single charactor]



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

$T_1 = \epsilon\text{-closure}(\delta(T_1, \cdot)) = \{B, C, D, E\}$

$T_2 = \epsilon\text{-closure}(\delta(T_1, digit)) = \epsilon\text{-closure}(F) = \{F, I\}$

$T_3 = \epsilon\text{-closure}(\delta(T_1, English\ letter)) = \epsilon\text{-closure}(G) = \{G, I\}$

$T_4 = \epsilon\text{-closure}(\delta(T_1, blank)) = \epsilon\text{-closure}(H) = \{H, I\}$

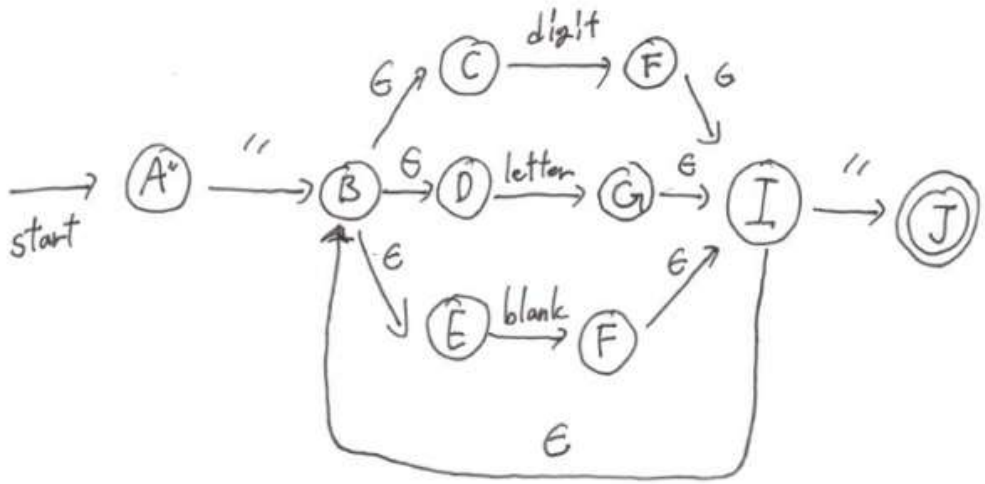
$T_5 = \epsilon\text{-closure}(\delta(T_2, ')) = \epsilon\text{-closure}(J) = \{J\}$

$\epsilon\text{-closure}(\delta([T_1, T_3, T_4, T_5], [digit, letter, blank])) = \phi$

$\epsilon\text{-closure}(\delta([T_1, T_5], ') = \phi$

	digit	letter	blank	'	other
T_0	ϕ	ϕ	ϕ	T_1	ϕ
T_1	T_2	T_3	T_4	ϕ	ϕ
T_2	ϕ	ϕ	ϕ	T_5	ϕ
T_3	ϕ	ϕ	ϕ	T_5	ϕ
T_4	ϕ	ϕ	ϕ	T_5	ϕ
T_5	ϕ	ϕ	ϕ	ϕ	ϕ

[Literal string]



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

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

$T_2 = \varepsilon\text{-closure}(\delta(T_1, \text{digit})) = \{F, I, B\}$

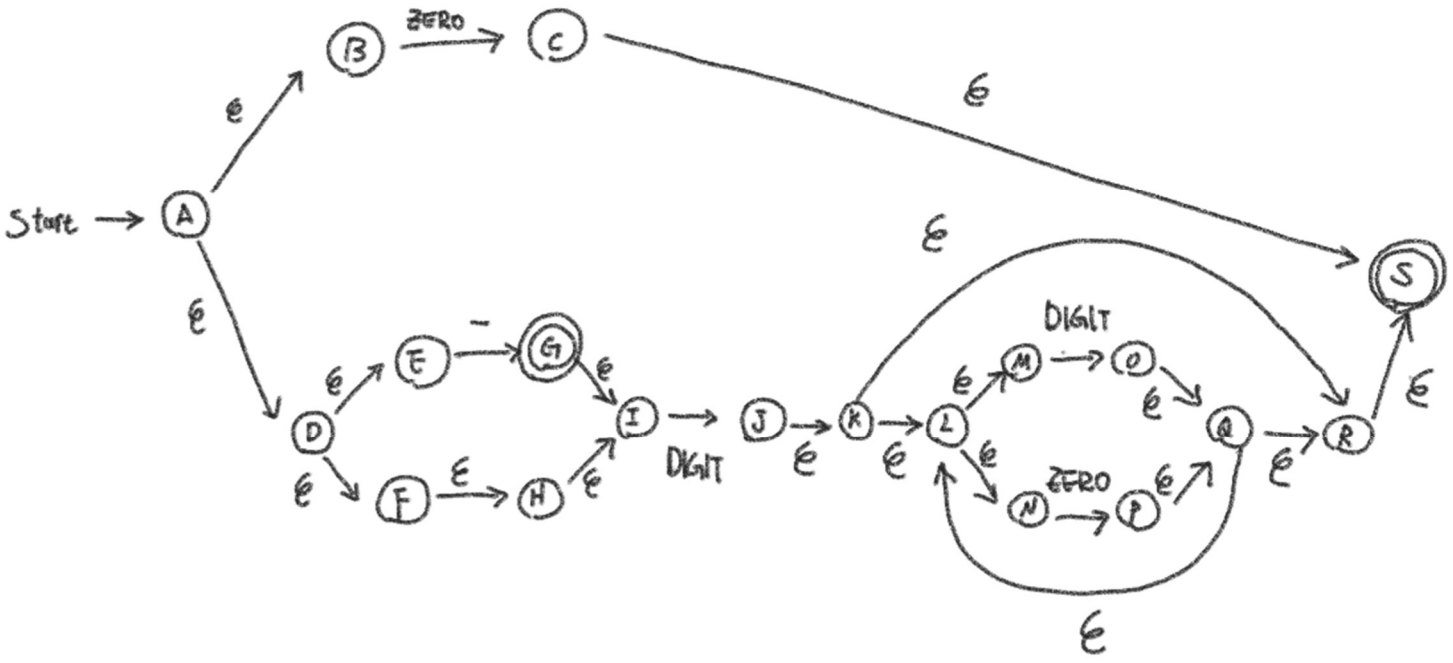
$T_3 = \varepsilon\text{-closure}(\delta(T_1, \text{letter})) = \{G, I, B\}$

$T_4 = \varepsilon\text{-closure}(\delta(T_1, \text{blank})) = \{H, I, B\}$

$T_5 = \varepsilon\text{-closure}(\delta(T_3, \text{"}")) = \{J\}$

	digit	letter	blank	"	other
T_0	ϕ	ϕ	ϕ	T_1	ϕ
T_1	T_2	T_3	T_4	ϕ	ϕ
T_2	T_2	T_3	T_4	T_5	ϕ
T_3	T_2	T_3	T_4	T_5	ϕ
T_4	T_2	T_3	T_4	T_5	ϕ
T_5	ϕ	ϕ	ϕ	ϕ	ϕ

[Signed Integer & operator '-']



INT = - | ZERO | (-|ε) DIGIT (ZERO | DIGIT)*

$T_0 = \varepsilon(A) = \{A, B, D, E, F, H, I\}$

$T_1 = \varepsilon(\delta(T_0, \text{zero})) = \varepsilon(C) = \{C, S\}$

$T_2 = \varepsilon(\delta(T_0, -)) = \varepsilon(G) = \{G, I\}$

$T_3 = \varepsilon(\delta(T_0, \text{DIGIT})) = \varepsilon(J) = \{J, K, L, M, N, R, S\}$

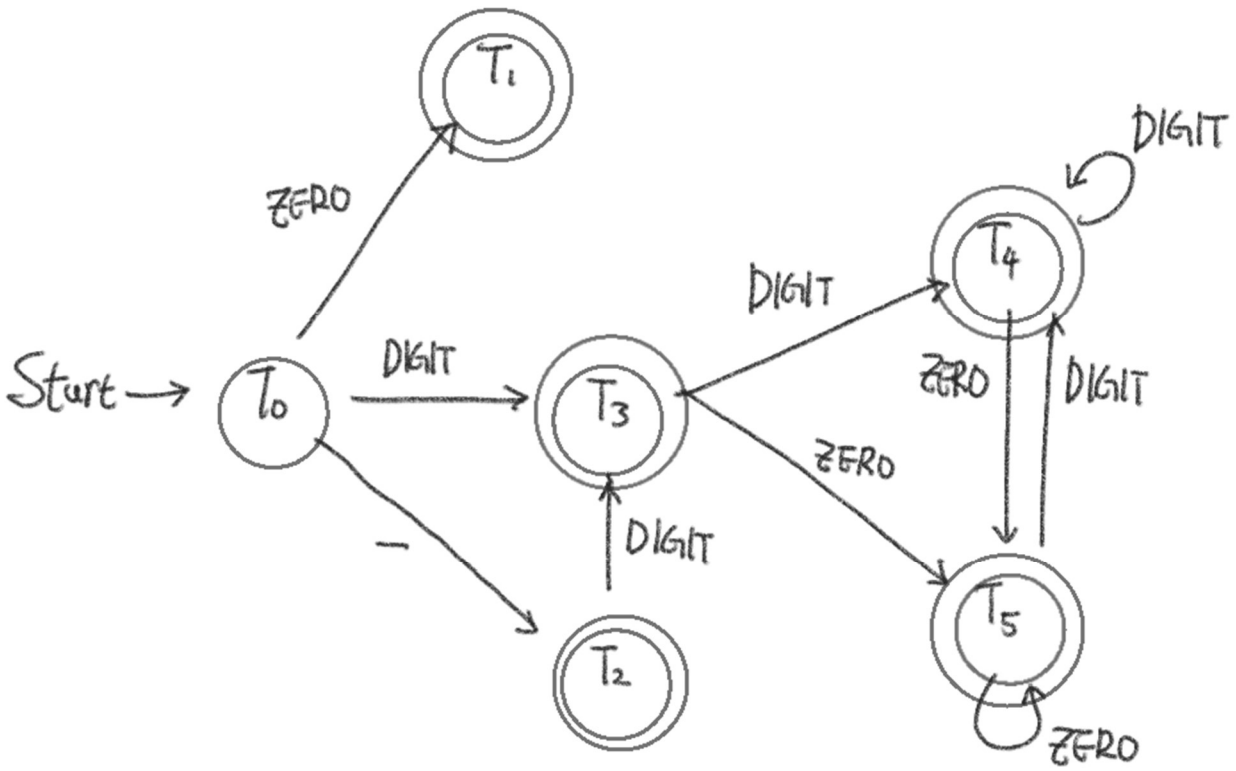
$T_4 = \varepsilon(\delta(T_3, \text{DIGIT})) = \varepsilon(O) = \{O, Q, R, S, L, M, N\}$

$T_5 = \varepsilon(\delta(T_3, \text{ZERO})) = \varepsilon(P) = \{P, Q, R, S, L, M, N\}$

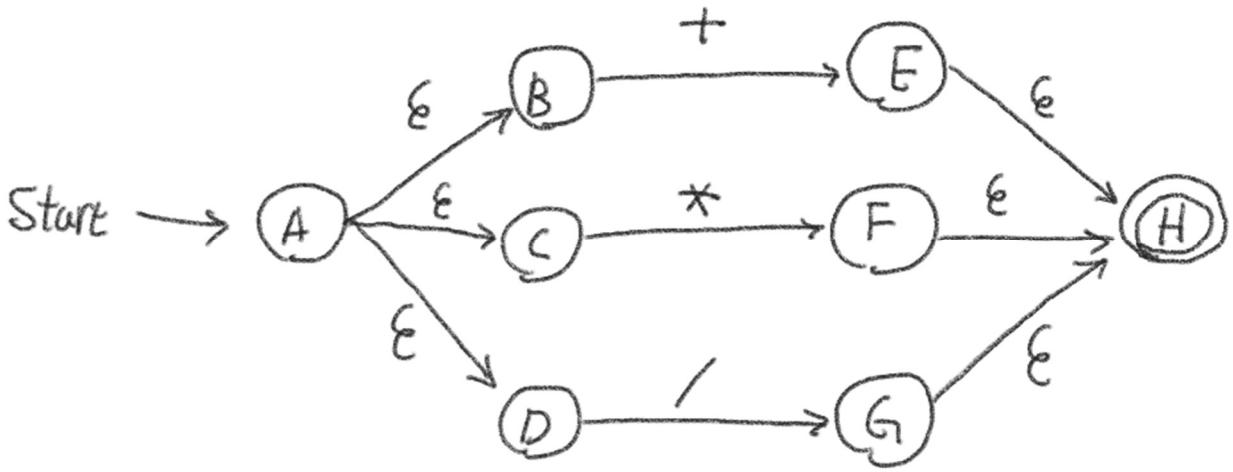
$\varepsilon(\delta(T_4, \text{DIGIT})) = T_4$, $\varepsilon(\delta(T_4, \text{ZERO})) = T_5$

$\varepsilon(\delta(T_5, \text{DIGIT})) = T_4$, $\varepsilon(\delta(T_5, \text{ZERO})) = T_5$

	ZERO	–	DIGIT	Other
T_0	T_1	T_2	T_3	$\emptyset$
T_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$	T_3	$\emptyset$
T_3	T_5	$\emptyset$	T_4	$\emptyset$
T_4	T_5	$\emptyset$	T_4	$\emptyset$
T_5	T_5	$\emptyset$	T_4	$\emptyset$



[Arithmetic_Op]



Arithmetic_Op = + | * | /

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

$T_1 = \varepsilon(\sigma(T_0, +)) = \varepsilon(E) = \{E, H\}$

$T_2 = \varepsilon(\sigma(T_0, *)) = \varepsilon(F) = \{F, H\}$

$T_3 = \varepsilon(\sigma(T_0, /)) = \varepsilon(G) = \{G, H\}$

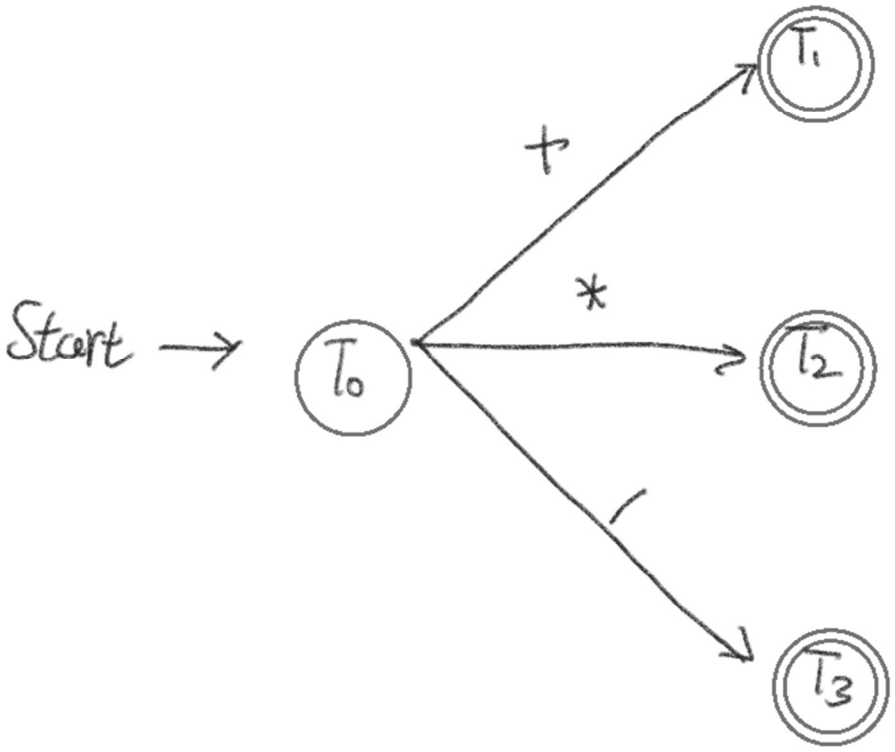
$\varepsilon(\sigma(T_n, +)) = \emptyset$

$\varepsilon(\sigma(T_n, *)) = \emptyset$

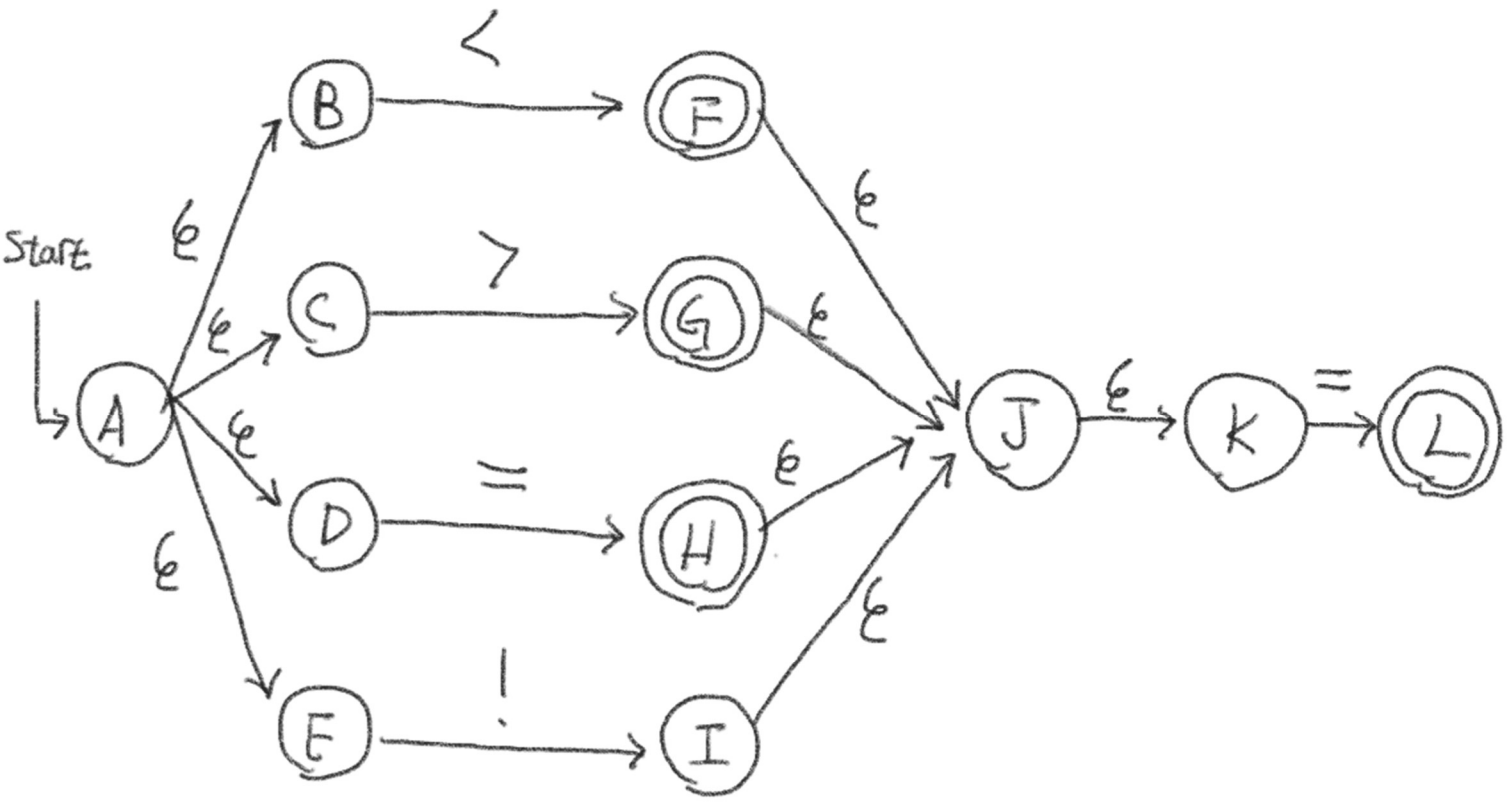
$\varepsilon(\sigma(T_n, /)) = \emptyset$

* $n = 1, 2, 3$

	+	*	/	other
T_0	T_1	T_2	T_3	$\emptyset$
T_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
T_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$



[Comparison_Op & Assign_op]



$$Op = != | (< | > | =) (= | \varepsilon)$$

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

$$T_1 = \varepsilon(\sigma(T_0, <)) = \varepsilon(F) = \{F, J, K\}$$

$$T_2 = \varepsilon(\sigma(T_0, >)) = \varepsilon(G) = \{G, J, K\}$$

$$T_3 = \varepsilon(\sigma(T_0, =)) = \varepsilon(H) = \{H, J, K\}$$

$$T_4 = \varepsilon(\sigma(T_0, !)) = \varepsilon(I) = \{I, J, K\}$$

$T_5 = \mathcal{E}(\mathcal{E}(T_n, =)) = \mathcal{E}(L) = \{L\}$

$* n = 1, 2, 3, 4$

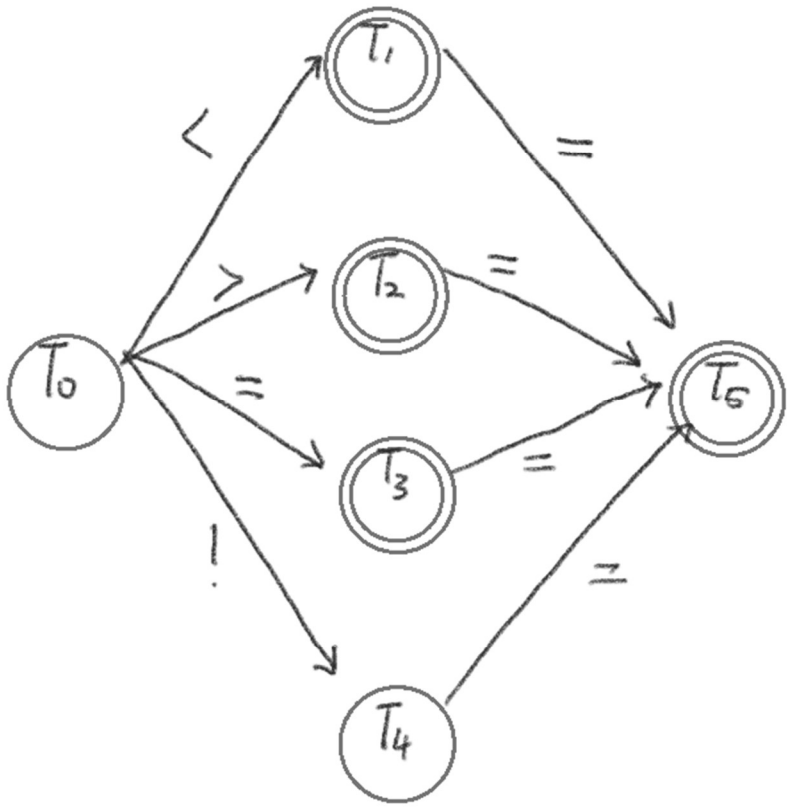
$\mathcal{E}(\mathcal{E}(T_n, <)) = \emptyset$

$\mathcal{E}(\mathcal{E}(T_n, >)) = \emptyset$

$\mathcal{E}(\mathcal{E}(T_n, !)) = \emptyset$

$* n = 1, 2, 3, 4, 5$

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



[Semi_Colon]

Semi-Colon = ;

$T_0 = \varepsilon(A) = \{A\}$

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

$\varepsilon(\delta(T_1, ;)) = \emptyset$

	;	Other
T_0	T_1	$\emptyset$
T_1	$\emptyset$	$\emptyset$

[Comma]

Comma = ,

$T_0 = \varepsilon(A) = \{A\}$

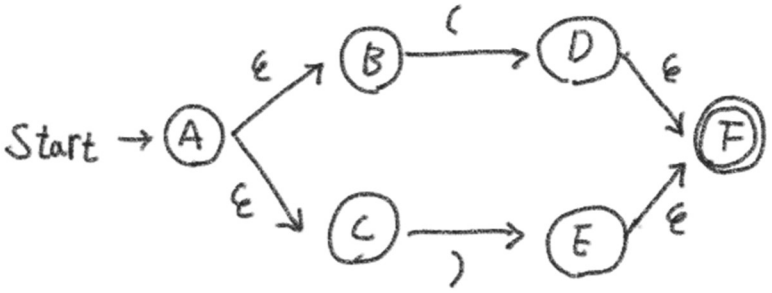
$T_1 = \varepsilon(\delta(T_0, ,)) = \varepsilon(B) = \{B\}$

$\varepsilon(\delta(T_1, ,)) = \emptyset$

	,	Other
T_0	T_1	$\emptyset$
T_1	$\emptyset$	$\emptyset$

[Parentheses]

Paren = (|)



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

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

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

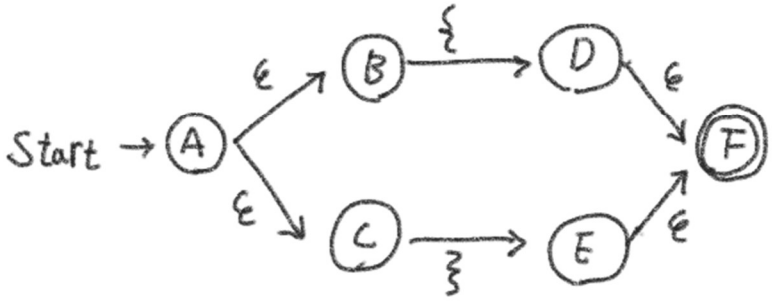
$\varepsilon(\delta(T_1, ()) = \varepsilon(\delta(T_1,))) = \emptyset$

$\varepsilon(\delta(T_2, ()) = \varepsilon(\delta(T_2,))) = \emptyset$

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

[brace]

brace = { | }



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

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

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

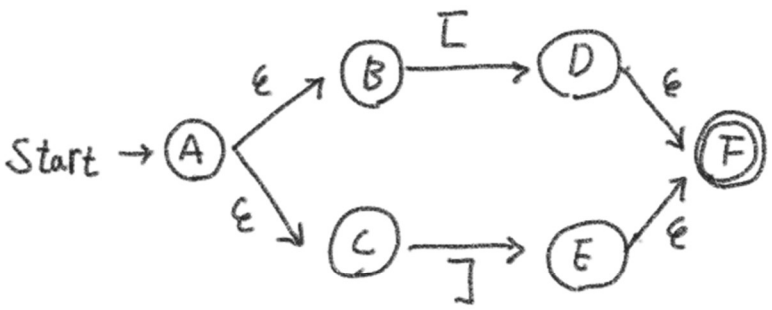
$\varepsilon(\delta(T_1, \{)) = \varepsilon(\delta(T_1, \})) = \emptyset$

$\varepsilon(\delta(T_2, \{)) = \varepsilon(\delta(T_2, \})) = \emptyset$

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

[bracket]

bracket = [|]



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

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

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

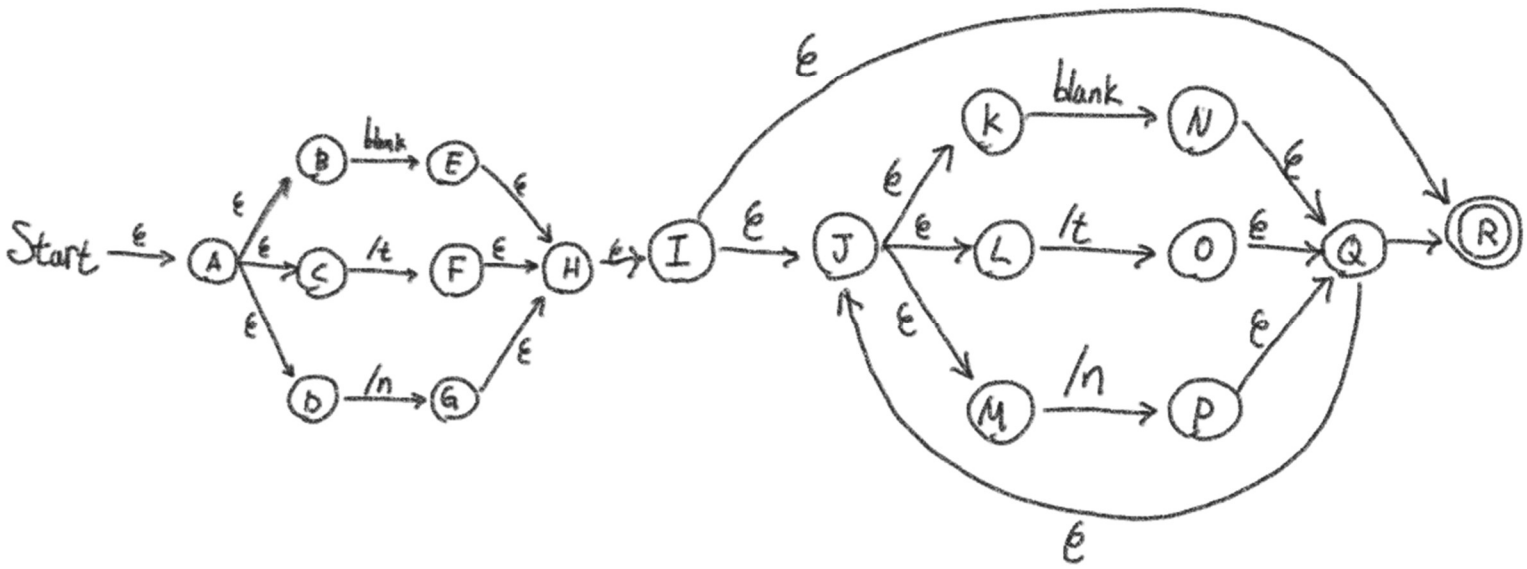
$\varepsilon(\delta(T_1, [)) = \varepsilon(\delta(T_1,])) = \emptyset$

$\varepsilon(\delta(T_2, [)) = \varepsilon(\delta(T_2,])) = \emptyset$

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

[White_Space]

White_Space = (blank | /t | /n) (blank | /t | /n)*



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

$T_1 = \varepsilon(G(T_0, \text{blank})) = \varepsilon(E) = \{E, H, I, J, k, L, M, R\}$

$T_2 = \varepsilon(G(T_0, /t)) = \varepsilon(F) = \{F, H, I, J, k, L, M, R\}$

$T_3 = \varepsilon(G(T_0, /n)) = \varepsilon(G) = \{G, H, I, J, k, L, M, R\}$

$T_4 = \varepsilon(G(T_1, \text{blank})) = \varepsilon(N) = \{N, Q, J, k, L, M, R\}$

$T_5 = \varepsilon(G(T_1, /t)) = \varepsilon(O) = \{O, Q, J, k, L, M, R\}$

$T_6 = \varepsilon(G(T_1, /n)) = \varepsilon(P) = \{P, Q, J, k, L, M, R\}$

$\epsilon(\delta(T_2, \text{blank})) = \epsilon(N) = T_4$, $\epsilon(\delta(T_3, \text{blank})) = \epsilon(N) = T_4$

$\epsilon(\delta(T_2, /t)) = \epsilon(N) = T_5$, $\epsilon(\delta(T_3, /t)) = \epsilon(N) = T_5$

$\epsilon(\delta(T_2, /n)) = \epsilon(N) = T_6$, $\epsilon(\delta(T_3, /n)) = \epsilon(N) = T_6$

* n - 4, 5, 6

$\epsilon(\delta(T_n, \text{blank})) = \epsilon(N) = T_4$

$\epsilon(\delta(T_n, /t)) = \epsilon(N) = T_5$

$\epsilon(\delta(T_n, /n)) = \epsilon(N) = T_6$

	blank	/t	/n	Other
T_0	T_1	T_2	T_3	$\emptyset$
T_1	T_4	T_5	T_6	$\emptyset$
T_2	T_4	T_5	T_6	$\emptyset$
T_3	T_4	T_5	T_6	$\emptyset$
T_4	T_4	T_5	T_6	$\emptyset$
T_5	T_4	T_5	T_6	$\emptyset$
T_6	T_4	T_5	T_6	$\emptyset$