

## 1. Define Tokens

Type : int | float | bool | String

Id : identifier

Keyword : if | else | while | for | return

A-O : + | - | \* | /

B-O : << | >> | && | ||

Assign O : =

C-O : < | > | == | != | <= | >=

terminating : ;

LBraket : {

RBraket : }

LParan : (

RParan : )

Separating : ,

whitespaces: tab | tab | blanks

## 2. Regular Expression

### 1) Type

• INT :  $(0|(-1\varepsilon))((\frac{\text{non-zero}}{\text{digit}})(\frac{\text{non-zero}}{\text{digit}})\text{digit}^*)$

• float : (INT).  $(0|\frac{\text{non-zero}}{\text{digit}})\text{digit}^*(\frac{\text{non-zero}}{\text{digit}})$

• String :  $(\"(\text{digit}|\text{letter}|\text{whitespace})^*\")$

- bool : (true | false)

2) Id

- Id : (Letter | \_)(Letter, Digit | \_)\*

3) keywords.

- keywords : (if | else | while | for | return)

4) A-D

- A-D : (+ | - | \* | /)

5) B-D

- B-D : (<< | >> | && | ||)

6) Assign O

- Assign O : (=)

7) C-D

- C-D : (<(=|<) | >(=|>) | (=|!:)= )

8) terminate

- terminate : ;

9) LBasket

- LBasket : ( ( ) )

10) RBasket

- RBasket : ( ? )

11) Lparen

- Lparen : ( ( )

12) Rparen

- Rparen : ( ) )

13) Separating

- Separating : ( , )

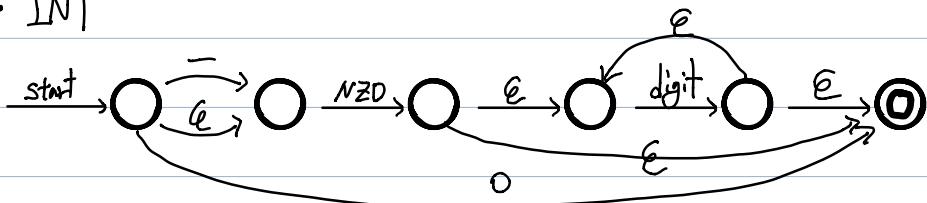
14) whitespaces

- whitespaces : ( wt | wh | blank ) \*

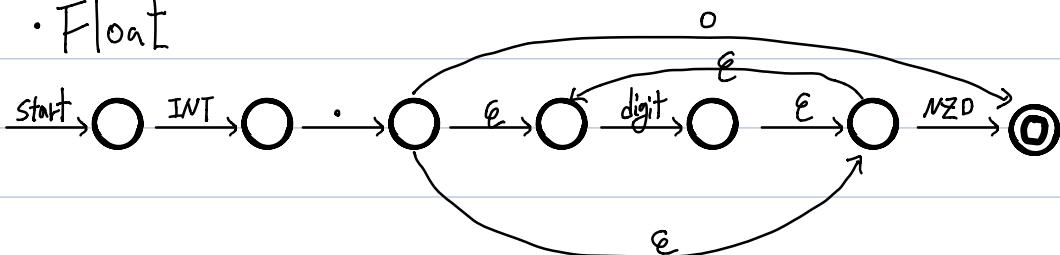
### 3. NFA

Non-zero digit = NZD

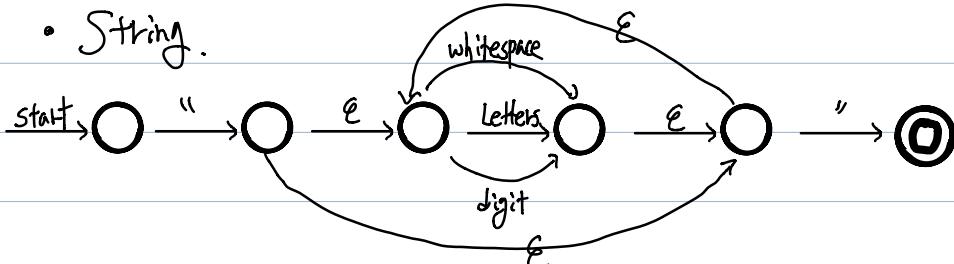
- INT



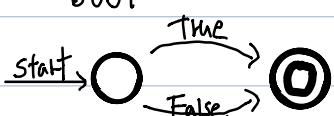
- Float



- String

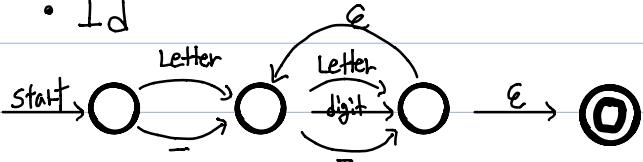


- bool



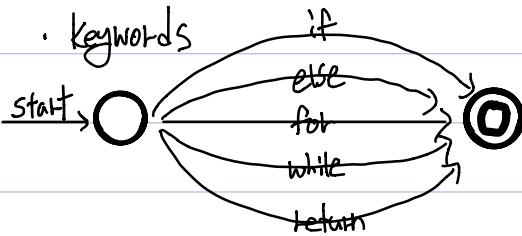
2) Id

• Id



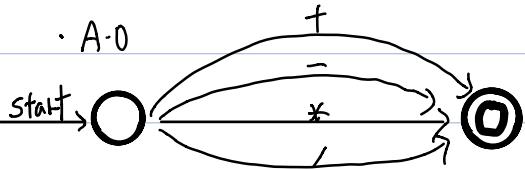
3) keywords

• keywords



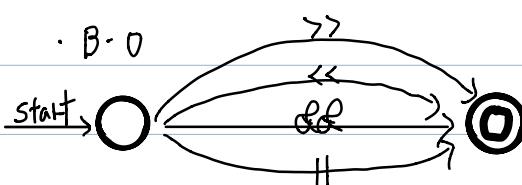
4) A·0

• A·0



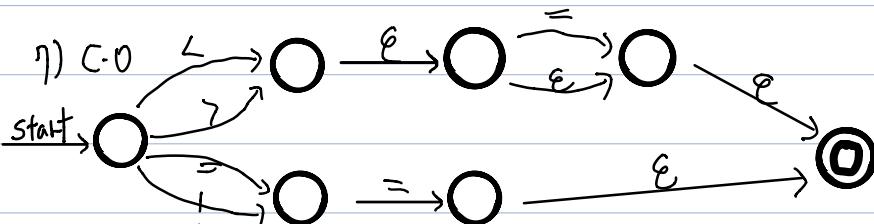
5) B·0

• B·0



6) Assign 0

• Assign 0



8) terminate



9) LBasket



10) RBasket



11) LParen



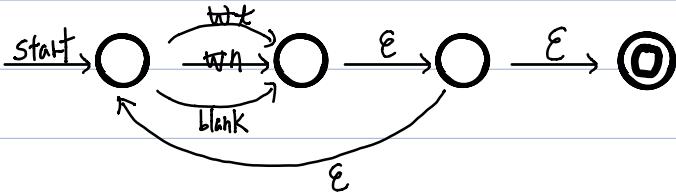
12) RParen



13) Separating



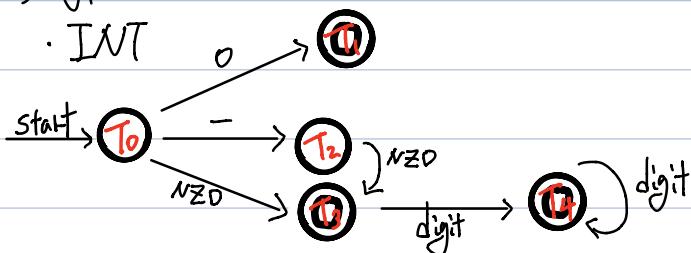
#### 14) whitespaces



#### 4. DFA

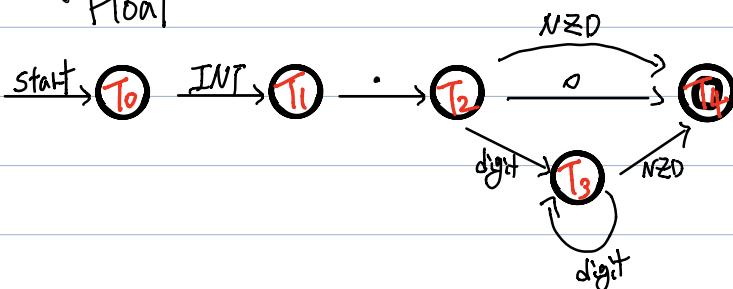
##### 1) Type

- INT



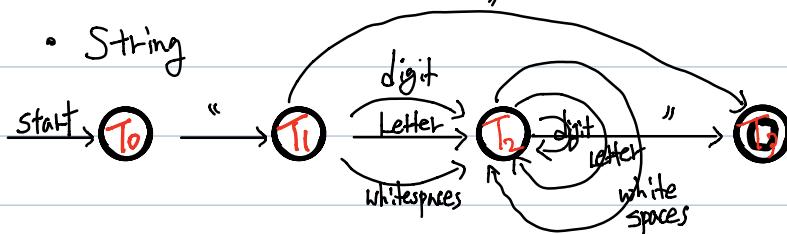
	0	-	NZD	digit
T <sub>0</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	X
T <sub>1</sub>	X	X	X	X
T <sub>2</sub>	X	X	T <sub>3</sub>	X
T <sub>3</sub>	X	X	X	T <sub>4</sub>
T <sub>4</sub>	X	X	X	T <sub>4</sub>

- Float



	INT	.	NZD	0	digit
T <sub>0</sub>	T <sub>1</sub>	X	X	X	X
T <sub>1</sub>	X	T <sub>2</sub>	X	X	X
T <sub>2</sub>	X	X	T <sub>4</sub>	T <sub>4</sub>	T <sub>3</sub>
T <sub>3</sub>	X	X	T <sub>4</sub>	X	T <sub>3</sub>
T <sub>4</sub>	X	X	X	X	X

- String



	"	digit	letter	white spaces
T <sub>0</sub>	T <sub>1</sub>	X	X	X
T <sub>1</sub>	T <sub>3</sub>	T <sub>2</sub>	T <sub>2</sub>	T <sub>2</sub>
T <sub>2</sub>	T <sub>3</sub>	T <sub>2</sub>	T <sub>2</sub>	T <sub>2</sub>
T <sub>3</sub>	X	X	X	X

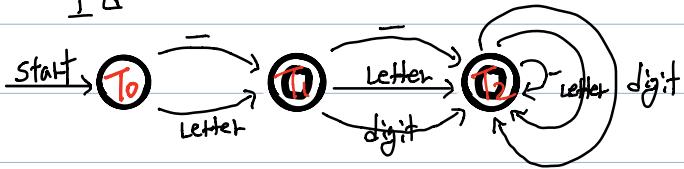
- bool



	True	False
T <sub>0</sub>	T <sub>1</sub>	X
T <sub>1</sub>	X	X

2) I.d

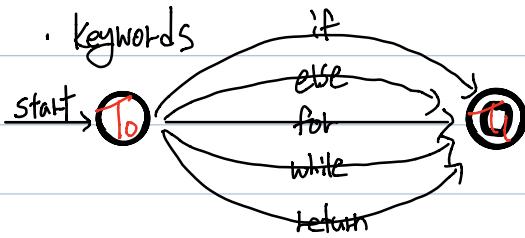
• I.d



	-	letter	digit
$T_0$	$T_1$	$T_1$	X
$T_1$	$T_2$	$T_2$	$T_2$
$T_2$	$T_3$	$T_3$	$T_3$

3) keywords

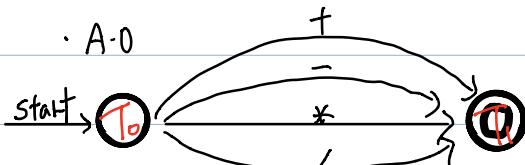
• keywords



	if	else	for	while	return
$T_0$	$T_1$	$T_1$	$T_1$	$T_1$	$T_1$
$T_1$	X	X	X	X	X

4) A.0

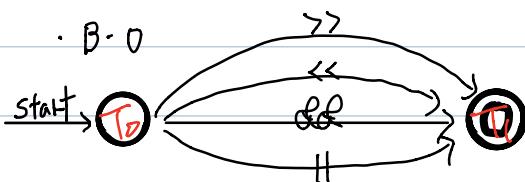
• A.0



	+	-	*	/
$T_0$	$T_1$	$T_1$	$T_1$	$T_1$
$T_1$	X	X	X	X

5) B.0

• B.0



	>>	<<	&&	
$T_0$	$T_1$	$T_1$	$T_1$	$T_1$
$T_1$	X	X	X	X

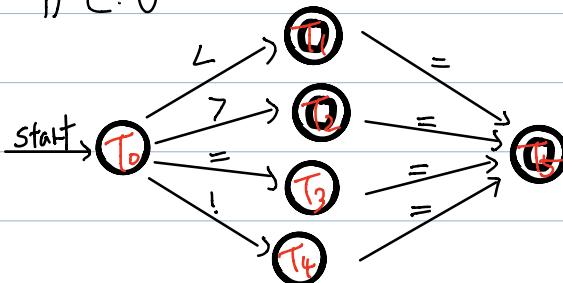
6) Assign 0

• Assign 0



	=
$T_0$	$T_1$
$T_1$	X

7) C.0



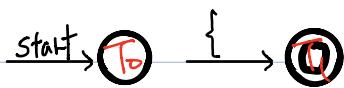
	<	>	=	!=
$T_0$	$T_1$	$T_2$	$T_3$	$T_4$
$T_1$	X	X	$T_5$	X
$T_2$	X	X	$T_5$	X
$T_3$	X	X	$T_5$	X
$T_4$	X	X	$T_5$	X
$T_5$	X	X	X	X

8) terminate



		j
	$T_0$	$T_1$
	$T_1$	x

9) L Basket



		{
	$T_0$	$T_1$
	$T_1$	x

10) R Basket



		}
	$T_0$	$T_1$
	$T_1$	x

11) L Paren



		(
	$T_0$	$T_1$
	$T_1$	x

12) R Paren



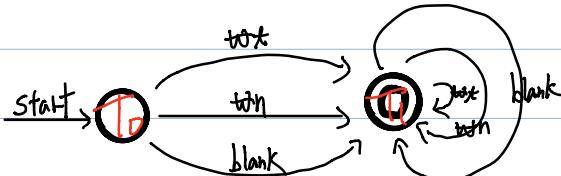
		)
	$T_0$	$T_1$
	$T_1$	x

13) Separating



		,
	$T_0$	$T_1$
	$T_1$	x

14) whitespaces.



	text	text	blank
	$T_0$	$T_1$	$T_1$
	$T_1$	$T_1$	$T_1$