Assignment 1 Report

Yizhou Sun 40056775

Lexical specifications:

id ::= [a-zA-Z][a-zA-Z0-9_]*

alphanum ::= [a-zA-Z0-9_]

integer ::= [1-9][0-9]*|0

float ::= ([1-9][0-9]*|0).([0-9]*[1-9]|0)(e(+|-)([1-9][0-9]*|0))?

fraction ::= .([0-9]*[1-9]|0)

letter ::= [a-zA-Z]

digit ::= [0-9]

nonzero ::= [1-9]

Operators, punctuations, and reserved words:

==	+	|	(	;	if	public	read
<>	-	&	)	,	then	private	write
<	*	!	{	.	else	func	return
>	/		}	:	integer	var	self
<=	=		[	->	float	struct	inherits
>=			]		void	while	let
							impl

coloncolon ::= :: (I include this because it's in the given test file, although it's not in the table above)

blockcmt ::= `/\*(.|[\r\n])*?\*/` (block comment)

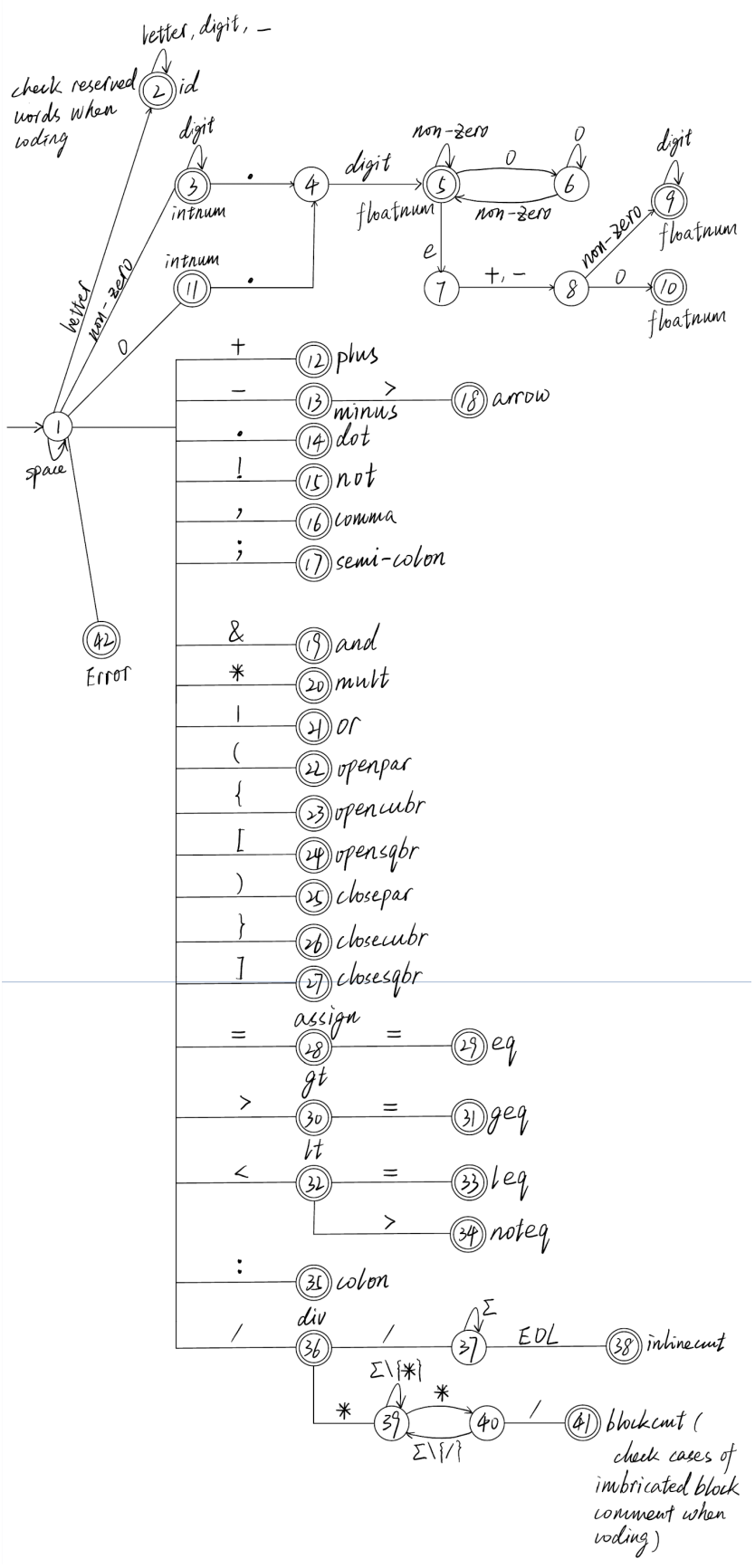
inlinecmt ::= `\/\/.*` (inline comment)

Finite state automaton:

Notation used in DFA in addition to the lexical specification:

Σ: set of all symbols in the lexical specification

EOL: End of Line (\n, \r, etc)


letter, digit, _
check reserved words when coding
id
digit
intnum
.
digit
floatnum
non-zero
0
non-zero
0
digit
floatnum
e
+, -
non-zero
0
floatnum
letter
non-zero
0
intnum
.
space
Error
+
plus
-
minus
>
arrow
.
dot
!
not
,
comma
;
semi-colon
&
and
*
mult
|
or
(
openpar
{
opencubr
[
opensqbr
)
closepar
}
closecubr
]
closesqbr
=
assign
=
eq
>
gt
=
geq
<
lt
=
leq
>
noteq
:
colon
/
div
/
Σ
EOL
inlinecmt
*
Σ\{*}
*
*
Σ\{/}
/
blockcmt (
check cases of imbricated block comment when coding)

Design:

The Table-Driven Scanner approach is used to analyze the state transitions. I created a state transition table and implemented using an array of State objects, each containing a HashMap<String, Integer> transition map, indicating the acceptable inputs for a state and the corresponding destination state ids for the inputs.

In addition to the State class, I also created the Token class and the TokenName class to represent both valid and invalid tokens.

Use of tools:

- Java project on Eclipse: I’m familiar with the language and the platform and have been using it since the first programming course.
- Notability note app for drawing DFA: I prefer drawing by hand as I could adjust the layout as I wish. It’s more flexible than using online tools for drawing in my opinion.
- Word for creating the state transition table: no particular reason, I could have used Excel.