LEXICAL ANALYZER FOR THE BIGADD LANGUAGE

PRE-ACCEPTANCES

a) Recognized characters are:

1. . (EndOfLine)
2. , (Seperator)
3. - (MinusSign)
4. [(OpenBlock)
5.] (CloseBlock)
6. { (LeftCurlyBracket)
7. } (RightCurlyBracket)
8. " (StringConstantStarterAndEnder)
9. _ (UnderScore)

and alphanumeric characters. (Locale Turkish characters are unrecognizable.)

b) Maximum size for specific tokens are:

IntConstant → 100
VariableName → 20
StringConstant → 1000

c) (“) and (”) are unrecognizable. String constant can start and end with (").

d) 054374, -0, -053 or 0054374 **does** throw error because of the first index being **0**.

e) - 5, - 5 or --5 is not a valid integer for the BigAdd language but this L.A. will not catch those errors. Becasuse, when parser finds the MinusSign, it will indicate that minus sign and integer have at least one blank between thus, there’s an error. If it was not - 5 but -5, then minus sign would not be a different token and this would indicate that there is no error. This logic applies to 3.14159, 3.0, 3 and .5 too.

ERROR MESSAGES

1. FileNotOpen

2. UnrecognizedCharacter

3. VariableNameTooLong

4. IntConstantTooBig

5. IntConstantCannotStartWithZero =

0 is valid.

05 is **not** valid.

-**0** is **not** valid.

-**0**5 is **not** valid.

6. StringConstantTooLong

7. LeftCurlyBracketMissing =

{ } is valid.

{ { } is valid.

{ { { { } is valid.

{ { { { } **}** is **not** valid.

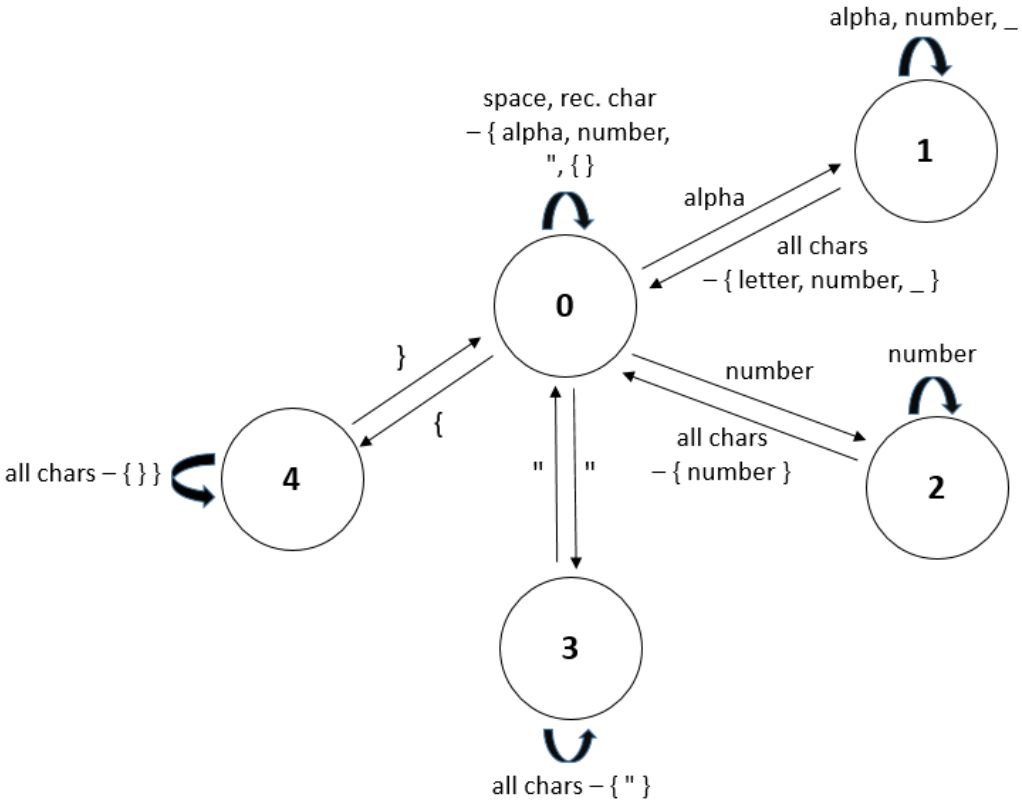
{ { { { } **}** } is **not** valid.

} is **not** valid.

8. CommentLeftOpen (RightCurlyBracketMissing)

9. StringConstantLeftOpen

VISUALIZATION OF THE STATE MACHINE DESIGN



State 0: Determining The Next State Number

- Unrecognized characters will be catched in this state.

State 1: Reading Identifier or Keyword

State 2: Reading Integer Constant

- Integer constant cannot start with 0.

State 3: Reading String Constant

State 4: Comment

NOTE: In State 1 or State 2, when the unwanted character gets read, curser will go back 1 char size. And this unwanted character will be read on State 0.

RUN EXAMPLE 1 (GIVEN EXAMPLE IN THE PROJECT1 PDF)

INPUT:

```
int size.
int sum.
move 5 to size.
loop size times {ignore me, I am a comment
[ out size, newline.
add size to sum.
]
out newline, "Sum:", sum.
```

OUTPUT:

```
Keyword int
Identifier size
EndOfLine
Keyword int
Identifier sum
EndOfLine
Keyword move
IntConstant 5
Keyword to
Identifier size
EndOfLine
Keyword loop
Identifier size
Keyword times
OpenBlock
Keyword out
Identifier size
Seperator
Keyword newline
EndOfLine
Keyword add
Identifier size
Keyword to
Identifier sum
EndOfLine
CloseBlock
Keyword out
Keyword newline
Seperator
StringConstant "Sum:"
Seperator
Identifier sum
EndOfLine
```

RUN EXAMPLE 2 (UNRECOGNIZED CHARACTER)

INPUT:

```
int size.*
int sum.
move 5 to size.
loop size times {ignore me, I am a comment}
[ out size, newline.
add size to sum.
]
out newline, "Sum:", sum.
```

OUTPUT:

```
Keyword int
Identifier size
EndOfLine
UnrecognizedCharacter *
```

RUN EXAMPLE 3 (COMMENT LEFT OPEN)

INPUT:

```
int size.
int sum.
move 5 to size.
loop size times {ignore me, I am a comment
[ out size, newline.
add size to sum.
]
out newline, "Sum:", sum.
```

OUTPUT:

```
Keyword int
Identifier size
EndOfLine
Keyword int
Identifier sum
EndOfLine
Keyword move
IntConstant 5
Keyword to
Identifier size
EndOfLine
Keyword loop
Identifier size
Keyword times
CommentLeftOpen
```

RUN EXAMPLE 4 (VARIABLE NAME TOO LONG)

INPUT:

```
int size.
int sum.
move 5 to size.
int size_sometimes_matter.
loop size times {ignore me, I am a comment
[ out size, newline.
add size to sum.
]
out newline, "Sum:", sum.
```

OUTPUT:

```
Keyword int
Identifier size
EndOfLine
Keyword int
Identifier sum
EndOfLine
Keyword move
IntConstant 5
Keyword to
Identifier size
EndOfLine
Keyword int
VariableNameTooLong
```

RUN EXAMPLE 5 (INT CONST. CANNOT START WITH ZERO)

INPUT:

```
intintint INT.
move id3_ntif_i3r.
- 5
loop loop loop times {{{{{}
"i want to walk in the open wind"
...
[[[out size, newline.
312.312
]
037 rocks.
```

OUTPUT:

```
Identifier intintint
Identifier INT
EndOfLine
Keyword move
Identifier id3_ntif_i3r
EndOfLine
MinusSign
IntConstant 5
Keyword loop
Keyword loop
Keyword loop
Keyword times
StringConstant "i want to walk in the open wind"
EndOfLine
EndOfLine
EndOfLine
OpenBlock
OpenBlock
OpenBlock
Keyword out
Identifier size
Seperator
Keyword newline
EndOfLine
IntConstant 312
EndOfLine
IntConstant 312
CloseBlock
IntConstantCannotStartWithZero
```