

CS 155 Problem Set 2

EASY Flex Analyzer Regex Table

Modified Token Classifications Table

To satisfy the updated EASY Language guidelines and to pass the test files, the Token Classifications Table from Problem Set 1 has been modified to the table below.

Token	Token Classification	Regex
'<string>'	String	'[^']*'
//<comment>EOL	Comment	\\/\\.*\$
' ', '\t', '\r', '\f'	Whitespace	[\t\r\f]*
'\n'	Newline	[\n\r\t\f]
ℓ, ⓵, ⓶, ⓷, mod, log_<base>	Math_Notation	[ℓ ⓵ ⓶ ⓷] mod log_ ([0-9]+ ([.] [0-9]*)? [.] [0-9]+)
<integer>, <float>	Number	-? ([0-9]+ ([.] [0-9]*)? [.] [0-9]+)
+, -, /, *, ^	Arithmetic_Op	[+ \- \/ * ^]
and, or, not	Logical_Op	and or not
<, <=, ==, !=, >, >=	Relational_Op	== <= >= != [<>]
=	Assignment	=
go, to, exit, if, then, else, case, endcase, while, do, endwhile, repeat, until, loop, forever, for, by, do, input, output, array, node, call, return, stop, end, procedure	Keyword	go to exit if then else case endcase while do endwhile repeat until loop forever for by do input output array node call return stop end procedure
[Open_Sqr_Bracket	\[
]	Close_Sqr_Bracket	\]
:	Colon	:
true, false	Boolean	true false
,	Comma	,
(Open_Paren	\(
)	Close_Paren	\)
<variable>, <label>, <name>, <procname>	Identifier	[_a-zA-Z] [_a-zA-Z0-9]*

The list of modifications and their justifications are the following:

1. Removed the Character token

With `String` tokens defined to be surrounded by single quotes, the distinction between `strings` and `characters` has been dropped, and `characters` are just treated as single-letter strings.

2. Added the Newline token

There was a realization that some EASY structures are distinguished from the next structure with an in between-ing newline. Lists for example, the last element of a `list` (which could be `strings` or `identifiers`) is distinguished from a next line's `string` or `identifier` token if we read and accept `Newline` tokens.

3. Updated Token Regexes

- a. **String** - Now surrounded by single quotes, also modified to not include nested single quotes within it, so it breaks immediately upon reading the second single quote.
- b. **Comments** - Now prefixed with `//` instead of `▷`
- c. **Assignment** - Now uses `=` instead of `←`
- d. **Arithmetic Operations** - Now uses `*` instead of `×`
- e. **Relational Operations** - Now uses `<=`, `==`, `!=`, `>=` instead of `≤`, `=`, `≠`, `≥`

4. Fixed Test Files

- a. Replaced instances of `=` being used as a relational operation with `==`
- b. In `reserved_test.easy`, fixed `until` to `until`