

Documentation of Lexical Analyzer of CBaby Language

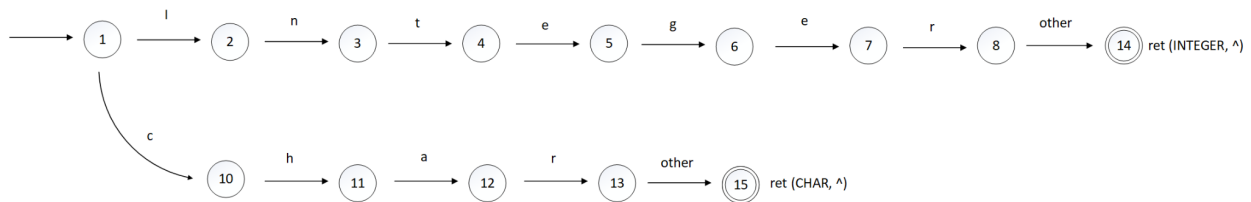
This documentation contains regular definitions and their transition diagrams (DFA) of basic features of the CBaby language.

1. Basic Type of Data: Integer, char

Regular Definition

TD -> Integer | char

Transition Diagram

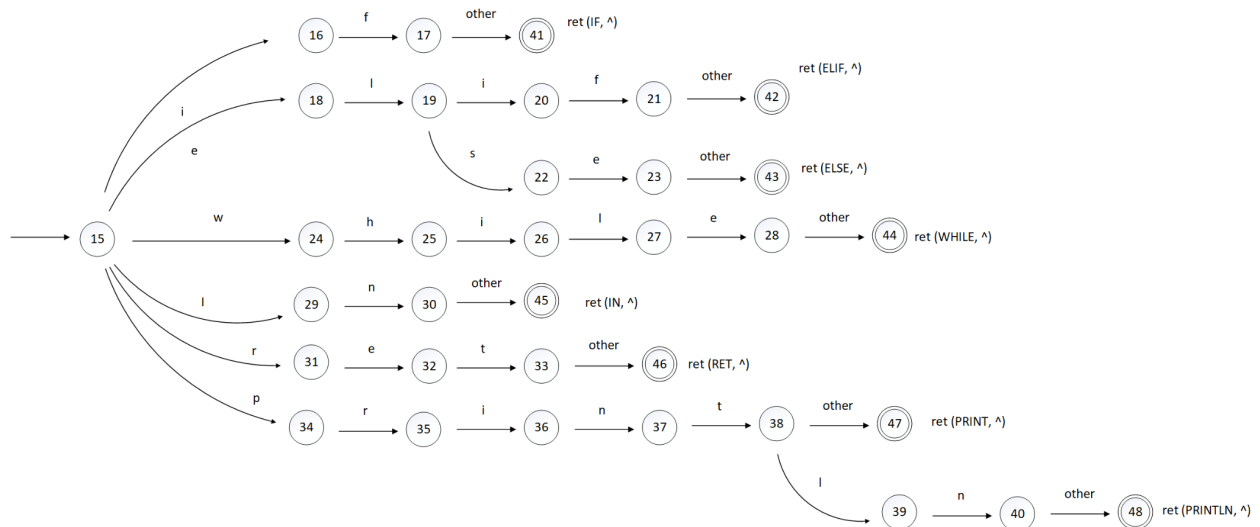


2. Keywords: if, elif, else, while, In, func, print, println, ret

Regular Definition

KW -> func | if | elif | else | while | In | ret | ((print)(^ | In))

Transition Diagram

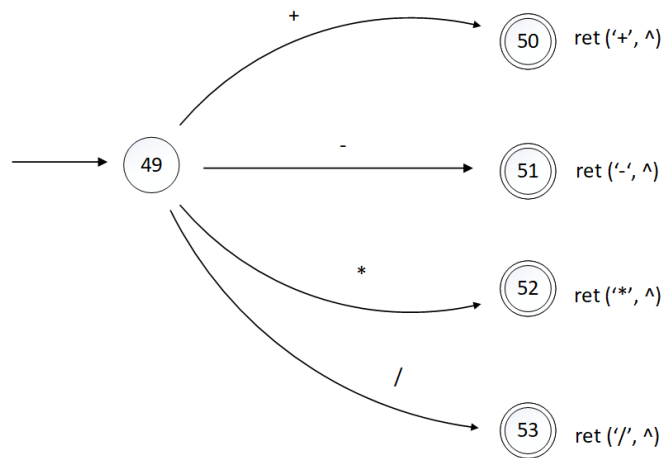


3. Arithmetic Operators: +, -, *, /

Regular Definition

ArOp -> + | - | * | /

Transition Diagram

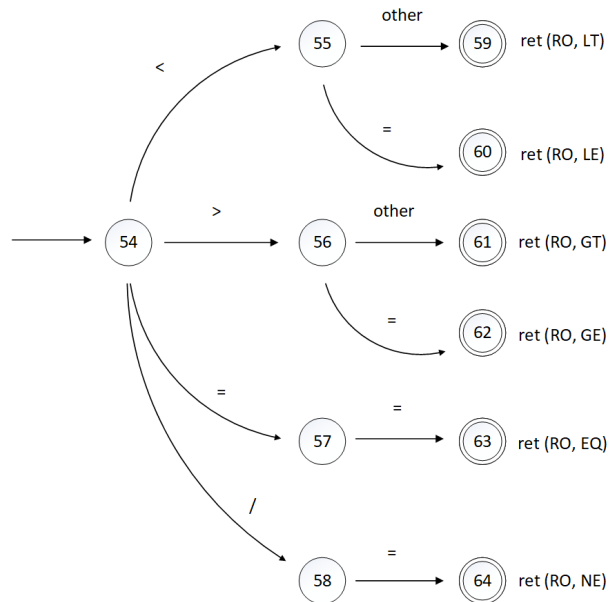


4. Relational Operators: <, <=, >, >=, ==, /=

Regular Definition

ReOp -> (< | (^ | =)) | (> | (^ | =)) | (==) | (/=)

Transition Diagram



5. Single line/Multi line Comments: /* enclosed comment */

Regular Definition

letter -> a | b | c | ... | z | A | B | C | ... | Z

digit -> 0 | 1 | 2 | ... | 9

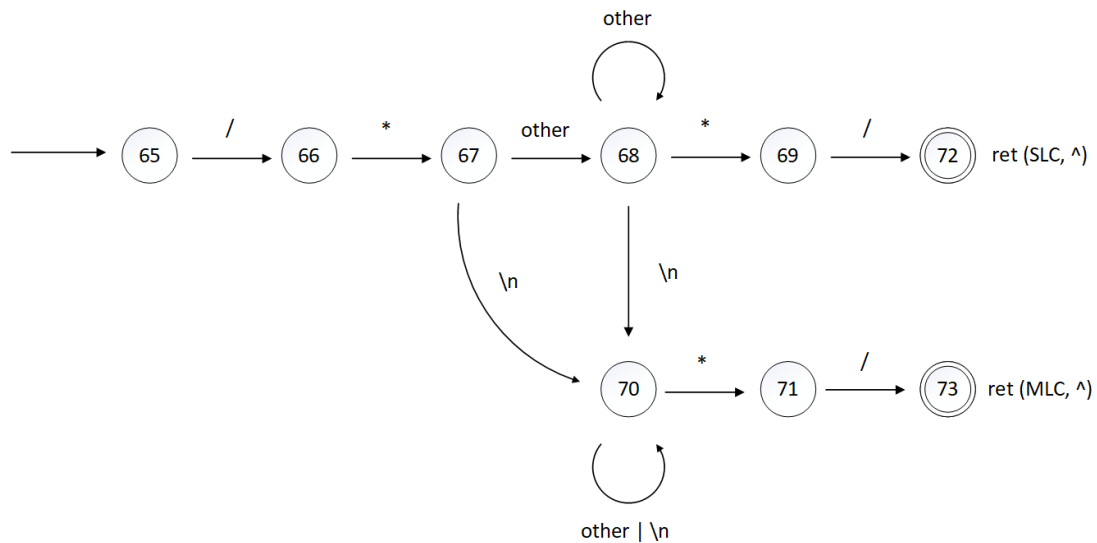
spC -> ~ | ` | ! | @ | \$ | % | & | * | (|) | { | } | [|] | + | = | _ | - | \ | / | < | > | . | , | " | ' | space | : | ;
| ? | | |

newLine -> \n

SLC -> /* (letter | digit | spC | ^) * */

MLC -> /* (letter | digit | spC | newLine | ^) * */

Transition Diagram



6. Identifier: a letter followed by any number of letters or digits

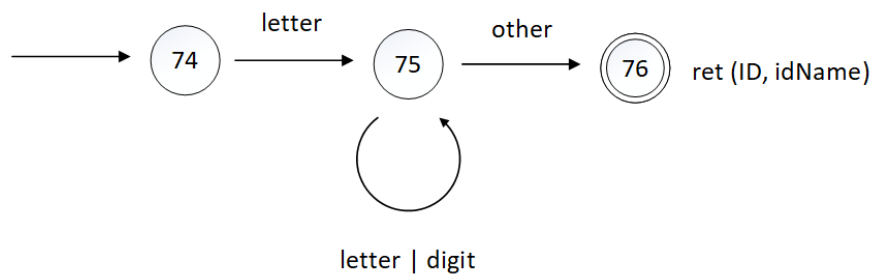
Regular Definition

letter -> a | b | c | ... | z | A | B | C | ... | Z

digit -> 0 | 1 | 2 | ... | 9

ID -> letter (letter | digit)*

Transition Diagram



(where idName is the name of the given Identifier)

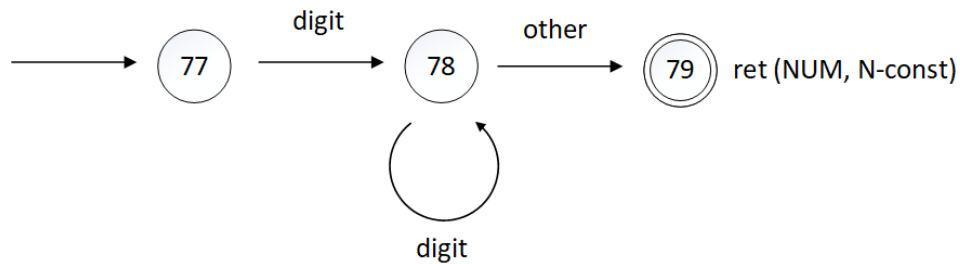
7. Numeric Constant: only integers

Regular Definition

digit -> 0 | 1 | 2 | ... | 9

NC -> digit (digit)*

Transition Diagram



(where N-const is the given integer value)

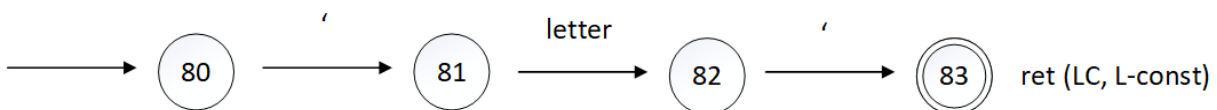
8. Literal Constant: a letter enclosed in single quotes

Regular Definition

letter -> a | b | c | ... | z | A | B | C | ... | Z

LC -> 'letter'

Transition Diagram



(where L-const is the given literal constant)

9. Strings: only used in print statements

Regular Definition

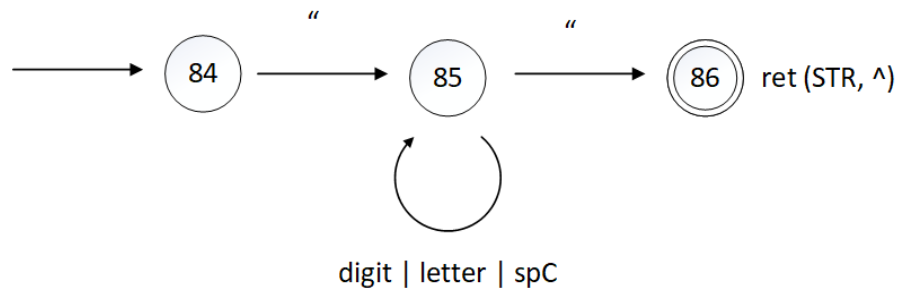
letter -> a | b | c | ... | z | A | B | C | ... | Z

digit -> 0 | 1 | 2 | ... | 9

spC -> ~ | ` | ! | @ | \$ | % | & | * | (|) | { | } | [|] | + | = | _ | - | \ | / | < | > | . | , | " | ' | space | : | ;
| ? | | |

STR -> " (letter | digit | spC | ^) * "

Transition Diagram

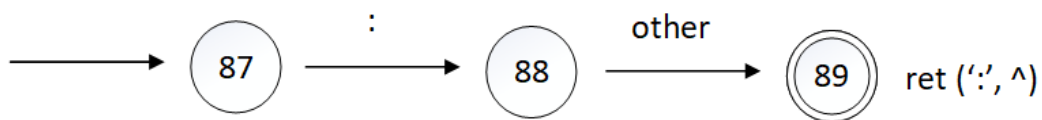


10. Variable Declaration Operator: :

Regular Definition

VDO -> :

Transition Diagram

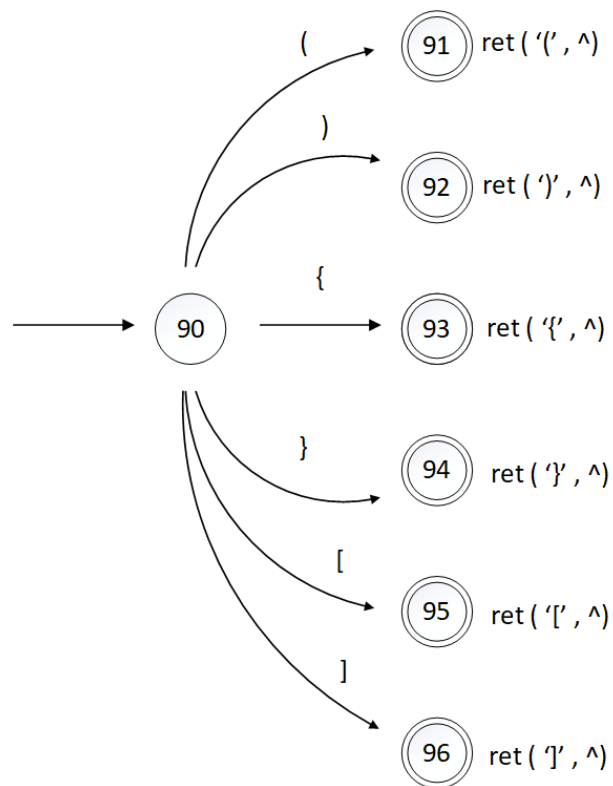


11. Parenthesis, Braces, Square Brackets: (), {}, []

Regular Definition

PBS -> (|) | { | } | [|]

Transition Diagram

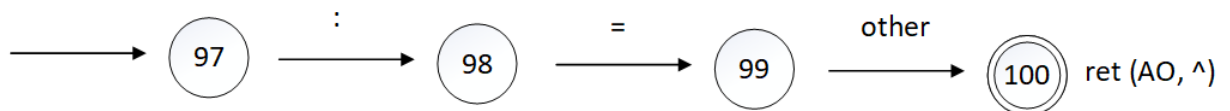


12. Assignment Operator: :=

Regular Definition

AO -> :=

Transition Diagram

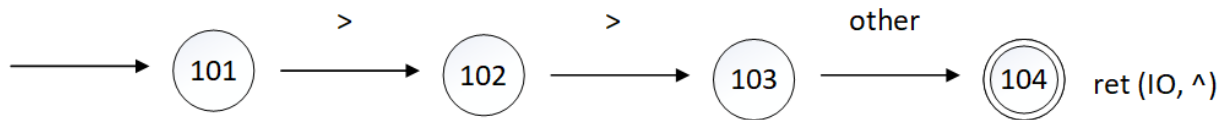


13. Input Operator: >>

Regular Definition

IO -> >>

Transition Diagram



14. Semicolon, Colon, Comma: ; , : , ,

Regular Definition

SCC -> ; | : | ,

Transition Diagram

