
SAM

Nitish Kumar Dwivedi (2018201068)

Nilabja Bhattacharya (2018201036)

OVERVIEW

- SAM is a programming language which is derived from C.
- Language supports 3 elementary data types. (int, bool, char).
- It also supports single and two-dimensional arrays of the same. All arrays are one-dimensional and have a compile-time indexed size. Arrays are indexed from 0 to N – 1, by the usual bracket notation where N > 0 is the size of the array.
- Operators supported : and, or, not
- Control Statements : if, else, for, while, ternary operator
- Function : call by value, recursion
- I/O routines

RESERVED KEYWORDS

int, char, if, else, break, for, while, main, break, continue, void, bool, true, false

CONTEXT FREE LANGUAGE

PROGRAM -> STATEMENTS

STATEMENTS -> STATEMENT STATEMENTS | STATEMENT

STATEMENT -> VAR_DECLARATION | FXN_DECLARATION

VAR_DECLARATION -> TYPE_DEF VAR_LIST VAR_DECLARATION; | E

TYPE_DEF -> TYPE | TYPE [NUM_CONST] | TYPE[NUM_CONST][NUM_CONST]

VAR_LIST -> VAR_DEC_ID | VAR_DEC_ID = EXPR

TYPE -> int | char | bool | signed int | unsigned int

FXN_DECLARATION -> TYPE ID (FUNC_ARGS) STATEMENT_BLOCK

FUNC_ARGS -> TYPE_DEF ID, FUNCTION_ARGS | TYPE_DEF ID

STATEMENT_BLOCK -> { CODE_LINE STATEMENT_BLOCK } | { CODE_LINE }

CODE_LINE -> VAR_DECLARATION ; | ITERATION_STATEMENT | CONTROL_STATEMENT ; |
RETURN_STATEMENT ; | ASSIGNMENT_STATEMENT ; | SIMPLE_STATEMENT ; | E

ITERATION_STATEMENT -> ITERATION_STATEMENT -> for (EXPR; EXPR; EXPR)
STATEMENT_BLOCK | while (EXPR) STATEMENT_BLOCK

ASSIGNMENT_STATEMENT -> ID = SIMPLE_STATEMENT | ID = SIMPLE_STATEMENT,
ASSIGNMENT_STATEMENT;

SIMPLE_STATEMENT -> EXPR | FXN_CALL

CONTROL_STATEMENT -> if (EXPR) BLOCK_STATEMENT | if (EXPR) BLOCK_STATEMENT else
BLOCK_STATEMENT

FXN_CALL -> ID (PARAMS_LIST)

RETURN_STATEMENT -> return ID | return CONST | return

PARAMS_LIST -> SIMPLE_STATEMENT, PARAMS_LIST | SIMPLE_STATEMENT

EXPR -> CONST | EXPR BINOP EXPR | (EXPR) | UNARYOP EXPR | EXPR UNARYOP |
TERNARY_EXPR

TERNARY_EXPR -> EXPR ? EXPR : EXPR

CONST -> NUM_CONST | CHAR_CONST

MICRO-SYNTAX

BINOP -> + | - | * | / | AND | OR | ^ | < | > | == | !=

UNARYOP -> ++ | -- | ! | -

ID -> ALPHA [ALPHA | NUM_CONSTANT]*

ALPHA -> [A-Z | a-z] +

CHAR_CONST -> ALPHA

NUM_CONSTANT -> [0-9] +

SEMANTICS CHECKS:

- 1) No two data types for a single variable.
- 2) No use of variable or assignment before the declaration
- 3) Return type of the function in function declaration must be same what it is returning.
- 4) The same variable should not be declared twice in the same scope.
- 5) Array size which is going to be used in the declaration must have a proper value before being used.