Three-Address Code Interpreter Documentation

Ahmet Emre Boyaci

The interpreter design consists of a run( ) function that acts as the spine of the interpreter, and the functions defined to accomplish the given tasks:

-mov( ):

Moves variables and their types into the “variables” dictionary.
-If there are more than one “dst”, “src1” or “src2”s, it exits with an error code of 5.

-add( ):

Adds two inputs from “src1” and “src2”, and moves the result to “dst”.
-If there are more than one “dst”, “src1” or “src2”s, it exits with an error code of 5.

-sub( ):

Subtracts the input “src2” from the input “src1”, and moves the result to “dst”.
-If there are more than one “dst”, “src1” or “src2”s, it exits with an error code of 5.

-mul( ):

Multiplies two inputs from “src1” and “src2”, and moves the result to “dst”.
-If there are more than one “dst”, “src1” or “src2”s, it exits with an error code of 5.

-div( ):

Divides the input from “src1” by the input from “src2”, and moves the result to “dst”.
-If there are more than one “dst”, “src1” or “src2”s, it exits with an error code of 5.
-If user tries to divide a number with zero, it throws a ZeroDivisionError and exits with the error code 12.

-read_int( ):

Reads an input from input( ) function of Python and moves the result to “dst”.
-If there are more than one “dst”s, it exits with an error code of 5.
-If the read value is not an integer, it throws a ValueError exception and exits with an error code of 14.

-print_( ):

Prints the input from “src1” with the print( ) function of Python.
-If there are more than one “dst”s, it exits with an error code of 5.

-jump( ):

Jumps to the label that is stored in “dst”.
-If there are more than one “dst”s, it exits with an error code of 5.

-jumpifeq( ):

Jumps to the label that is stored in “dst” if the values stored in “src1” and “src2” are equal.
-If there are more than one “dst”s, it exits with an error code of 5.

-jumpifgr( ):

Jumps to the label that is stored in “dst” if the value stored in “src1” is greater than the value in “src2”.

-If there are more than one “dst”s, it exits with an error code of 5.

-call( ):

Pushes the program counter into callstack[ ] array and jumps to the label stored in “dst”.
-If there are more than one “dst”s, it exits with an error code of 5.

-return( ):

Pops the last value from callstack[ ] array and moves it to program counter, then jumps to it.
-When there are no values to pop, it exits with an error code of 15.

-push( ):

Pushes the value in “src1” to the stack[ ] array.

-pop( ):

Pops a value from stack[ ] array and stores it in “dst”.
-When there are no values to pop, it exits with an error code of 15.
-If there are more than one “dst”s, it exits with an error code of 5.

-readstr( ):

Reads a string from Python’s input( ) function into str variable.

-concat( ):

Concatenates “src1” and “src2”, then assigns the result to “dst”

-strint( ):

Converts a string input into an integer.

-intstr( ):

Converts an integer input into a string.

-len( ):

Gives the length of a string input.

-getat( ):

Assigns the one-character string at index i of string src into dst (string).

-The program itself exits with an error code of 3 if the XML input is invalid, or if it can’t be parsed.