THE ACP PROGRAMMING LANGUAGE

Reference manual

**Authors:**

Avijith

Chinmay

Priyanka vats

Aditya Jagannathan

Team 6 | Arizona State University | CST 502 | Prof. John Femiani | 2015

Contents

1. Introduction…………………………………………
2. Tools used in ACP…………………………………
3. Compiling and running ACP…………………..
4. ACP type system…………………………………..
5. Declarations and Arithmetic……….…………
6. Scopes……………………………………….…………
7. If-else statement…………………………..……….
8. While loop……………………………………..……..
9. Functions………………………………………..……
10. Sample program………………………………..….
11. References……………………………………………
12. Introduction

The goal of ACP is to create a minimalistic language with basic arithmetic, Boolean and stack functionalities along with function implementations. Creating the language will help the team understand the design principles of programming language and also the implementation issues in designing one.

//The ACP compiler is written in Java.

1. Tools used in ACP

ANTLR - From a formal language description (ACP grammar), ANTLR was used to build a parser for that language that can automatically build parse trees, which are data structures representing how a grammar matches the input

//tools used for lexer

//any other tools used

1. Compiling and running ACP

// Language in which compiler runs

// Tools used for compiling

// Bash code for compiling

//Runtime environment

//Bash code for running

1. ACP type system

ACP supports three types:

* int – for signed integer types
* boolean – for assigning a true of false value to a variable
* stack – creating and basic operations of a stack

code snippet:

int term1 = 6;

boolean term2 = YES;

stack term3;

term3.push(8);

term3.pop();

ACP operators (complete list)

* +
* -
* \*
* /
* =
* ==
* >
* <
* >=
* <=
* !=

Parenthesis

* <<
* >>

1. Declarations and Arithmetic

When you’re defining a variable in ACP the variable has to have a value assigned to it during definition.

int x; // ERROR

int x = 7; // Correct ACP declaration

ACP supports basic arithmetic operations: Multiplication/Division, Addition/ Subtraction in that order of precedence.

Code snippet:

Function MDAS ()

<<

int x1 = 6;

int x2 = 3;

int a = x1 + x2;

int s = x1 – x2;

int m = x1\*x2;

int d = x1/x2;

print [a];

print [s];

print [m];

print [d];

>>

1. Scopes

ACP is a block-structured language where a ‘<<’ introduces a new scope. Scopes are nested when a new variable is declared in an inner scope.

//Is this right?

//Code snippet to show scope

7. If-else statement

ACP supports the use of if and else statements. Also, the if and else statements can be nested.

The if statement in ACP can be used with or without the else statement

Code snippet:

if ( flag == YES)

<< int x = a\*b; return x; >>

if (flag == NO)

<< int x = a\*b; return x; >>

else

<< return y; >>

if (flag == YES)

<<

if( x > 0)

<< return x; >>

else

<< return y; >>

>>

8. While loop

ACP supports while loop just like C does.

Sample code snippet to demonstrate while loop is given below.

while ( a<=5)

<<

print [a];

a=a+1;

>>

9. Functions

Functions in ACP can be used with function parameters or without them and also it can be just a block of code. It always begins with the keyword ‘function’ .

A regular block of code in ACP can be one or more of the following:

* If statement
* While statement
* Call
* Return statement
* Print statement
* Stack declaration
* Stack operations

A sample code snippet to show functions in ACP:

function example ( int x, int y, boolean flag)

<<

if(flag == YES)

<< return x\*y; >>

else

<< return x/y; >>

>>

10. Sample program

An ACP sample program to demonstrative recursion

function main()

<<

print[This is a sample program to find the factorial of 5];

int a =5;

int ans = factorial(5,1);

print[ans];

>>

function factorial(int fact, int ans)

<<

if( fact >= 1)

<<

ans = ans \* fact;

fact = fact -1;

factorial( fact, ans);

>>

else

<<

return fact;

>>

>>

This is a sample program to find the factorial of 5

120

11. References

//Add all references used.

<http://gnuu.org/2009/09/18/writing-your-own-toy-compiler/>

<http://stackoverflow.com/questions/21534316/is-there-a-simple-example-of-using-antlr4-to-create-an-ast-from-java-source-code>

<http://web.mit.edu/dmaze/school/6.824/antlr-2.7.0/doc/sor.html>

<https://vimeo.com/groups/29150/videos/8137747>