CS420: Compiler Design

Fall, 2019

Optional Feature 2 : Recursive Function Call

- **Recursive Function Call**

  Students who choose this option should implement recursive function call to operate correctly by the criteria below.

  - Statements while the function call must be executed in a correct order
  - Scopes of variables in each call must be correct in their function
  - Overall semantic values and operations must be correct

- **Expected Result for the interpreter implementation**

  The implemented interpreter is expected to operate as an example below

Sample code for recursive function call	Interpreter input commands and results

1 int sum(int num)
2 {
3 if(num > 1)
4 {
5 result = num + sum(num-1);
6 }
7
8 printf("%d added\n", num);
9 return num;
10 }
11
12
13 int main(void)
14 {
15 int result;
16 result = sum(8);
17 printf("\nTotal : %d\n", result);
18 }
	(In this example, the interpreter starts at the top of main function, line 14)
	
	>> **next** 10
	1 added
	2 added
	3 added
	4 added
	5 added
	6 added
	7 added
	8 added
	
	Total : 45
	End of program
	>>

Requirement Specification

Non-functional	
Recursive function call	The interpreter must be able to parse and execute recursive function call correctly as criteria below. 1. Correct statements while the recursive function call must executed in a correct order 2. Scopes of variables in each call must be correct in their function 3. Overall semantic values and operations must be correct Mandatory scope on calling pattern is A(A(A(A(...)))) and A(B(A(B(...)))). The maximum depth of recursive call stack is 1024.