CS420: Compiler Design

Fall, 2019

Optional Feature 3 : Code Generation

- **Code generation**

  Student who choose this option should implement code generation feature in the program. The real assembly must be generated before interpretation.

  1. Format

  - Data: Static variables in the assembly code must contain their allocated addresses. Initial value must be in the data section for literals.
  - Instructions: The format of the generated instruction is quite flexible. Assembly code must consist of operation and operands, basically, *opcode op1 op2 [op3]* type is acceptable. If you want to implement in other format, contact TA.

  2. Architecture

  The implemented code generation feature will operate in a certain architecture. Instruction Set Architecture (ISA), bit width of the processor such as 8 bit, 32 bit or 64bit, addressing and other details are required to be described in the first internal structure document for TA's feedback.

  3. Optimization level

  For the generated assembly code, optimization level is not restricted. Code optimization is out of scope in this option, but you may be adapt some optimizations for your convenience.

- **Expected result for the code generation**

  The implemented feature is expected to export an assembly code as an example below

Example input code	Exported assembly code

1 int main(void)
2 {
3 int scores[3], sum;
4
5 sum = 0;
6 score[0] = 80;
7 score[1] = 75;
8 score[2] = 100;
9
10 for(i = 0; i < 3; i++)
11 {
12 sum = sum + score[i] ;
13 }
14 printf("%d\n", sum);
15 }
	Data:
	0x00000000 scores
	0x00000012 sum
	
	Instruction:
	STORE 0x0, sum
	STORE 0x50, score
	STORE 0x4B, score+0x4
	STORE 0x64, score+0x8
	MOV r1, 0
	MOV r2, scores
	LABEL1:
	BGE r1, 0x3, LABEL2
	LOAD r3, sum
	LOAD r4, r2
	ADDi r3, r4
	STORE r3, sum
	ADDi r1, 0x1
	ADDi r2, 0x4
	GOTO LABEL1
	LABEL2:
	LOAD r1, sum
	SYSCALL "printf" "%d₩n", r1
	RETURN

Requirement Specification

Functional	
Code generation	The interpreter must export correspondent assembly code as a .txt file before interpretation. The generated assembly code must consist of data section and instruction section. Optimization level is not restricted. In most cases, architecture does not matter, but required to be described in the first internal structure document.