DF00100 Compiler Labs: Part 2 - LLVM back-end

August Ernstsson (based on previous material by Mattias Eriksson)

1. Adding an instruction to LLVM

One of the simplest target architectures that is supported by LLVM is Sparc. In this assignment you will add a theoretical instruction to the Sparc target.

Create an add-instruction that takes three operands: addthree a,b,c,d. This instruction should match the computation d=a+b+c, where:

A. a,b,c are all integers located in registers,

B. a,b,c are all floats located in registers,

C. a,b are integers in registers, and c is an immediate value. What is the largest value that c can have in this instruction? (This value depends on how you made your implementation.)

This assignment only includes making LLVM generate assembler code. I.e. you do not need to think about JIT-compiling.

Your report should include:

- Detailed explanations of what you modified in LLVM.
- Example runs showing that your new instruction is used.
- See lab intro slides for further information.

1.1 Hints

You can compile your example program to sparc assembler like this:

```
clang -O2 -S -emit-llvm foo.c
llvm-as foo.ll
llc --march=sparc --mcpu=generic --asm-verbose foo.bc
cat foo.s
```

Declare your variables to be volatile in your example program (foo.c). This will instruct the compiler not do optimize away all of your program. Like this:

```
volatile int x;
```