Assignment 3

1 Student Information

Name : Hady Januar Willi
Student ID : A0174396R
School Email : E0210429@u.nus.edu
Email : hadywilli@outlook.com

2 Design

Both task 1 and task 2 analysis are implemented on the basis of interval design, even though task 1 does not require completion of interval analysis. The ability of getting the maximum difference between two variables is a side effect of the result of interval analysis, which is more general analysis than maximum difference analysis.

The analysis is divided into two steps, widening first and then narrowing. In the implementation and analysis of result, any value that is -1000 represents negative infinity and any value that is 1000 represents positive infinity. All widening and narrowing handle of variable interval (Range in code) will check against this sanity, that is any value above 1000 will be set to 1000 and any value below -1000 will be set to -1000.

2.1 Branch Comparison

This analysis covers the following comparison instruction, with the assumption that either one side of the comparison is a constant, or the right hand side is a user input (i.e. not part of the analysis):

1. Greater than (>)
2. Greater than or equal to(>=)
3. Less than (<)
4. Less than or equal to(<=)
5. Equals(==)

Not equals comparison is not covered in the analysis as the condition for interval analysis is complex. The complexity can be thought of being analoguous to having two comparison of greater than and less than.

2.2 Instructions

The following instructions are covered according to the requirement in the assignment handout:

1. Alloca
2. Store

3. Load
4. SAdd
5. SSub
6. SMul
7. SDiv
8. SRem

The transfer function of instructions are simplified for generalization of interval analysis, therefore may not be fully accurate.

3 Submission Folder Structure

> zip		–	Folder containing all files
	> bin	–	Folder containing compiled assignment binary
	> graphs	–	Folder containing dot and pdf files generated from LLVM intermediate representation
	> installation	–	Folder containing LLVM 7 installation script
	> llvms	–	Folder containing LLVM intermediate representation
	> source	–	Folder containing assignment source code
	> target	–	Folder containing target C code

4 Running the Implementation

After extracting the file, the following commands can be run in sequence:

```
./compile
./llvm
./task2
./task3
```

After compilation, to run against any other LLVM intermediate representation file, the following command can be used:

```
./bin/assignment1 other-file.ll
```

To generate the graphs, the following command can be run:

```
./graph
```

If all else fails, the compiled binary is also included in the assignment, and can be used with the following command which hopefully runs with or without LLVM libraries:

```
./bin/assignment1/test.ll
```