Programming Assignment 2

Control Flow Graph & Data Flow Analysis

Objectives

In this assignment, you are asked to build a Control Flow Graph (CFG) for each function based on an Abstract Syntax Tree (AST) which you already built from Assignment 1. You also need to analyze live variables (Liveness Analysis) from beginning to the end of every basic block.

Due & Submission

- **Due:** Dec. 15 (Friday) midnight (23:59:59)
- **Submission:** Make a tarball(.tgz) which includes the following files and submit via icampus.
  - source code (parser + cfg, liveness)
  - Makefile
  - README

Requirements for submission

- Print out a Control Flow Graph from result of Assignment 1 (AST) as an input.
  - Control Flow Graph must be printed to CFG.out
- Find out Live variables from DFA
  - Result must be printed to Liveness.out
  - Use worklist algorithm with priority queue of postorder of CFG
  - Measure DFA running time by including Timer Code in beginning and end of the function
  - Print out DFA running time on the terminal

Notes:

- CFG and DFA must be implemented in different function
- You must follow the print out format on page 2 and 3 (exclude CFG figure)
- Deduct 10% of your total grade for every 1 day delayed submission (Max 50%)
- No cheat or plagiarize is allowed, 0 for every team

Questions & T.A. office hours

If you are not clear about this assignment, you can ask through Q/A by email, or you can visit TAs with a prior appointment.
CFG output format (Example)

```
# Control Flow Graph
B0
{  
  n<2
}
Predecessor: start
Successor: B1, B2
B1
{  
  x=3
}
Predecessor: B0
Successor: B3
B2
{  
  x=0
}
Predecessor: B0
Successor: B3
B3
{  
  goto 1
}
Predecessor: B1, B2
Successor: B5
B4
{  
  x=x+5
  y=z*3
}
Predecessor:
Successor: B5
B5
{  
  w=x+!0
  y=w*2
  return y
}
Predecessor: B3, B4
Successor: end
```
Liveness output format (Example)

```
# The Result of Liveness Analysis

<table>
<thead>
<tr>
<th></th>
<th>Begin (IN)</th>
<th>End (OUT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>{}</td>
<td>{i}</td>
</tr>
<tr>
<td>B1</td>
<td>{i}</td>
<td>{a, c, i}</td>
</tr>
<tr>
<td>B2</td>
<td>{i}</td>
<td>{a, b, c, d, i}</td>
</tr>
<tr>
<td>B3</td>
<td>{a, b, c, d, i}</td>
<td>{a, c, d, i}</td>
</tr>
<tr>
<td>B4</td>
<td>{}</td>
<td>{a, c, d, i}</td>
</tr>
<tr>
<td>B5</td>
<td>{a, c, d, i}</td>
<td>{i}</td>
</tr>
</tbody>
</table>
```

Timer Code

Refer to attached file. (‘timer.h’)