Distribution of Exam-I Scores

Average: 70
Median: 72
Highest: 92
CSC 201
Exam-I
April 28, 2016

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. **(30 points: 5+5+5+5+10)** For the control flow graph given below: (a) provide the depth first ordering of the nodes; (b) provide the dominator tree; (c) identify the loop back edges; (d) provide the dominance frontier set of each node; and (e) provide the corresponding SSA-form.

(a) Depth first ordering: 0, 1, 2, 5, 6, 5, 2, 3, 2, 1, 4, 1, 0

(b) Dominator relationships:

(c) Loop back edges: 6 → 5 and 5 → 1

(d) Immediate dominator:

(e) Dominance frontier:

\[
DF(\{0, 3, 6\}) = \{5\}
\]

\[
DF(\{0, 3, 6, 5\}) = \{1, 5\}
\]

Put % nodes here.
2. (20 points) Given the following intermediate code:

```
Read A1
B = A1
P = (B != A1)
If P then C1 = 1
Else C2 = 2
C3 = φ (C1,C2)
A2 = C3 + A1
X = C3 + B
D = B
Y = D + C3
```

Give the initial **congruence classes** constructed by the optimistic Global Value Numbering algorithm and the **final congruence classes** generated once all splitting has been performed.
3. (20 points) **Equality of Variables (X,Y)** - Given a pair of variables (X,Y), develop data flow analysis that computes **EQUAL(X,Y)** at each program point such that it is true if the values of X and Y are the same and false otherwise. The determination of equality should exploit following sources of equality: (a) copy statements of the form \( X = Y \) or \( Y = X \); and (b) conditionals of the form \( \text{if } X == Y \).

   a. Specify the meet operator for **EQUAL(X,Y)**
   b. Provide the transfer function for **EQUAL(X,Y)**
   c. Provide the data flow equations.
a) \[ \text{yes} \land \text{yes} = \text{yes} \quad \text{yes} \]
\[ \text{any} \land \text{no} = \text{no} \quad \text{no} \]

b) Assignment statement \( S \)
\[ f_S(v) = \]
\[ \begin{cases} \text{yes} & \text{if } S = x = y \text{ or } y = x \\ \text{no} & \text{if } S = x = \_ \text{ or } y = \_ \\ \text{\_} & \text{if } S \text{ is otherwise} \end{cases} \]

Conditional statement \( P \)
\[ f_P(v) = \]
\[ \begin{cases} \text{yes} & \text{if } P = x = y \\ \text{no} & \text{if } P = x \neq y \\ \text{\_} & \text{if } P \text{ is otherwise} \end{cases} \]

\[ \text{SOLN}(n) = \bigwedge_{S \in \text{pred}(n)} f_S(\text{SOLN}(S)) \bigwedge_{P_T \in \text{Tpred}(n)} f_{P_T}(\text{SOLN}(P_T)) \bigwedge_{P_F \in \text{Fpred}(n)} f_{P_F}(\text{SOLN}(P_F)) \]

- \( \text{Pred}(n) \) — edges have no label
- \( \text{Tpred}(n) \) — edges have label \( T \)
- \( \text{Fpred}(n) \) — edges have label \( F \)
4. (30 points) Given a variable $X$, develop data flow analysis that classifies the value of $X$ at each program point $p$ as being: (a) **DefinitelyAssignedConstant (DAC)** - if along every path leading to $p$, the latest definition of $X$ encountered assigns a constant value to $X$; (b) **PossiblyAssignedConstant (PAC)** - if along some (but not all) paths leading to $p$, the latest definition of $X$ encountered assigns a constant value to $X$; or (c) **NeverAssignedConstant (NAC)** - if along no path leading to $p$, the latest definition of $X$ encountered assigns a constant value to $X$.

Your solution must provide the following: (i) the information set $L$ and the top and bottom elements in $L$; (ii) the meet operator $\land$; (iii) the pictorial representation of the partial order; (iv) the transfer function; (v) the data flow equations; and (vi) the initialization of data flow values.
(i) \( L = \{ \text{UNDEF, DAC, PAC, NAC} \} \)

\[ T = \text{UNDEF} \quad (\text{iii}) \quad \text{UNDEF} (T) \quad \text{any} \land \text{UNDEF} = \text{any} \]

\[ L = \text{PAC} \quad \begin{array}{c}
\text{DAC} \\
\text{NAC} \\
\text{PAC (+)}
\end{array} \quad \text{any} \land \text{DAC} = \text{PAC} \quad \text{DAC} \land \text{NAC} = \text{PAC} \]

(iv) \( f_s (v) = \begin{cases} 
\text{DAC} & \text{if } S \text{ assigns a constant to } X \\
\text{NAC} & \text{if } S \text{ assigns non-constant to } X \\
v & \text{otherwise}
\end{cases} \)

(v) \( \text{SOLN}_x (n) = \bigwedge \left\{ f_s (\text{SOLN}_x (s)) \right\} \quad S \in \text{Pred}(n) \)

(vi) **Initialization:**

\( \text{SOLN}_x (n_0) = \text{PAC} \quad n_0 - \text{start node} \)

\( \text{SOLN}_x (n) = \text{UNDEF} \quad \forall n \in N - \{n_0\} \) (i.e., all other nodes)