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) Immediate Dominator Relationships

(c) Loop back edges: 6 → 5 and 5 → 1

(d) SSA-form: 
0: read x
1: if x > 15
2: if x < 10
3: x = x - 1
4: print x
5: if x > 10
6: x = x + 1

(e) Dominance Frontiers:

- \( DF(\{0, 3, 6\}) = \{2, 3, 4, 5, 6\} \)
- \( DF(\{0, 3, 6, 5\}) = \{1, 5\} \)

Put all nodes here.
(2)

- read $x_1$

- $x_2 = \phi(x_1, x_3)$
  - if $x_2 > 15$
  - if $x_2 < 10$
    - $x_3 = x_2 - 1$
  - print $x_2$

- $x_4 = \phi(x_2, x_3, x_5)$
  - if $x_4 > 10$
    - $x_5 = x_1 + 1$
2. (20 points) Given the following intermediate code:

Read A1
B = A1
P = (B != A1)
If P then C1 = 1
Else C2 = 2
C3 = \phi (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 \textsc{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 if \( X = Y \).

   a. Specify the meet operator for \textsc{Equal}(X,Y)
   b. Provide the transfer function for \textsc{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 \( S \)
\[ f_s(v) = \begin{cases} \text{yes} & S: x = y \text{ or } y = x \\ \text{no} & S: x = \_ \text{ or } y = \_ \\ \_ & S: \text{ otherwise} \end{cases} \]

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

\[ f_{p_F}(v) = \begin{cases} \text{yes} & P: x = y \\ \text{no} & P: x \neq y \\ \_ & P: \text{ otherwise} \end{cases} \]

\[ \text{SOLN}(n) = \bigwedge_{S \in \text{pred}(n)} f_s(\text{sOLN}(S)) \land \bigwedge_{p_T \in \text{Tpred}(n)} f_{p_T}(\text{sOLN}(p_T)) \land \bigwedge_{p_F \in \text{Fpred}(n)} f_{p_F}(\text{sOLN}(p_F)) \]

\text{pred}(n) - \text{edges have no label}
\text{Tpred}(n) - \text{edges have label } T
\text{Fpred}(n) - \text{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 $\Lambda$; (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) \( L = \{ \text{UNDEF, DAC, PAC, NAC} \} \)

\( T = \text{UNDEF} \)

\( T = \text{PAC} \)

\( L = \text{PAC} \)

\( \text{DAC} \)

\( \text{NAC} \)

\( \text{PAC}(+) \)

(ii) any \& \text{UNDEF} = \text{any} \)

any \& \text{PAC} = \text{PAC} \)

\( \text{DAC} \& \text{NAC} = \text{PAC} \)

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

(v) \( \text{SOLN}_X(h) = \bigwedge_{S \in \text{Pred}(h)} f_S(\text{SOLN}_X(s)) \)

(vi) Initialization:

\( \text{SOLN}_X(h_0) = \text{PAC} \quad h_0 \text{- start node} \)

\( \text{SOLN}_X(h) = \text{UNDEF} \quad \forall h \in \mathbb{N} - \{h_0\} \) (i.e., all other nodes)