Adventures in

Dataflow Analysis

CSE 401 Section 9-ish
Aaron Johnston & Nate Yazdani
Announcements

- Compiler Additions due next Thursday, 5/31
  - Involves revisiting all parts of the compiler
Announcements

- Compiler Additions due next Thursday, 5/31
  - Involves revisiting all parts of the compiler

- Final Report due the following Saturday, 6/2
  - Ideally, also involves revisiting all parts of the compiler
Review of Optimizations

Source Code → Front End → IR → Back End → Target Code

Scanner → Parser → Semantic Analysis → Optimization → Code Generation
Review of Optimizations

Peephole
Local
Intraprocedural / Global
Interprocedural
Review of Optimizations

- Peephole: A few Instructions
- Local
- Intraprocedural / Global
- Interprocedural
Review of Optimizations

Peephole  A few Instructions
Local     A Basic Block
Intraprocedural / Global
Interprocedural
Review of Optimizations

Peephole: A few Instructions
Local: A Basic Block
Intraprocedural / Global: A Function/Method
Interprocedural
Review of Optimizations

Peephole: A few Instructions
Local: A Basic Block
Intraprocedural / Global: A Function/Method
Interprocedural: A Program
Overview of Dataflow Analysis

- A framework for exposing properties about programs
- Operates using sets of “facts”
Overview of Dataflow Analysis

- A framework for exposing properties about programs
- Operates using sets of “facts”
- Just the initial discovery phase
  - Changes can then be made to optimize based on the analysis
Overview of Dataflow Analysis

- Basic Set Definitions for a Basic Block $b$:
  - $\text{IN}(b)$: facts true on entry to $b$
  - $\text{OUT}(b)$: facts true on exit from $b$
  - $\text{GEN}(b)$: facts created (and not killed) in $b$
  - $\text{KILL}(b)$: facts killed in $b$
1 (a) & (b)
Equations for Reaching Definitions

- Sets:
  - \( \text{DEFOUT}(b) \): set of definitions in \( b \) that reach the end of \( b \) (i.e., not subsequently redefined in \( b \))
  - \( \text{SURVIVED}(b) \): set of all definitions not obscured by a definition in \( b \)
  - \( \text{REACHES}(b) \): set of definitions that reach \( b \)

- Equations:
  - \( \text{REACHES}(b) = \bigcup_{p \in \text{preds}(b)} \text{DEFOUT}(p) \cup \bigcap_{p \in \text{preds}(b)} (\text{REACHES}(p) \cap \text{SURVIVED}(p)) \)
Overview of Dataflow Analysis

- Basic Set Definitions for a Basic Block \( b \):
  - \( \text{IN}(b) \): facts true on entry to \( b \)
  - \( \text{OUT}(b) \): facts true on exit from \( b \)
  - \( \text{GEN}(b) \): facts created (and not killed) in \( b \)
  - \( \text{KILL}(b) \): facts killed in \( b \)

\[
\text{OUT}(b) = \text{GEN}(b) \cup (\text{IN}(b) - \text{KILL}(b))
\]
1 (c) & (d)
<table>
<thead>
<tr>
<th>Block</th>
<th>GEN</th>
<th>KILL</th>
<th>IN (1)</th>
<th>OUT (1)</th>
<th>IN (2)</th>
<th>OUT (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>L0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>L2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3</td>
<td>L3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
L0:  a = 0  
L1:  b = a + 1  
L2:  c = c + b  
L3:  a = b * 2  
L4:  if a < N goto L1  
L5:  return c
L0:  \( a = 0 \)
L1:  \( b = a + 1 \)
L2:  \( c = c + b \)
L3:  \( a = b * 2 \)
L4:  \( \text{if } a < N \text{ goto L1} \)
L5:  return \( c \)

<table>
<thead>
<tr>
<th>Block</th>
<th>GEN</th>
<th>KILL</th>
<th>IN (1)</th>
<th>OUT (1)</th>
<th>IN (2)</th>
<th>OUT (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>L0</td>
<td>L3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>L1</td>
<td></td>
<td></td>
<td>L0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>L2</td>
<td></td>
<td></td>
<td>L0, L1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3</td>
<td>L3</td>
<td>L0</td>
<td></td>
<td>L0, L1, L2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4</td>
<td></td>
<td></td>
<td></td>
<td>L1, L2, L3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td></td>
<td></td>
<td></td>
<td>L1, L2, L3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
L0:  \( a = 0 \)  
L1:  \( b = a + 1 \)  
L2:  \( c = c + b \)  
L3:  \( a = b \times 2 \)  
L4:  \( \text{if } a < N \text{ goto L1} \)  
L5:  \( \text{return } c \)  

<table>
<thead>
<tr>
<th>Block</th>
<th>GEN</th>
<th>KILL</th>
<th>IN (1)</th>
<th>OUT (1)</th>
<th>IN (2)</th>
<th>OUT (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>L0</td>
<td>L3</td>
<td></td>
<td>L0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>L1</td>
<td></td>
<td>L0</td>
<td>L0, L1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>L2</td>
<td></td>
<td>L0, L1</td>
<td>L0, L1, L2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L3</td>
<td>L3</td>
<td>L0</td>
<td>L0, L1, L2</td>
<td>L1, L2, L3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L4</td>
<td></td>
<td></td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td></td>
<td></td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
L0:  \( a = 0 \)
L1:  \( b = a + 1 \)
L2:  \( c = c + b \)
L3:  \( a = b \times 2 \)
L4:  if \( a < N \) goto L1
L5:  return \( c \)

<table>
<thead>
<tr>
<th>Block</th>
<th>GEN</th>
<th>KILL</th>
<th>IN (1)</th>
<th>OUT (1)</th>
<th>IN (2)</th>
<th>OUT (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>L0</td>
<td>L3</td>
<td>L0</td>
<td>L0</td>
<td>L0</td>
<td>L0</td>
</tr>
<tr>
<td>L1</td>
<td>L1</td>
<td></td>
<td>L0</td>
<td>L0, L1</td>
<td>L0, L1, L2, L3</td>
<td>L0, L1, L2, L3</td>
</tr>
<tr>
<td>L2</td>
<td>L2</td>
<td></td>
<td>L0, L1</td>
<td>L0, L1, L2</td>
<td>L0, L1, L2, L3</td>
<td>L0, L1, L2, L3</td>
</tr>
<tr>
<td>L3</td>
<td>L3</td>
<td>L0</td>
<td>L0, L1, L2</td>
<td>L1, L2, L3</td>
<td>L0, L1, L2, L3</td>
<td>L1, L2, L3</td>
</tr>
<tr>
<td>L4</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td></td>
</tr>
</tbody>
</table>
L0:  a = 0  
L1:  b = a + 1  
L2:  c = c + b  
L3:  a = b * 2  
L4:  if a < N goto L1  
L5:  return c

<table>
<thead>
<tr>
<th>Block</th>
<th>GEN</th>
<th>KILL</th>
<th>IN (1)</th>
<th>OUT (1)</th>
<th>IN (2)</th>
<th>OUT (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>L0</td>
<td>L3</td>
<td>L0</td>
<td>L0</td>
<td>L0</td>
<td>L0</td>
</tr>
<tr>
<td>L1</td>
<td>L1</td>
<td></td>
<td>L0</td>
<td>L0, L1</td>
<td>L0, L1, L2, L3</td>
<td>L0, L1, L2, L3</td>
</tr>
<tr>
<td>L2</td>
<td>L2</td>
<td></td>
<td>L0, L1</td>
<td>L0, L1, L2</td>
<td>L0, L1, L2, L3</td>
<td>L0, L1, L2, L3</td>
</tr>
<tr>
<td>L3</td>
<td>L3</td>
<td>L0</td>
<td>L0, L1, L2</td>
<td>L1, L2, L3</td>
<td>L0, L1, L2, L3</td>
<td>L1, L2, L3</td>
</tr>
<tr>
<td>L4</td>
<td></td>
<td></td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
</tr>
<tr>
<td>L5</td>
<td></td>
<td></td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
<td>L1, L2, L3</td>
</tr>
</tbody>
</table>

Convergence!
2 (a) & (b)
1. 
   \[ Z = 4 \times B \]
   \[ Y = A + C \]

2. 
   \[ Y = 5 \]
   \[ Z = Y + B \]

3. 
   \[ X = A \times B \]
   \[ Z = Y + X \]

4. 
   \[ X = A \times B \]
   \[ Z = Y + X \]

5. 
   \[ Y = 3 \times B \]
   \[ Z = A + B \]

6. 
   \[ Y = 3 \times B \]
   \[ X = A \times B \]

7. 
   \[ Y = 2 \times B \]
1. 
   \[ Z = 4 \times B \]
   \[ Y = A + C \]
2. 
   \[ Y = 5 \]
   \[ Z = Y + B \]
3. 
   \[ X = A \times B \]
   \[ Z = Y + X \]
   \[ T_1 = 3 \times B \]
4. 
   \[ X = A \times B \]
   \[ Z = Y + X \]
   \[ T_2 = 2 \times B \]
5. 
   \[ Y = T_1 \]
   \[ Z = A + B \]
6. 
   \[ Y = T_1 \]
   \[ X = A \times B \]
7. 
   \[ Y = T_2 \]