Iterative Program Analysis

Mooly Sagiv
http://www.cs.tau.ac.il/~msagiv/courses/pal6.html
Tel Aviv University
640-6706

Textbook: Principles of Program Analysis
Chapter 2.1 (modified)
Subjects

- From programs to equations
- Examples of chaotic iterations
- Why can’t we stop early?
- Why can’t we start from top?
- Incompleteness
- Efficiency issues
Computing Constants

- Construct a control flow graph (CFG)
- Associate transfer functions with control flow graph edges
- Define a system of equations
- Compute the simultaneous least fixed point via Chaotic iterations
- The solution is unique
  - But order of evaluation may affect the number of iterations
A Simple Example

\[ z := 3 \]
\[ x := 1 \]

while \((x > 0)\) (  
  if \(x = 1\) then \[y := 7\]  
  else \[y := z + 4\]  
  \[x := 3\]  
)
A Simple Example: System of Equations

DF[1] = [x→0, z→0]
DF[2] = DF[1][z→3]#
DF[3] = DF[2][x→1]#
DF[4] = DF[3][x>0]# ⊓ DF[7][y:=7]#
DF[5] = DF[4][x≠1]#
DF[6] = DF[4][x=1]#
DF[7] = DF[5][y:=7]# ⊓ DF[7][y:=z+4]#
DF[8] = DF[3][x≤1]#
A Simple Example: Chaotic Iterations

\[
\begin{align*}
    &z := 3, \\
    &x := 1, \\
    &x \leq 0, \\
    &x > 0, \\
    &x = 1, \\
    &x \neq 1, \\
    &y := 7, \\
    &y := z + 4
\end{align*}
\]

<table>
<thead>
<tr>
<th>N</th>
<th>DF[N]</th>
<th>WL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([x \mapsto 0, y \mapsto 0, z \mapsto 0])</td>
<td>{1}</td>
</tr>
<tr>
<td>2</td>
<td>([x \mapsto 0, y \mapsto 0, z \mapsto 3])</td>
<td>{2}</td>
</tr>
<tr>
<td>3</td>
<td>([x \mapsto 1, y \mapsto 0, z \mapsto 3])</td>
<td>{3}</td>
</tr>
<tr>
<td>4</td>
<td>([x \mapsto 1, y \mapsto 0, z \mapsto 3])</td>
<td>{4, 8}</td>
</tr>
<tr>
<td>5</td>
<td>([x \mapsto 1, y \mapsto 0, z \mapsto 3])</td>
<td>{5, 6, 8}</td>
</tr>
<tr>
<td>6</td>
<td>([x \mapsto 1, y \mapsto 0, z \mapsto 3])</td>
<td>{6, 7, 8}</td>
</tr>
<tr>
<td>7</td>
<td>([x \mapsto 1, y \mapsto 7, z \mapsto 3])</td>
<td>{7, 8}</td>
</tr>
<tr>
<td>8</td>
<td>([x \mapsto 1, y \mapsto 7, z \mapsto 3])</td>
<td>{3, 8}</td>
</tr>
<tr>
<td>4</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{4, 8}</td>
</tr>
<tr>
<td>5</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{5, 6, 8}</td>
</tr>
<tr>
<td>6</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{6, 7, 8}</td>
</tr>
<tr>
<td>7</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{7, 8}</td>
</tr>
<tr>
<td>8</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{4, 8}</td>
</tr>
<tr>
<td>9</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{8}</td>
</tr>
<tr>
<td>10</td>
<td>([x \mapsto \top, y \mapsto \top, z \mapsto 3])</td>
<td>{}</td>
</tr>
</tbody>
</table>
A Simple Example: System of Equations

x := 3
z := 3
x := 1
x ≤ 0
x > 0
y := 7
y := z + 4

DF[1] = [x → 0, z → 0]
DF[2] = DF[1][z → 3]#
DF[3] = DF[2][x → 1]#
DF[4] = DF[3][x > 0]# ∩ DF[7][y := 7]#
DF[5] = DF[4][x ≠ 1]#
DF[6] = DF[4][x = 1]#
DF[7] = DF[5][y := 7]# ∩ DF[7][y := z + 4]#
DF[8] = DF[3][x ≤ 1]#

What happens when values are initialized to τ?
A Simple Example: System of Equations

1
\[ z := 3 \]

2
\[ x := 1 \]

3
\[ x \leq 0 \]

4
\[ x > 0 \]

5
\[ x = 1 \]

6
\[ x \neq 1 \]

7
\[ y := 7 \]

8
\[ y := z + 4 \]

DF[1] = [x \mapsto \perp, z \mapsto \perp]

DF[2] = DF[1][z \mapsto 3]

DF[3] = DF[2][x \mapsto 1]

DF[4] = DF[3][x > 0] \sqcap DF[7][y := 7]

DF[5] = DF[4][x \neq 1]

DF[6] = DF[4][x = 1]


DF[8] = DF[3][x \leq 1]
When do we lose precision

- Dynamic vs. Static values
- Correlated branches
- Locality of transformers (Join over all path)
- Initial value
Low Level View

- Explicitly represent the program counter
- Create an abstract transition system which represents the analysis
- Execute transitions in arbitrary order
Low Level View (Example)

State : PC $\rightarrow$ (Var $\rightarrow$ Val)

Transformer: State $\rightarrow$ State

while 3: (x > 0) ( 

4: if (x = 1) then 5: y = 7 
else 6: y = z + 4 

7: x = 3 
8: print y 
)

\forall S. \forall pc. \forall v:

\begin{align*}
0 & \quad pc = 1 \\
S 1 [z \mapsto 3] v & \quad pc = 2 \\
(S 2 [x \mapsto 1] \sqcup S 8) v & \quad pc = 3 \\
S 3 v & \quad pc = 4 \\
(S 4 \sqcap [x \mapsto 1, y \mapsto \top, z \mapsto \top]) v & \quad pc = 5 \\
S 4 v & \quad pc = 6 \\
(S 5 [y \mapsto 7] \sqcup (S 6 [y \mapsto (S 6 z) + 4] pc = 7 \\
S 7 [x \mapsto 3] v & \quad pc = 8
\end{align*}
Summary

- Chaotic iterations is a powerful technique
- Easy to implement
- Rather precise
- But expensive

- More efficient methods exist for structured programs