Iterative Program Analysis Part II Mathematical Background

Mooly Sagiv
http://www.cs.tau.ac.il/~msagiv/courses/pa11.html
Tel Aviv University
640-6706

Textbook: Principles of Program Analysis

Appendix A

Content

- Mathematical Background
- Chaotic Iterations
- Soundness, Precision and more examples next week

Mathematical Background

- Declaratively define
  - The result of the analysis
  - The exact solution
  - Allow comparison

Posets

- A partial ordering is a binary relation
  $\sqsubseteq : L \times L \rightarrow \{\text{false, true}\}$
  - For all $l \in L : l \sqsubseteq l$ (Reflexive)
  - For all $l_1, l_2, l_3 \in L : l_1 \sqsubseteq l_2, l_2 \sqsubseteq l_3 \Rightarrow l_1 \sqsubseteq l_3$ (Transitive)
  - For all $l_1, l_2 \in L : l_1 \sqsubseteq l_2, l_2 \sqsubseteq l_1 \Rightarrow l_1 = l_2$
    (Anti-Symmetric)
- Denoted by $(L, \sqsubseteq)$
- In program analysis
  - $l_1 \sqsubseteq l_2 \Leftrightarrow l_1$ is more precise than $l_2 \Leftrightarrow$
    l_1 represents fewer concrete states than l_2
- Examples
  - Total orders $(N, \leq)$
  - Powersets $(P(S), \subseteq)$
  - Powersets $(P(S), \supseteq)$
  - Constant propagation

Posets

- More notations
  - $l_1 \sqsupseteq l_2 \Leftrightarrow l_2 \sqsubseteq l_1$
  - $l_1 \sqsubset l_2 \Leftrightarrow l_1 \sqsubseteq l_2 \wedge l_1 \neq l_2$
  - $l_1 \sqsupset l_2 \Leftrightarrow l_2 \sqsubset l_1$

Upper and Lower Bounds

- Consider a poset $(L, \sqsubseteq)$
- A subset $L' \subseteq L$ has a lower bound $l \in L$ if for all $l' \in L'$: $l \sqsubseteq l'$
- A subset $L' \subseteq L$ has an upper bound $u \in L$ if for all $l' \in L'$: $l' \sqsubseteq u$
- A greatest lower bound of a subset $L' \subseteq L$ is a lower bound $l_0 \in L$ such that $l \sqsubseteq l_0$ for any lower bound l of L'
- A lowest upper bound of a subset $L' \subseteq L$ is an upper bound $u_0 \in L$ such that $u_0 \sqsubseteq u$ for any upper bound u of L'
- For every subset $L' \subseteq L$:
  - The greatest lower bound of L' is unique if at all exists
    - » $\sqcap L'$ (meet) $a \sqcap b$
  - The lowest upper bound of L' is unique if at all exists
    - » $\sqcup L'$ (join) $a \sqcup b$

Complete Lattices

- A poset $(L, \sqsubseteq)$ is a complete lattice if every subset has least and upper bounds
- $L = (L, \sqsubseteq) = (L, \sqsubseteq, \sqcup, \sqcap, \perp, \top)$
  - $\perp = \bigsqcup \emptyset = \bigsqcap L$
  - $\top = \bigsqcup L = \bigsqcap \emptyset$
- Examples
  - Total orders $(N, \leq)$
  - Powersets $(P(S), \subseteq)$
  - Powersets $(P(S), \supseteq)$
  - Constant propagation

Complete Lattices

- Lemma For every poset $(L, \sqsubseteq)$ the following conditions are equivalent
  - L is a complete lattice
  - Every subset of L has a least upper bound
  - Every subset of L has a greatest lower bound

Cartesian Products

- A complete lattice $(L_1, \sqsubseteq_1) = (L_1, \sqsubseteq, \sqcup_1, \sqcap_1, \perp_1, \top_1)$
- A complete lattice $(L_2, \sqsubseteq_2) = (, \sqsubseteq, \sqcup_2, \sqcap_2, \perp_2, \top_2)$
- Define a Poset $L = (L_1 \times L_2 , \sqsubseteq)$ where
  – $(x_1, x_2) \sqsubseteq (y_1, y_2)$ if
    » $x_1 \sqsubseteq y_1$ and
    » $x_2 \sqsubseteq y_2$
- L is a complete lattice

Finite Maps

- A complete lattice $(L_1, \sqsubseteq_1) = (L_1, \sqsubseteq, \sqcup_1, \sqcap_1, \perp_1, \top_1)$
- A finite set V
- Define a Poset $L = (V \rightarrow L_1, \sqsubseteq)$ where
  - $e_1 \sqsubseteq e_2$ if for all $v \in V$
    - » $e_1 v \sqsubseteq e_2 v$
- L is a complete lattice

Chains

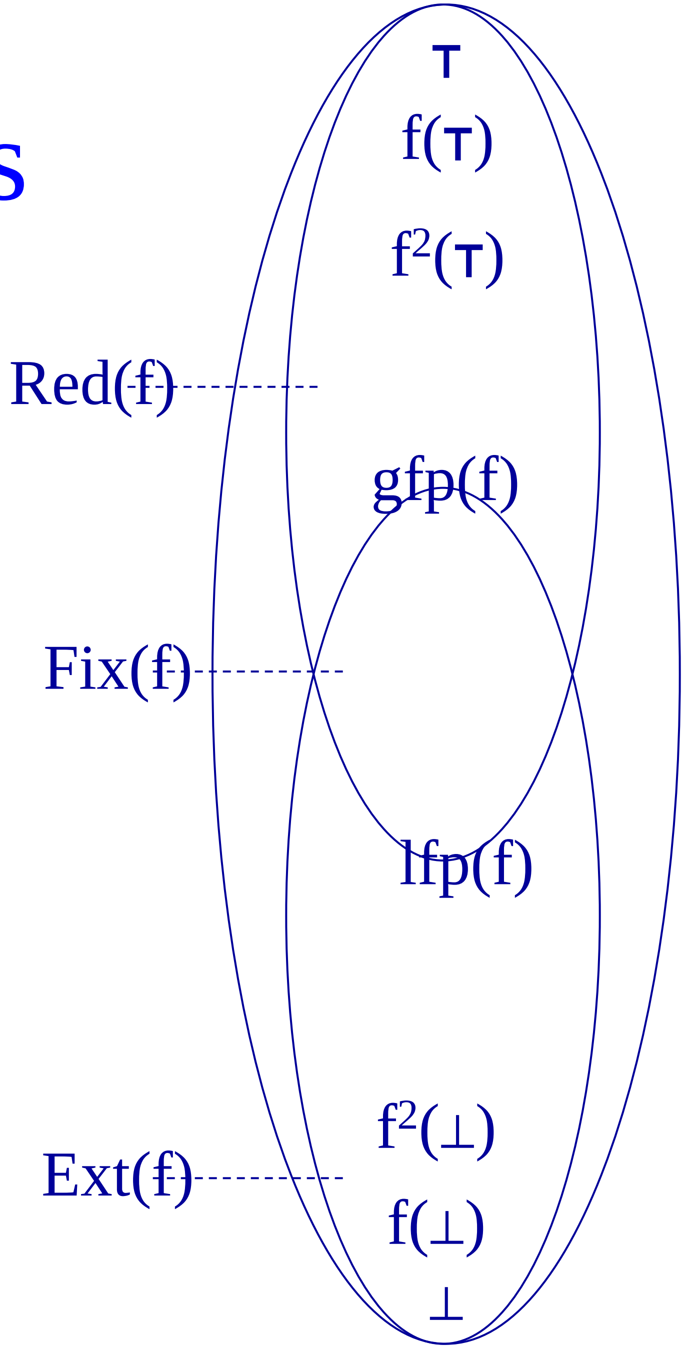
- A subset $Y \subseteq L$ in a poset $(L, \sqsubseteq)$ is a chain if every two elements in Y are ordered
  - For all $l_1, l_2 \in Y$: $l_1 \sqsubseteq l_2$ or $l_2 \sqsubseteq l_1$
- An ascending chain is a sequence of values
  - $l_1 \sqsubseteq l_2 \sqsubseteq l_3 \sqsubseteq \ldots$
- A strictly ascending chain is a sequence of values
  - $l_1 \sqsubset l_2 \sqsubset l_3 \sqsubset \ldots$
- A descending chain is a sequence of values
  - $l_1 \sqsupseteq l_2 \sqsupseteq l_3 \sqsupseteq \ldots$
- A strictly descending chain is a sequence of values
  - $l_1 \sqsupset l_2 \sqsupset l_3 \sqsupset \ldots$
- L has a finite height if every chain in L is finite
- Lemma A poset $(L, \sqsubseteq)$ has finite height if and only if every strictly decreasing and strictly increasing chains are finite

Monotone Functions

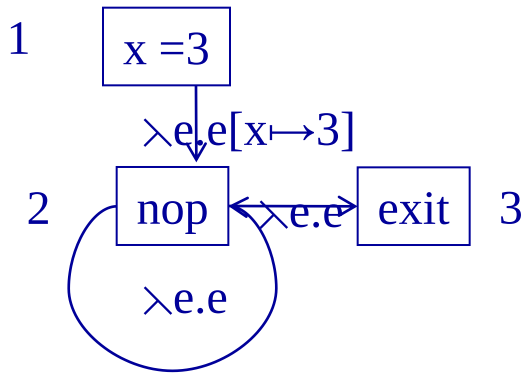
- A poset $(L, \sqsubseteq)$
- A function $f: L \to L$ is monotone if for every $l_1, l_2 \in L$:
  - $l_1 \sqsubseteq l_2 \Rightarrow f(l_1) \sqsubseteq f(l_2)$

Fixed Points

- A monotone function $f: L \to L$ where $(L, \sqsubseteq, \sqcup, \sqcap, \perp, \top)$ is a complete lattice
- $Fix(f) = \{ l: l \in L, f(l) = l\}$
- $Red(f) = \{l: l \in L, f(l) \sqsubseteq l\}$
- $Ext(f) = \{l: l \in L, l \sqsubseteq f(l)\}$
  - $l_1 \sqsubseteq l_2 \Rightarrow f(l_1) \sqsubseteq f(l_2)$
- Tarski's Theorem 1955: if f is monotone then:
  - $lfp(f) = \sqcap Fix(f) = \sqcap Red(f) \in Fix(f)$
  - $gfp(f) = \sqcup Fix(f) = \sqcup Ext(f) \in Fix(f)$



Example Constant Propagation



$$CP(1) = [x \mapsto 0]$$

$$CP(2) = CP(1)[x \mapsto 3] \sqcup CP(2)$$

$$CP(3) = CP(2)$$

Chaotic Iterations

- A lattice $L = (L, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$ with finite strictly increasing chains
- $L^n = L \times L \times \ldots \times L$
- A monotone function $\underline{f}: L^n \to L^n$
- Compute lfp($\underline{f}$)
- The simultaneous least fixed of the system $\{x[i] = \underline{f}_i(x) : 1 \leq i \leq n\}$

$\underline{x} := (\bot, \bot, \ldots, \bot)$

while ($\underline{f}(x) \neq \underline{x}$) do

$\quad \underline{x} := f(\underline{x})$

for i :=1 to n do

$\quad x[i] = \bot$

WL = {1, 2, …, n}

while (WL ≠ ∅) do

$\quad$ select and remove an element $i \in$ WL

$\quad$ new := $f_i(\underline{x})$

$\quad$ if (new ≠ x[i]) then

$\quad\quad$ x[i] := new;

$\quad\quad$ Add all the indexes that directly depends on i to WL

Chaotic Iterations

- $L^n = L \times L \times \ldots \times L$
- A monotone function $\underline{f}: L^n \rightarrow L^n$
- Compute lfp($\underline{f}$)
- The simultaneous least fixed of the system $\{x[i] = \underline{f}_i(x) : 1 \leq i \leq n \}$
- Minimum number of non-constant
- Maximum number of $\perp$

```
for i :=1  to n do
    x[i] = ⊥
WL = {1, 2, …, n}
while (WL ≠  Ø )  do
   select and remove an element i ∈ WL
   new := f_i(x)
    if (new ≠ x[i]) then
         x[i] := new;
         Add all the indexes that directly depends on i to WL
```

Specialized Chaotic Iterations System of Equations

$S =$

$$\left\{ \begin{array}{l} df_{entry}[s] = \iota \\ df_{entry}[v] = \sqcup\{f(u, v)\,(df_{entry}[u]) \mid (u, v) \in E\} \end{array} \right\}$$

$F_S : L^n \rightarrow L^n$

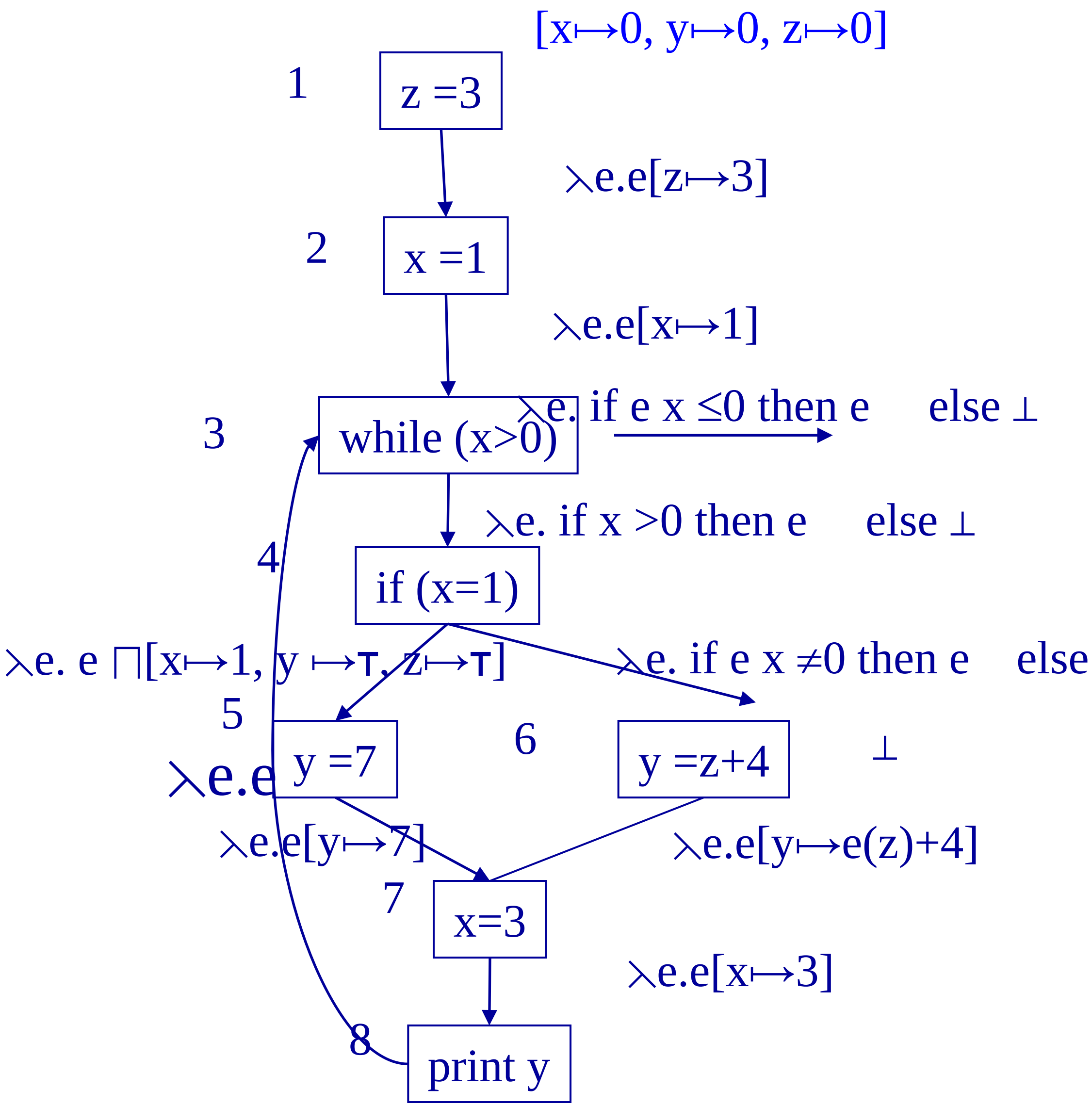
$$F_S\,(X)[s] = \iota$$

$$F_S(X)[v] = \sqcup\{f(u, v)(X[u]) \mid (u, v) \in E\}$$

$$lfp(S) = lfp(F_S)$$

Specialized Chaotic Iterations

```
Chaotic(G(V, E): Graph, s: Node, L: Lattice, ι: L, f: E →(L →L) ){
  for each v in V to n do df_entry[v] := ⊥
  df[s] = ι
  WL = {s}
  while (WL ≠ Ø ) do
    select and remove an element u ∈ WL
    for each v, such that. (u, v) ∈E do
          temp = f(e)(df_entry[u])
          new := df_entry(v)⊔ temp
          if (new ≠ df_entry[v]) then
                df_entry[v] := new;
                WL := WL ∪{v}
```



WL	$df_{entry}[v]$
{1}	
{2}	df[2]:=[x↦0, y↦0, z↦3]
{3}	df[3]:=[x↦1, y↦0, z↦3]
{4}	df[4]:=[x↦1, y↦0, z↦3]
{5}	df[5]:=[x↦1, y↦0, z↦3]
{7}	df[7]:=[x↦1, y↦7, z↦3]
{8}	df[8]:=[x↦3, y↦7, z↦3]
{3}	df[3]:=[x↦⊤, y↦⊤, z↦3]
{4}	df[4]:=[x↦⊤, y↦⊤, z↦3]
{5,6}	df[5]:=[x↦1, y↦⊤, z↦3]
{6,7}	df[6]:=[x↦⊤, y↦⊤, z↦3]
{7}	df[7]:=[x↦⊤, y↦7, z↦3]

Complexity of Chaotic Iterations

- Parameters:
  - n the number of CFG nodes
  - k is the maximum outdegree of edges
  - A lattice of height h
  - c is the maximum cost of
    - applying $f_{(e)}$
    - $\sqcup$
    - L comparisons
- Complexity

  $O(n * h * c * k)$

Conclusions

- Chaotic iterations is a powerful technique
- Easy to implement
- Rather precise
- But expensive
  - More efficient methods exist for structured programs