More
Data Flow Analyses

Reading: NNH 2.1

17-654/17-765
Analysis of Software Artifacts
Jonathan Aldrich

Announcements

- Assignment due at 11:59pm Thursday
  – Under Dean's door (Wean 8130)
  – Or via Blackboard
- Nicholas Sherman office hours
  – Tuesday at 4pm
- Reading for Thursday: NNH 2.2
- Questions on the homework?
  – Applications of sign analysis?

Some Notation

- This will help us describe analyses in a more precise and general way
  - *init*(S) – the label of the first statement in S
  - *final*(S) – the set of labels of the last statements in S
    - the last statement on each branch of an if
    - the test of a while
  - *blocks*(S) – the set of primitive statements and tests in S
  - *labels*(S) – the set of labels of blocks in S
  - *flow*(S) – $\{(\ell,\ell')$ | control may transfer from block ℓ to block ℓ' $\}$
    - A pair for each edge in the control flow graph

- The text defines these formally

General Data Flow Equations

Available Expressions

$AE_{entry}(\ell)$	=	$\varnothing$	if $(\ell = \mathit{init}(S_*))$
	=	$\cap \{ AE_{exit}(\ell') \mid (\ell', \ell) \in \mathit{flow}(S_*) \}$	otherwise

$AE_{exit}(\ell) = (AE_{entry}(\ell) \setminus kill_{AE}(B^\ell)) \cup gen_{AE}(B^\ell)$

Reaching Definitions

$RD_{entry}(\ell)$	=	$\mathbf{\{(x,?) \mid x \in FV(S_*)\}}$	if $\ell = \mathit{init}(S_*)$
	=	$\cup \{ RD_{exit}(\ell') \mid (\ell', \ell) \in \mathit{flow}(S_*) \}$	otherwise

$RD_{exit}(\ell) = (RD_{entry}(\ell) \setminus kill_{RD}(B^\ell)) \cup gen_{RD}(B^\ell)$

Safety and Precision: May vs. Must

[y := x]^{1};

[z := 1]^{2};

while [y>1]^{3} do

    [z := z * y]^{4};

    [y := y – 1]^{5};

[y := 0]^{6};

- What definitions ***may*** reach entry to 5?
  - Best answer? ***{ (y,1), (y,5), (z,4) }***
  - More precise (but unsafe)? ***{ (y,1), (z,4) }***
  - Safe but less precise? ***{ (y,1), (y,5), (z,1), (z,4) }***
- What expressions ***must*** be available at entry to 5?
  - Best answer? ***{ y>1 }***
  - More precise (but unsafe)? ***{ y>1, z*y }***
  - Safe but less precise? ∅

Safety and Precision: May vs. Must

[y := x]^1;

[z := 1]^2;

while [y>1]^3 do

  [z := z * y]^4;

  [y := y – 1]^5;

[y := 0]^6;

- What definitions ***may*** reach entry to 5?
  - Best answer? ***{ (y,1), (y,5), (z,4) }***
  - More precise (but unsafe)? ***{ (y,1), (z,4) }***
  - Safe but less precise? ***{ (y,1), (y,5), (z,1), (z,4) }***
- What expressions ***must*** be available at entry to 5?
  - Best answer? ***{ y>1 }***
  - More precise (but unsafe)? ***{ y>1, z*y }***
  - Safe but less precise? ∅

Safety and Precision: May vs. Must

```
[y := x]1;
[z := 1]2;
while [y>1]3 do
    [z := z * y]4;
    [y := y – 1]5;
[y := 0]6;
```

- What definitions ***may*** reach entry to 5?
  - Best answer? ***{ (y,1), (y,5), (z,4) }***
  - More precise (but unsafe)? ***{ (y,1), (z,4) }***
  - Safe but less precise? ***{ (y,1), (y,5), (z,1), (z,4) }***
- What expressions ***must*** be available at entry to 5?
  - Best answer? ***{ y>1 }***
  - More precise (but unsafe)? ***{ y>1, z*y }***
  - Safe but less precise? ∅

Reaching Defs. vs. Available Exp.

- Reaching Defs. ***May*** analysis
  - Initial dataflow values: *empty* sets
  - *Union* at control flow merge
  - Precision: want *least* fixed point
  - Safety: err on the side of *larger* sets
- Available Exp. ***Must*** analysis
  - Initial dataflow values: *universal* sets
  - *Intersection* at control flow merge
  - Precision: want *greatest* fixed point
  - Safety: err on the side of *smaller* sets

Constant Propagation

- For each program point, what value *may* each variable hold?

Constant Propagation

$CP_{entry}(\ell) = \{(x,n) \mid x \in FV(S_*), n \in N\}$ if $(\ell = init(S_*))$
$= \cup \{ CP_{exit}(\ell') \mid (\ell', \ell) \in flow(S_*) \}$ otherwise

$CP_{exit}(\ell) = (CP_{entry}(\ell) \setminus kill_{CP}(B^\ell)) \cup gen_{CP}(B^\ell)$

$kill_{CP}([x := a]^\ell) =$
$kill_{CP}([skip]^\ell) =$
$kill_{CP}([b]^\ell) =$

$gen_{CP}([x := a]^\ell) =$
$gen_{CP}([skip]^\ell) =$
$gen_{CP}([b]^\ell) =$

Constant Propagation

$CP_{entry}(\ell) = \{(x,n) \mid x \in FV(S_*), n \in N\}$ if $(\ell = init(S_*))$
$= \cup \{ CP_{exit}(\ell') \mid (\ell', \ell) \in flow(S_*) \}$ otherwise

$CP_{exit}(\ell) = (CP_{entry}(\ell) \setminus kill_{CP}(B^{\ell})) \cup gen_{CP}(B^{\ell})$

$kill_{CP}([x := a]^{\ell}) = \{ (x,n) \mid n \in N \}$
$kill_{CP}([skip]^{\ell}) = \varnothing$
$kill_{CP}([b]^{\ell}) = \varnothing$

$gen_{CP}([x := a]^{\ell}) = \{ (x, n) \mid n \in \mathbf{CP}^{\ell}(a)) \}$
$gen_{CP}([skip]^{\ell}) = \varnothing$
$gen_{CP}([b]^{\ell}) = \varnothing$

Constant Propagation

$\mathbf{CP}^{\ell}(x) = \{ \, CP_{entry}(\ell)(x) \, \}$

$\mathbf{CP}^{\ell}(n) = \{ \, n \, \}$

$\mathbf{CP}^{\ell}(a_1 \; op_a \; a_2) = \mathbf{CP}^{\ell}(a_1) \; \widehat{op}_a \; \mathbf{CP}^{\ell}(a_2)$

$set_1 \; \widehat{op}_a \; set_2 = \{ \, n_1 \; op_a \; n_2 \mid n_1 \in set_1, \; n_2 \in set_2 \, \}$

Constant Propagation Example

[y := 5]^1;
[z := 1+y]^2;
if [x = 5]^3
    then [w := x+1]^4;
    else [w := y+1]^5
[skip]^6

	x	y	z	w
$AE_{exit}(1)$ =	?	5	?	?
$AE_{exit}(2)$ =	?	5	6	?
$AE_{exit}(3)$ =	?	5	6	?
$AE_{exit}(4)$ =	?	5	6	?
$AE_{exit}(5)$ =	?	5	6	6
$AE_{exit}(6)$ =	?	5	6	?

Here ? is a shorthand for the set of all integers

Constant Propagation Example

[y := 5]^1;
[z := 1+y]^2;
if [x = 5]^3
  then [w := x+1]^4;
  else [w := y+1]^5
[skip]^6

	x	y	z	w
$AE_{exit}(1) =$	?	5	?	?
$AE_{exit}(2) =$	?	5	6	?
$AE_{exit}(3) =$	?	5	6	?
$AE_{exit}(4) =$	?	5	6	?
$AE_{exit}(5) =$	?	5	6	6
$AE_{exit}(6) =$	?	5	6	?

But we know that x=5 at statement 4. Can we do better?

Constant Propagation, Take 2

$CP_{entry}(\ell)$ = $\{(x,n) \mid x \in FV(S_*), n \in N\}$ if $(\ell = init(S_*))$
= $\cup \{ CP_{exit}(\ell') \mid (\ell', \ell) \in flow(S_*) \}$ otherwise

$CP_T(\ell) = (CP_{entry}(\ell) \setminus kill_{CP,T}(B^\ell)) \cup gen_{CP}(B^\ell)$
$CP_F(\ell) = (CP_{entry}(\ell) \setminus kill_{CP,F}(B^\ell)) \cup gen_{CP}(B^\ell)$

$kill_{CP,T}([x := a]^\ell) = \{ (x, n) \mid n \in N \}$
$kill_{CP,T}([skip]^\ell) = \varnothing$
$kill_{CP,T}([b]^\ell)$ = $\{ (x, n) \mid n \in N \text{ and } n \neq m\}$ if $b = (x=a)$ and $\mathbf{CP}^\ell(a)=\{m\}$
= $\varnothing$ otherwise
$kill_{CP,F}([b]^\ell)$ = $\{ (x, m) \}$ if $b = (x=a)$ and $\mathbf{CP}^\ell(a)=\{m\}$
= $\varnothing$ otherwise

$gen_{CP}([x := a]^\ell) = \{ (x, n) \mid n \in \mathbf{CP}^\ell(a)) \}$
$gen_{CP}([skip]^\ell) = \varnothing$
$gen_{CP}([b]^\ell) = \varnothing$

Constant Propagation Example

[y := 5]^1;
[z := 1+y]^2;
if [x = 5]^3
  then [w := x+1]^4;
  else [w := y+1]^5
[skip]^6

	x	y	z	w
$AE_{exit}(1)$ =	?	5	?	?
$AE_{exit}(2)$ =	?	5	6	?
$AE_T(3)$ =	5	5	6	?
$AE_F(3)$ =	?\5	5	6	?
$AE_{exit}(4)$ =	5	5	6	6
$AE_{exit}(5)$ =	?\5	5	6	6
$AE_{exit}(6)$ =	?	5	6	6

Keeping track of data flow values separately on each branch supports a more precise final result.

General Monotonicity Proofs

- We proved RD was monotone for data flow equations for a *specific program*
- Here's a more general proof, for the assignment flow function:
  - To show: If $RD_{entry}(\ell) \subseteq RD_{entry}'(\ell)$ then $RD_{exit}(\ell) \subseteq RD_{exit}'(\ell)$
    - case: $B^{\ell} = [x := a]^{\ell}$
  - Assume $RD_{entry}(\ell) \subseteq RD_{entry}'(\ell)$
  - Now $kill_{RD}([x := a]^{\ell}) = \{ (x, *) \}$ (where * is any label or ?)
  - Thus $RD_{entry}(\ell) \setminus kill_{RD}(B^{\ell}) \subseteq RD_{entry}'(\ell) \setminus kill_{RD}(B^{\ell})$
  - And $gen_{RD}([x := a]^{\ell}) = \{ (x, \ell) \}$
  - Therefore $(RD_{entry}(\ell) \setminus kill_{RD}(B^{\ell})) \cup gen_{RD}(B^{\ell}) \subseteq (RD_{entry}'(\ell) \setminus kill_{RD}(B^{\ell})) \cup gen_{RD}(B^{\ell})$
    - And we are done with the case for $[x := a]^{\ell}$