Reaching Definitions

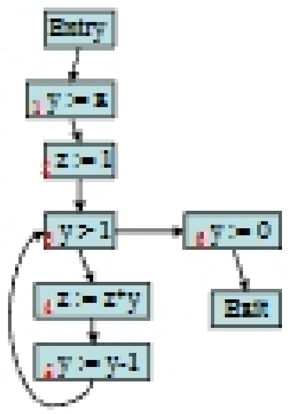
Done many ways

How many ways can we do Reaching Definitions?

- Data Flow analysis
- Constraint based approach
- Abstract Interpretation
- Type and Effect Systems

Running Example

```
y := x;  // 1
z := 1;  // 2
while y > 1 do {  // 3
    z := z*y;  // 4
    y := y-1 };  // 5
y := 0  // 6
```



Data Flow Analysis

- $RD_{exit}(1) = RD_{entry}(1) - \{ (y, l) \mid l \text{ in Lab} \} \cup \{(y, 1)\}$
- $RD_{exit}(2) = RD_{entry}(2) - \{ (z, l) \mid l \text{ in Lab} \} \cup \{(z, 2)\}$
- …
- $RD_{entry}(1) = \{ (x, ?), (y, ?), (z, ?) \}$
- $RD_{entry}(2) = RD_{exit}(1)$
- $RD_{entry}(3) = RD_{exit}(2) \cup RD_{exit}(5)$
- …

Finding a solution

- Let F be this set of equations
- Find RD s.t. RD = F(RD)
  - find least such solution
  - what is a non-minimal solution?

Constraint based approach

- $RD_{exit}(1) \supseteq RD_{entry}(1) - \{ (y, l) \mid l \text{ in Lab} \}$
- $RD_{exit}(1) \supseteq \{(y, 1)\}$
- $RD_{exit}(2) \supseteq RD_{entry}(2) - \{ (z, l) \mid l \text{ in Lab} \}$
- $RD_{exit}(2) \supseteq \{(z, 2)\}$
- …
- $RD_{entry}(1) \supseteq \{ (x, ?), (y, ?), (z, ?) \}$
- $RD_{entry}(2) \supseteq RD_{exit}(1)$
- $RD_{entry}(3) \supseteq RD_{exit}(2)$
- $RD_{entry}(3) \supseteq RD_{exit}(5)$
- …

Something different

- $[[\text{ fn } \mathbf{x} \Rightarrow [\mathbf{x}]^1]^2 [\text{ fn } y \Rightarrow [y]^3]^4]^5$
- For each function application, which function may be applied?

- $C(l)$ = values that subexpression l can evaluate to
- $p(x)$ = values that x can be bound to
- $[[\text{ fn } \mathbf{x} \Rightarrow [\mathbf{x}]^1]^2 [\text{ fn } y \Rightarrow [y]^3]^4]^5$
- $\{ \text{ fn } \mathbf{x} \Rightarrow [\mathbf{x}]^1 \} \subseteq C(2)$
- $\{ \text{ fn } y \Rightarrow [y]^3 \} \subseteq C(4)$

Values taken on by a variable use

- [[fn **x** => [**x**]^1]^2 [fn y => [y]^3]^4]^5
- p(**x**) $\subseteq$ C(1)
- p(y) $\subseteq$ C(3)

Results of function application

- [[fn x => [x]^1]^2 [fn y => [y]^3]^4]^5
- Only one function application
  - but we don't know what function will be applied
- If function might be fn x => [x]^1
  - { fn x => [x]^1 } $\subseteq$ C(2) $\Rightarrow$ C(4) $\subseteq$ p(x)
  - { fn x => [x]^1 } $\subseteq$ C(2) $\Rightarrow$ C(1) $\subseteq$ C(5)
- If function might be fn y => [y]^3
  - { fn y => [y]^3 } $\subseteq$ C(2) $\Rightarrow$ C(4) $\subseteq$ p(y)
  - { fn y => [y]^3 } $\subseteq$ C(2) $\Rightarrow$ C(3) $\subseteq$ C(5)

Least solution

- [[fn x => [x]^1]^2 [fn y => [y]^3]^4]^5
- C(1) = { fn y => [y]^3 }
- C(2) = { fn x => [x]^1 }
- C(3) = { }
- C(4) = { fn y => [y]^3 }
- C(5) = { fn y => [y]^3 }
- p(x) = { fn y => [y]^3 }
- p(y) = {}

Abstract Interpretation

- A trace is a series of (label, variable) pairs indicating the sequence of assignments
- A collecting semantics records the set of traces tr that can reach a given program point
- RD(tr)(**x**) = *l* iff the rightmost pair (**x**, *l'*) in tr has *l* = *l'*