Week 6.1

Tuesday, February 13, 2024 4:58 PM

$X \in$ Variable

$c \in$ Constructor

$t \in \text{Term} ::= x \mid c(t_1, \ldots, t_n)$

$e \in \text{Exp} ::= t \mid c^{-i}(x)$

$s \in \text{Stmt} ::= e_1 \subseteq e_2$

$f(X,Y) \subseteq A$

$A \subseteq B$

$B \subseteq C$

$C \subseteq E$

$g \subseteq D$

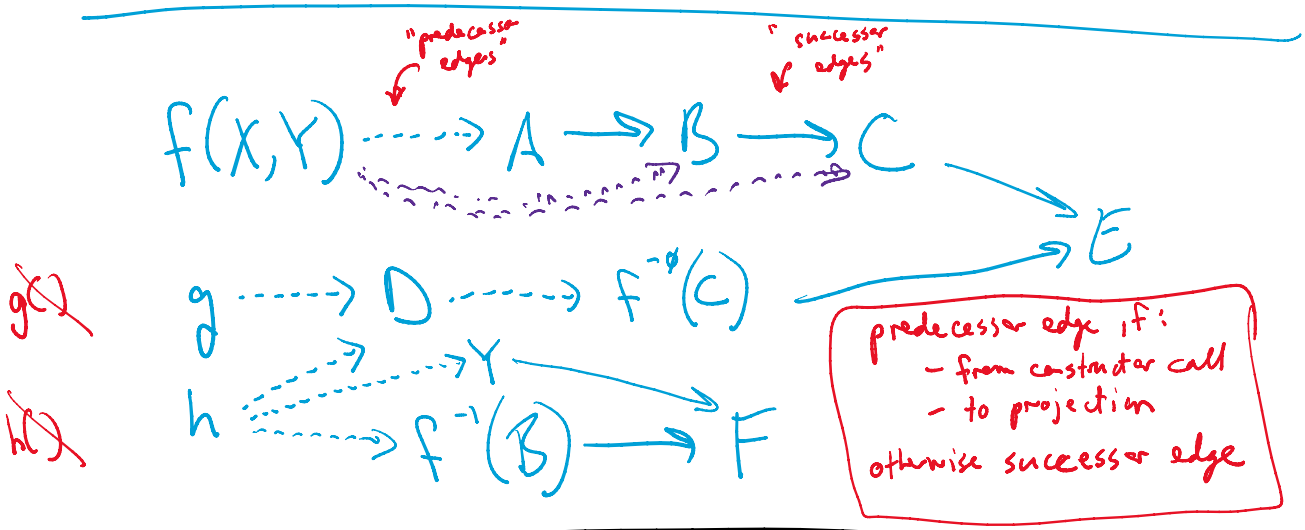
$h \subseteq D$

$h \subseteq f^{-1}(B)$

$D \subseteq f^{-\phi}(C)$

$f^{-\phi}(C) \subseteq E$

$f^{-1}(B) \subseteq F$



---

solving alg.

solving alg.

- worklist initialized w/ all set variable nodes that have a pred. edge
- while worklist isn't empty:
  - pop set variable X
  - propagate X's pred. edges along its succ. edges
    - if dest. node's pred edges change, put on worklist
  - for each projection node P of X:
    - let Y = value of P
    - for each predecessor p_i of P and successor s_i of P and each $y_i \in Y$:
      - add edge $p_i \to y_i$ } using standard rules for pred/succ edges
      - add edge $y_i \to s_i$ }
      - if p_i, s_i, y_i has new edges, add to worklist

what if we're adding an edge between constructor calls?

$$foo(X,Y) \dashrightarrow W \longrightarrow foo(A,B)$$

$$c(t_1, \ldots, t_n) \subseteq c(t_1', \ldots, t_n')$$

- if i is covariant: $t_i \subseteq t_i'$
- if i is contravariant: $t_i' \subseteq t_i$

what if we have $c(\ldots) \subseteq d(\ldots)$?

- technically an error

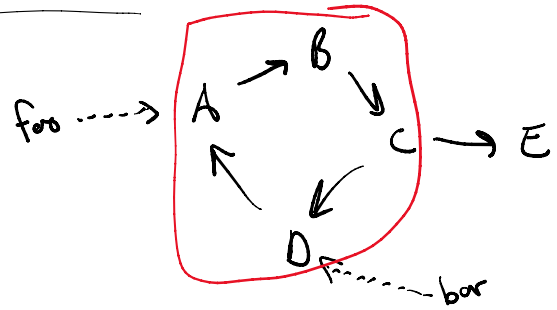
when it be ...

↳ technically an error
↳ usually ignored

---

optimizing solver

cycles



---

defining a program analysis

two steps:

1. define universe of discourse
2. defing constraint generation

Andersen-style pointer analysis

↳ for a given pointer access what object might be accessed?

memory abstraction

↳ static allocation site

x = $alloc <op> [id]

y = $copy x

z = $alloc <op> [id2]

universe of discourse

- a constant (nullary constructor) & a set variable for each program variable
  - constant: represents the program variable
  - set variable: represents the program variable's points-to set
- a ref/2 constructor (arity 2)
  - 1st arg is a constant
  - 2nd arg is a set variable

  } connect program variable w/ its points-to set

ex ref(x, X)

this is sufficient for all but indirect calls

generation

- (we only care about ptrs, so ignore all instructions whose lhs is not a pointer)
- (ignore null ptrs)
- notation: given program variable x
  - [x] to mean the set variable for x
  - const(x) to mean the constant for x

- `X = $copy y`

  $[y] \subseteq [x]$