Meet-over-paths
analysis using
automata.

Interprocedural flow analysis
using pushdown automata

Fix D – a set of facts (finite)

$$f_s : 2^D \longrightarrow 2^D$$

Program P.

$$\cup$$

$$MOP(n) = \prod_{\substack{\pi = n_0 n_1 \ldots n_k \,|\\ n_k = n}} f_\pi(\emptyset)$$

$$f_{n_0 \ldots n_k}(S) = f_{n_k}(\ldots f_{n_1}(f_{n_0}(S)))$$

Program P $\cancel{\in}$:

Loc: $\{p_1, p_2, \ldots p_n\}$

Edge: $\cancel{\{}$ $P \times S \times P$

Dataflow facts : D, Flow functions $\{f_s \mid s \in S\}$

$f_s : 2^D \to 2^D$

Automaton . $Q = P \times 2^D$

$$(p, x) \longrightarrow (p', x')$$

if $\exists s \in S.$ $p \xrightarrow{s} p'$ &

$x' = f_s(x)$ ...

Initial state : (P_{init}, ϕ)

$d \in MOP(n)$ iff

a state in

$\{(P_n, X) \mid d \in X\}$

is reachable.

$\{ \ldots d \} \bullet n$

<u>OR</u> Make $F = \{(P_n, X) \mid d \in X\}$

Is language of automaton empty?

Interprocedural flows.

```
main ( ) {
  1 int x;
  2 read (x)
  3 call P(x)
}

P ( a ) {
  if (a>0) 4 then {
    read (g) 5
    a := a-g 6
    call P(a) 7
  }
  else return;
```

push 3

push 7

pop 3

pop 7

Pushdown automata model
Interprocedural control flow.

- At a call, push current pc.
- At a return, pop pc and goto pc.
- Calls & Returns are deterministic.
- Captures control flow accurately.

Pushdown automata.

Σ

$$\mathcal{A} = (Q, q_0, \delta, \Gamma, F)$$

finite set of states

finite alphabet

$F \subseteq Q$

$(q, a, b, c, q') \in \delta$

$q \xrightarrow{a,\ \text{pop } b,\ \text{push } c} q'$

$$\delta \subseteq Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \times \Gamma_\varepsilon \times Q$$

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$$
$$\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$$

Config: (q, σ) $q \in Q$
$\sigma \in \Gamma^*$.

Moves: $a \in \Sigma_\epsilon$

$$(q, \sigma) \xrightarrow{a} (q', \sigma')$$

if $\exists\ (q, a, b, c, q') \in \delta$

s.t. $\sigma = b\sigma_1$ for some $\sigma_1 \in \Gamma^*$

$\sigma' = c\sigma_1$.

Note: # of configs can be infinite.

A word $w \in \Sigma^*$ is accepted if there is a path in the config graph

$$(q_0, \varepsilon) \xrightarrow{a_1} (q_1, \sigma_1) \to \dots \xrightarrow{a_n} (q_n, \sigma_n)$$

such that $w = a_1 \dots a_n$ and $q_n \in F$.

A word is acc. on empty-stack if there is a run with $\sigma_n = \varepsilon$.

Facts.

- Emptiness problem (i.e. $L(A) = \emptyset$) is decidable for PDAs. $O(n^3)$.
- Membership is decidable
- PDAs are <u>not</u> closed under complement.
- PDs are <u>not</u> determinizable.

Interprocedural flow analysis

$\longrightarrow$ PDA reachability/
emptiness.