Meet-over-paths
analysis using
automata.
Interprocedural flow analysis
using pushdown automata
Fix $D$ - a set of facts (finite)

$f_s : 2^D \rightarrow 2^D$

Program $P$.

$\text{MOP}(n) = \bigvee_{\pi = n_0 n_1 \ldots n_k} f_{\pi}(\phi)$

$f_{n_0 \ldots n_k}(s) = f_{n_k}(\ldots f_{n_1}(f_{n_0}(s)))$
Program $P = 1$

Loc: $\{P_1, P_2, \ldots, P_n\}$

Edge: $\forall P \times S \times P$

Dataflow facts: $D$, Flow finding

Automaton $Q = P \times 2^D$

$(P, x) \rightarrow (P', x')$

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

$x' = f_S(x)$
Initial state: \((P_{init}, \emptyset)\)

\(d \in \text{MOP}(n) \iff\)

a state in \(\{ (p_n, x) \mid d \in x \} \)

is reachable.

OR

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)

1. if (a > 0)
2. return
3. a := a - g
4. call P(a)

push

pop

5
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. \( \Sigma \)

\[ A = (Q, q_0, \delta, \Pi, F) \]

- \( Q \) is a finite set of states
- \( \Sigma \) is a finite alphabet
- \( q_0 \) is the initial state
- \( \delta \) is the transition function
- \( \Pi \) is the stack alphabet
- \( F \) is the set of final states

Transition function:

\[ \delta: (q, a, b, c, q') \in \delta \]

- \( q \) is the current state
- \( a \) is the input symbol
- \( b \) is the top of the stack
- \( c \) is the symbol pushed onto the stack
- \( q' \) is the next state

\[ \delta \subseteq Q \times \Sigma \times \Pi \times \Pi \times Q \]

- \( \Sigma \) is the input alphabet
- \( \Pi \) is the stack alphabet

Stack alphabet:

\[ \Sigma_\epsilon = \Sigma \cup \{ c \} \]

- \( \Sigma_\epsilon \) is the extended alphabet

Input alphabet:

\[ \Pi_\epsilon = \Pi \cup \{ \epsilon \} \]

- \( \Pi_\epsilon \) is the extended input alphabet
Config: \((q, \sigma) \quad q \in \mathcal{Q} \\
\quad \sigma \in \mathcal{I}^*\). 

Moves: 
\[
(q, \sigma) \xrightarrow{a \in \mathcal{I}_e} (q', \sigma')
\]
if \( \exists (q, a, b, c, q') \in \delta \) 

s.t. \( \sigma = b \sigma_i \quad \text{for some} \sigma, \sigma_i \in \mathcal{I}^* \)

\[
\sigma^{-1} = c \sigma_i
\]

Note: \# of configs can be infinite.
A word \( w \in \Sigma^* \) is accepted if there is a path in the config graph
\[
( q_0, \varepsilon ) \xrightarrow{a_1} ( q_1, \sigma_1 ) \xrightarrow{\cdots} ( q_n, \sigma_n )
\]
such that \( w = a_1 \cdots a_n \) and \( q_n \in F \).

A word is accepted on an 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 not closed under complement.
- PDs are not determinizable.
Interprocedural flow analysis

$\Rightarrow$ PDA reachability conjectures.