Generic Program
Querying of Higher-Order Languages

Jens Nicolay
Idea

SE tools require static analysis

Scheme
JavaScript
Java
...

liveness
dependence
purity
constant propagation
...

...
Idea

SE tools require static analysis

closures
objects
higher-order
late binding
mutation

liveness
dependence
control flow
value flow
effects
constant propagation

common ground
Idea

Don’t start from scratch.
Pushdown.pathsBwTo =
function (s, target, etg) {
  var todo = [s];
  var visited = ArraySet.empty();
  var paths = ArraySet.empty();
  while (todo.length > 0) {
    var q = todo.shift();
    if (q.equals(target) || visited.contains(q)) {
      continue;
    }
    visited = visited.add(q);
    var incoming = etg.incoming(q);
    paths = paths.addAll(incoming);
    var qs = incoming.map(Edge.source);
    todo = todo.concat(qs);
  }
  return paths.values();
}
Plan

- Liveness analysis
- Purity analysis
- Dependence analysis
- Constant propagation

...
Abstract Interpretation

- finite number of abstract values
- finite number of addresses

(Abstracting Abstract Machines, Van Horn and Might, ICFP2010)
(Pushdown control-flow analysis of higher-order programs, Earl et al., Scheme2010)
Pushdown Analysis

+ pushdown driver

(Pushdown control-flow analysis of higher-order programs, Earl et al., Scheme2010)
Dyck State Graph

A → B: +1
B → C: +2
C → D: −2
D → E: −1
B → F: +3
F → G: +4
G → D: −4
D → E: −1
Legal Path
finite representation of unbounded stack
Stack Change

A → B (+1) → F (+3) → G (+4) → D (-4) → E (-1)
B → D (-3)
C → D (-2)
D → E (-1)
Stack Change

A → B: +1
B → F: +3
F → G: +4
G → D: -4
D → E: -1
C → D: -2
D → B: -3
B → C: +2
DSG Querying
topOfStack
top0fStack

A → B → C → D → E

+1, +2, +3, +4
valueOf
Effects

A → B: +1
B → A: A $a_2$, R $a_1$
B → C: +2
C → D: A $a_1$, W $a_1$
C → B: R $a_2$
D → E: R $a_2$
D → B: −1
D → C: −2
C → B: −3
F → C: +3
F → G: +4
G → D: −4
G → B: −4
E → D: −1
Wrap Up

- build generic pushdown analyses machinery
  - input languages
  - Dyck state graph queries
- find and exploit commonalities
- implementations: https://github.com/jensnicolay

Thanks!
1 + 2