

# Smalltalk

## Metaprogramming supports Probabilistic Program Analysis

Dave Mason  
Ryerson University

# Why Path Discovery?

- Testing/coverage
- Program Comprehension/Visualization
- Software Reliability

# Domain/Path Discovery

- Depth-first
- Breadth-first
- Heuristic
- Probabilistic

# Operational Profile

- probability of each point in the input domain
- histogram or continuous probabilistic density function
- derived from real world / "expert"
- we will use to drive program analysis

# Continuation-Based

- all parameters are variables of the PDF
- at any decision point (loop/if) involving those parameters take a snapshot
- snapshot placed twice in priority queue
  - true case
  - false case

# Continuation-Based (cont)

- take first from priority queue to proceed
- when any path completes, take first from  $q$
- $q$  ordered by integral of domain over PDF
- many paths may be in partial execution
- by construction, paths complete in frequency order

# Monte Carlo

- generate a random point in the input space
- check that not in a domain already found
- execute, when loop/if add to path, returning with correct true/false

# Metaprogramming

- Smalltalk supports 4 key technologies:
  - Dynamic types
  - First-class booleans
  - Continuation capture
  - Dynamic code generation



# Dynamic Types

- because there are no “native” types can pass parameter types in place of e.g. numbers
- no change whatsoever required to source
- can trace deeply into library classes/methods

# First-class Booleans

- conceptually true/false are not special, they are simply objects
- Maybe can extend Boolean so that use in loop/if can capture information
- in practice, use sends #mustBeBoolean which can capture information

# Continuation Capture

- continuation is the information necessary to execute the rest of the program
- derived from Seaside continuation capture
- walks stack saving local variables
- first-class

# Dynamic Code Generation

- need to check complex predicates for inclusion
- could interpret the predicates
- not essential, but generating code for the check can speed up the implementation

# Conclusion

- Smalltalk provides the metaprogramming necessary to implement this analysis seamlessly
- may be the only one with all these features
- questions?