Roel Wuyts
Université Libre de Bruxelles

Benevol, May 26th 2005 (Eindhoven, The Netherlands)
Type Reconstruction

Context: program understanding in dynamically typed languages
- e.g. extraction of class diagrams

Type Reconstruction
- input: program without types
- output: program with types
Trade-offs

Precision vs. efficiency

- We chose efficiency for usage in a development browser
- Use Heuristics as basis for the reconstruction
- instead of full reconstruction
Heuristics

- Direct sends to instance variable
- Indirect sends to instance variables (getter methods)
- Direct assignment expressions
- Indirect assignment expressions (setter methods)
- (Type snooping)
Implementations

Using LiCoR (Library for Code Reasoning) in SOUL

- on the parse tree
- average: 500 milliseconds / instance variable
- more elaborate and easier to extend

Using partial evaluation on the byte code

- average: 30 milliseconds / class
Demo

Smalltalk.Core defineClass: #Point
  superclass: #{Core.ArithmeticValue}
  indexedType: #none
  private: false
  instanceVariableNames: 'x y '
  classInstanceVariableNames: "
  imports: "
  category: 'Graphics-Geometry'

Class: Core.Point
Parcel: none
Package: Graphics-Geometry
Conclusions & Future Work

Works

- About 80% of correctness on built-in libraries
- Better on domain-specific code

Future Work

- Fix About
- Will do this on (untyped) Java code and compare results