History

- 1.0 Feb 16, 2004 Formulation of XTA with GRS.
GRS-based Control-Flow Analysis

Uwe Aßmann
RISE
Linköpings Universitet
Remark

► When looking at the set constraints of CFA algorithms, it is simple to see that they can be turned into Edge Addition Rewrite Systems:
  ► Turn set variables into relations
  ► For global set variables, use a global source object, i.e., prog
  ► Interpret the subsetting as rewriting addition
► Then, EARS can be constructed for every constraint.
  ► They are congruent, i.e., give a unique solution as the set constraints
Name-based Resolution (RA)

- main in R
- For each method M with a virtual call e.m() and each method M' with name m:
  - (M in R) => (M' in R)

Clearly linear!
Class Hierarchy Analysis (CHA)

- main in R
- For each method M with a virtual call e.m(), each class C in Subtypes(StaticType(e)) with StaticLookup(C,m)=M':
  - (M in R) => (M' in R)

Still linear!
Rapid Type Analysis (RTA)

- main in R
- For each method $M$ with a virtual call $e.m()$, each class $C$ in $\text{Subtypes(StaticType}(e))$ with $\text{StaticLookup}(C,m)=M'$:
  - $(M \text{ in } R) \land (C \text{ in } S) \Rightarrow (M' \text{ in } R)$

Still linear!
EXtended Type Analysis (XTA)

- main in R
- For each method M with a virtual call e.m(), each class C in Subtypes(StaticType(e)) with StaticLookup(C,m)=M':
  - (M in R) and (C in S_M) => (M' in R) and .....
EXtended Type Analysis (XTA) Part II

- Field rules

Diagram:
- Node `prog` connected to `M:Method`
- `M:Method` connected to `x.write()`
- `x.write()` connected to `C:Class`
- Node `C:Class` connected to `C':Class`
- Node `prog` connected to `M:Method`
- `M:Method` connected to `x.write()`
- `x.write()` connected to `C:Class`
- Node `C:Class` connected to `Type`
Results

- Fast CFA algorithms can be represented by EARS
- Open: Can we derive their complexity automatically, by looking at the left hand sides of the rules?
The Common Core

- Datalog
- GRS
- Chain EARS
- CFA
- Subset Constraints
- Program Analysis
- IDFS
- Chain Datalog