Extending Attribute Grammars with Collection Attributes – Evaluation and Applications

Eva Magnusson *)

Torbjörn Ekman **)

Görel Hedin *)

***) Lund University, Sweden**

****) Oxford University**

Background

- Building tools using declarative programming
  - compilers, static program analysis,…
  - extensible, reusable implementations
- Support efficiency
  - capability to handle large programs

Motivation

- Attribute grammars
  - One equation for each attribute

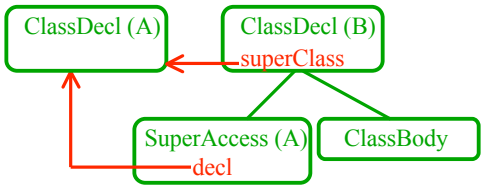
    a = f(…);

    used to define ”local” property
- Collection Attributes
  - Combination of properties of remote nodes

    a = initial;
    for each *contributing node*
    a.combine(contribution)
  - models whole program problems

Motivating example

class B extends A

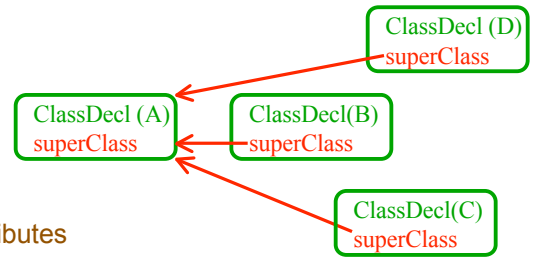


Attribute Grammar

ClassDecl.superClass =
getSuperAccess.decl ;

Motivating example

class B extends A
class C extends A
class D extends A



ClassDecl.subClasses = ..?..

- All extending classes contributes
- Can be practically anywhere in the AST

Results

- Declarative formalism for whole program problems

  ***coll** Set ClassDecl.subClasses [new Set()] **with** add;*

  *ClassDecl **contributes** this*
  ***to** ClassDecl.subClasses*
  ***for** superClass;*

- A number of algorithms for evaluation
- Implemented in our tool JastAdd
- Example applications
- Performance results

Evaluation Algorithms

- Naive
  - traverse AST for every demanded instance
- One-phase
  - evaluate all instances in one traversal
- Two-phase
  - one preparatory traversal, final computation on demand
- Additional variants of these, and also algorithms for circularly defined collection attributes

The Metrics Application

10k LOC

165 LOC
7 coll attrs
17 contribution decl

Chidamber and Kemerer's OO-metrics

- height of inheritance tree
- number of subclasses
- coupling between classes
- lack of cohesion
- …

Program size	Time (ms)	
	1Ph	2Ph
100k	2207	2585

Experimental results

Computation of subClasses

Size			Time			
Lines	Coll inst	Contr inst	Emul attrs	Naive	2-Ph	1-Ph
15k	77	77	6300	1600	60	50
36k	180	180	110000	15000	200	220

Collection attributes

- Specifications
  - whole program problems
  - concise
  - extensible
- Efficiency
  - outperform ”ordinary” attributes
  - can handle large practical applications