Profile-based Abstraction and Analysis of Attribute Grammar Evaluation

Anthony M. Sloane
Anthony.Sloane@mq.edu.au
@inkytonik

Programming Languages Research Group
Department of Computing
Macquarie University

June 25, 2012

Attribute grammar applications

- Java compilation (Ekman and Hedin, OOPSLA 2007)
- Protocol normalization (Davidson *et al.*, ESORICS, LNCS 5789, 2009)
- Image processing (Han and Zhu, IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009)
- XML integrity validation (Bouchou *et al.*, Database and Expert Systems Applications, LNCS 6860, 2011)
- Genotype-phenotype mapping (Karim and Ryan, Genetic Programming, LNCS 6621, 2011)

Kiama

A collection of Scala-based *internal domain-specific languages* for solving problems in the domain of language processing.

Problem	*Domain-specific language*
Structure representation	Algebraic data types
Text to structure	Parsing expression grammars
Output	Pretty printing
Transformation	Term rewriting
Decoration	Attribute grammars

PicoJava

```
{
    class A {
        int y;
        AA a;
        y = a.x;
        class AA {
            int x;
        }
        class BB extends AA {
            BB b;
            b.y = b.x;
        }
    }
}
```

Name analysis in PicoJava (1)

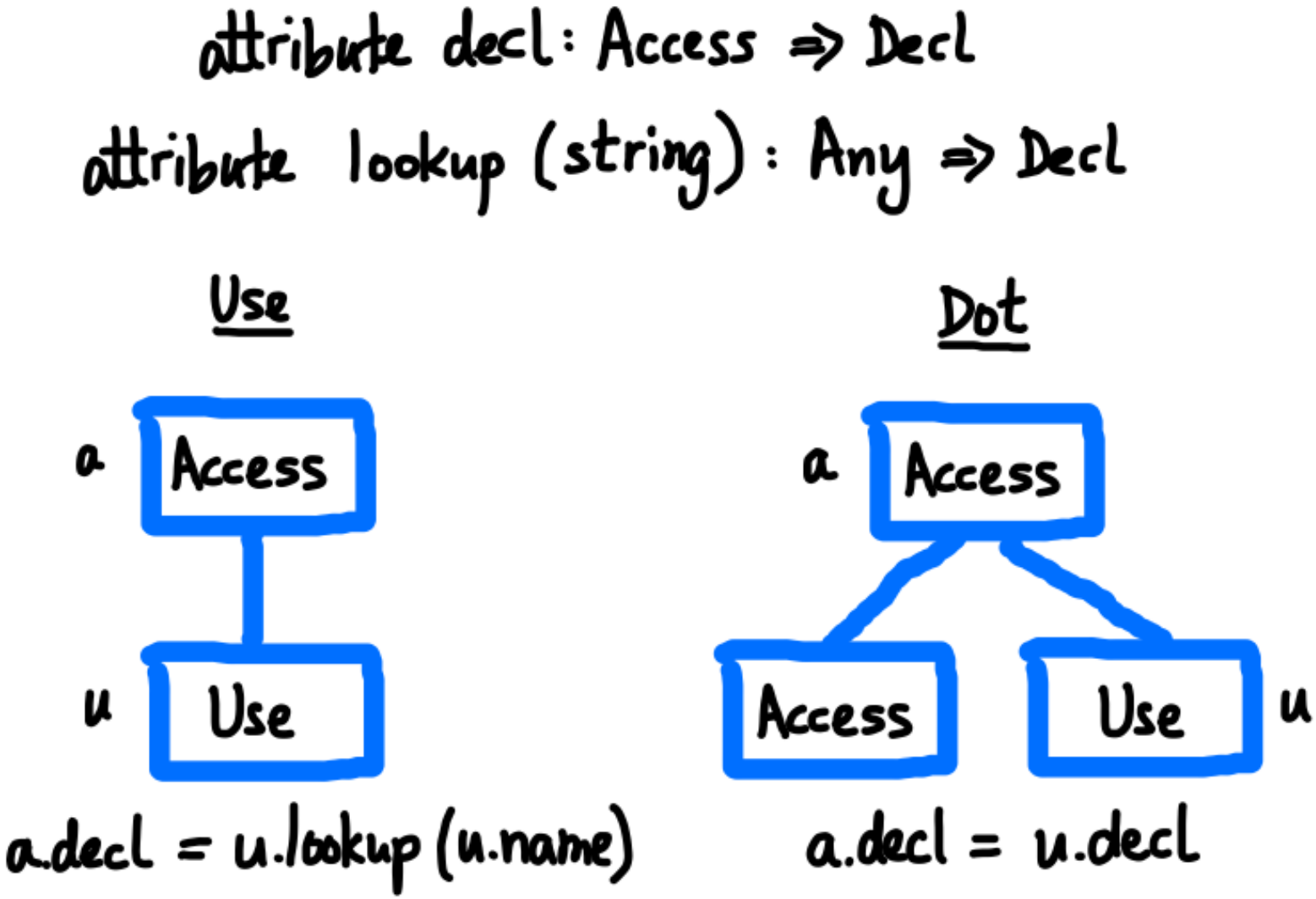


Figure: Declarations for name accesses

Name analysis in PicoJava (2)

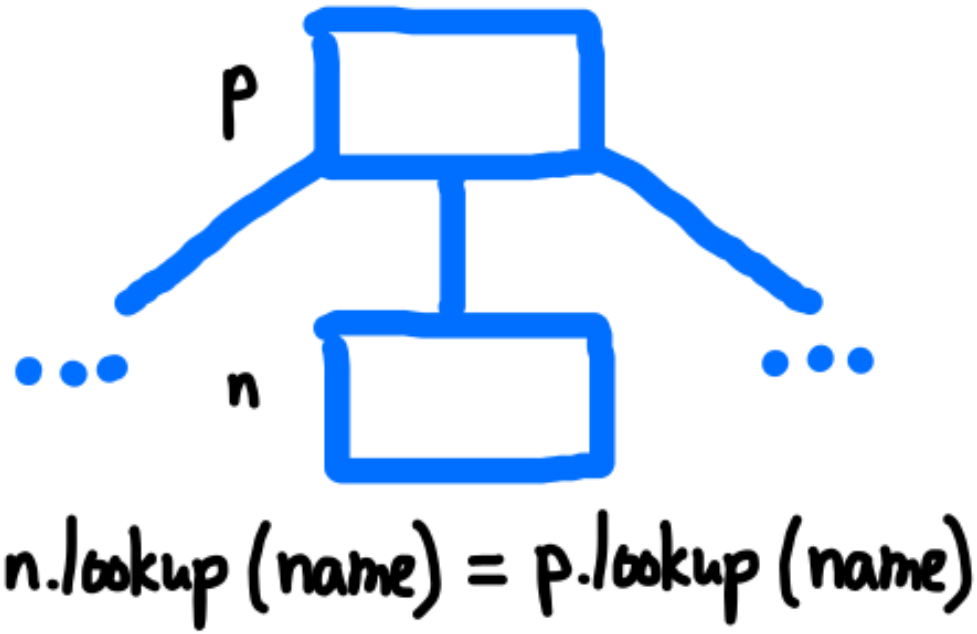


Figure: Declaration lookup: default case

Name analysis in PicoJava (3)

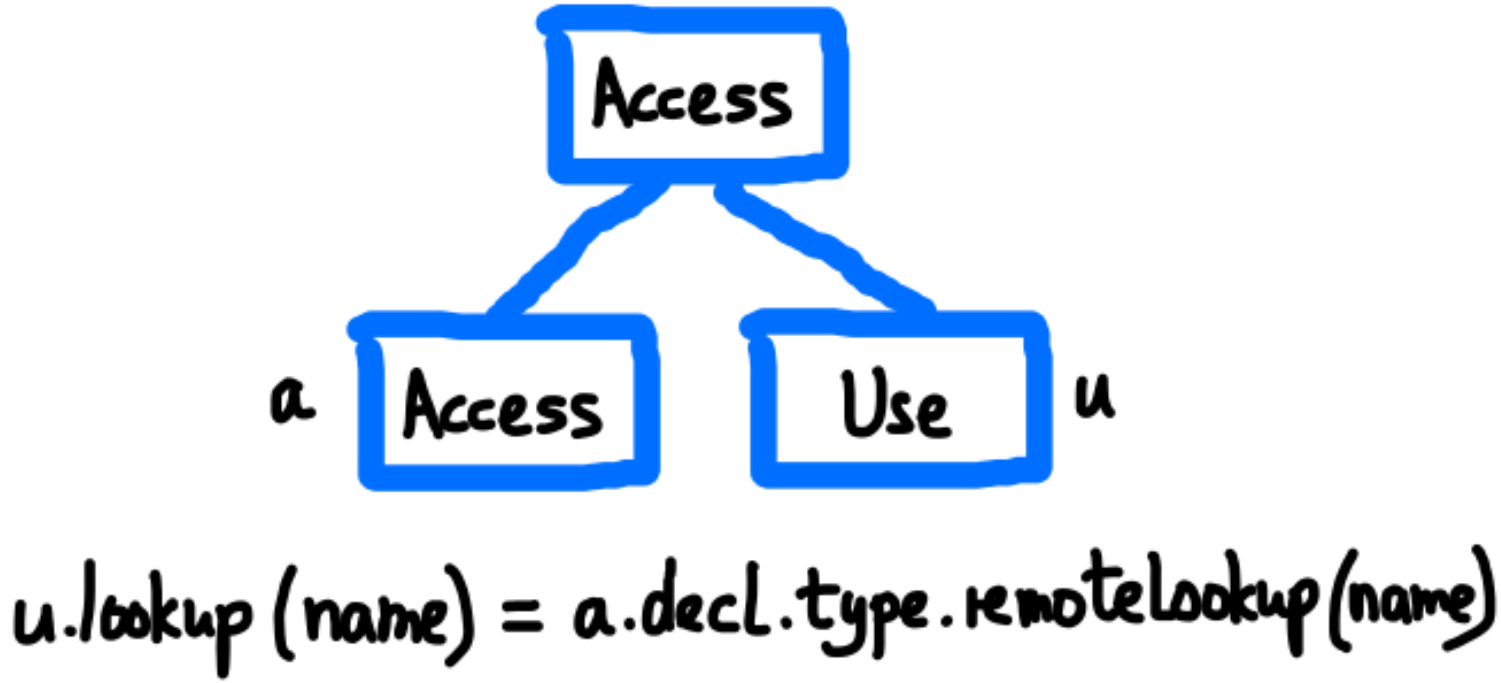


Figure: Declaration lookup: qualified access

Name analysis in PicoJava (4)

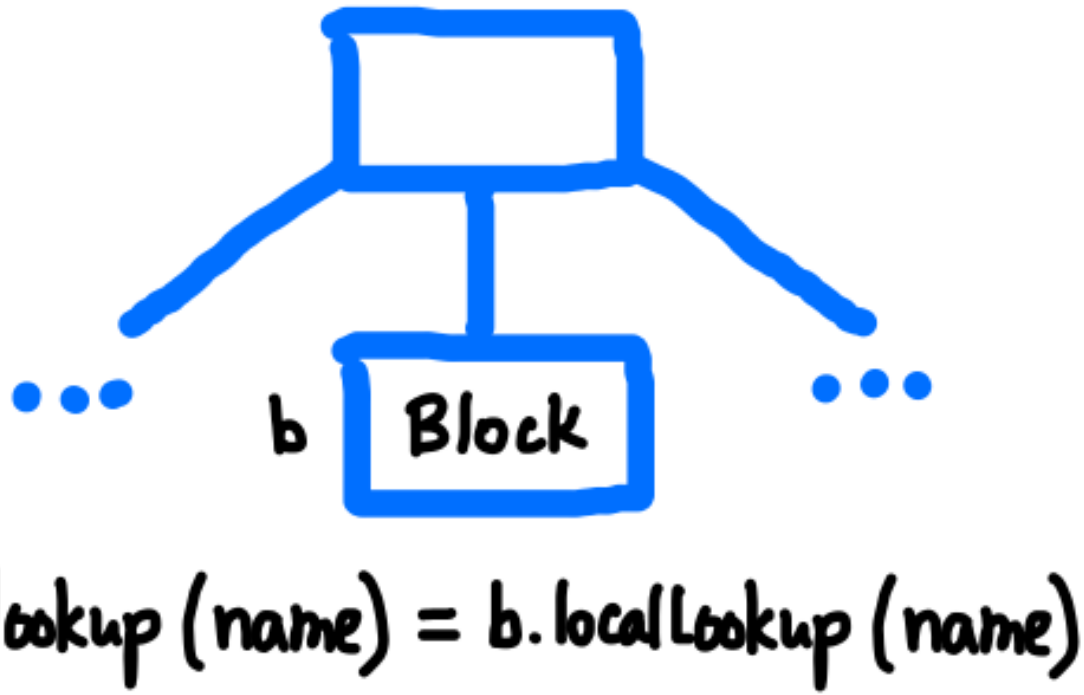
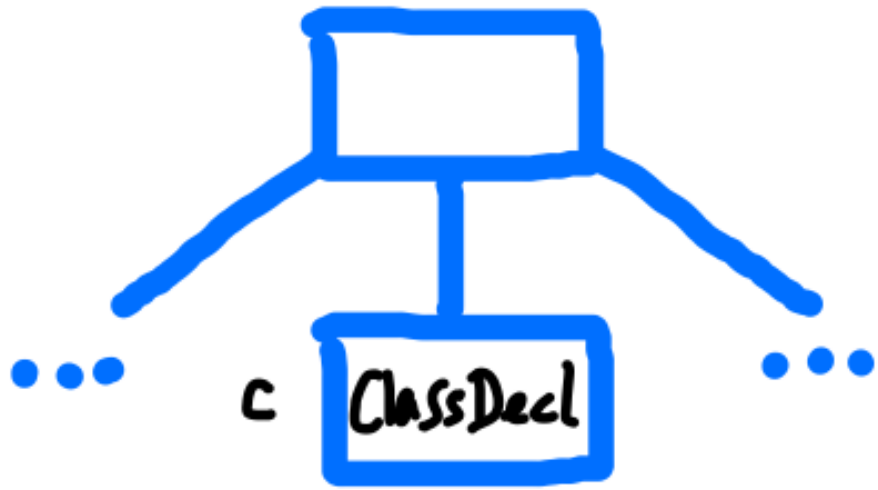


Figure: Declaration lookup: blocks

Name analysis in PicoJava (5)



$$c.lookup(name) = c.superClass.remoteLookup(name)$$

Figure: Declaration lookup: class declarations

Questions, questions, questions

- How many times does an attribute get evaluated?
- How much time do the evaluations take?
- Which values are computed?
- Which percentage of evaluations use a cached attribute value?
- Which parameters are passed to a parameterised attribute?
- At which types of nodes do the evaluations take place?
- Where are the evaluation nodes located in the tree?
- What patterns of attribute dependence occur?

Data collection

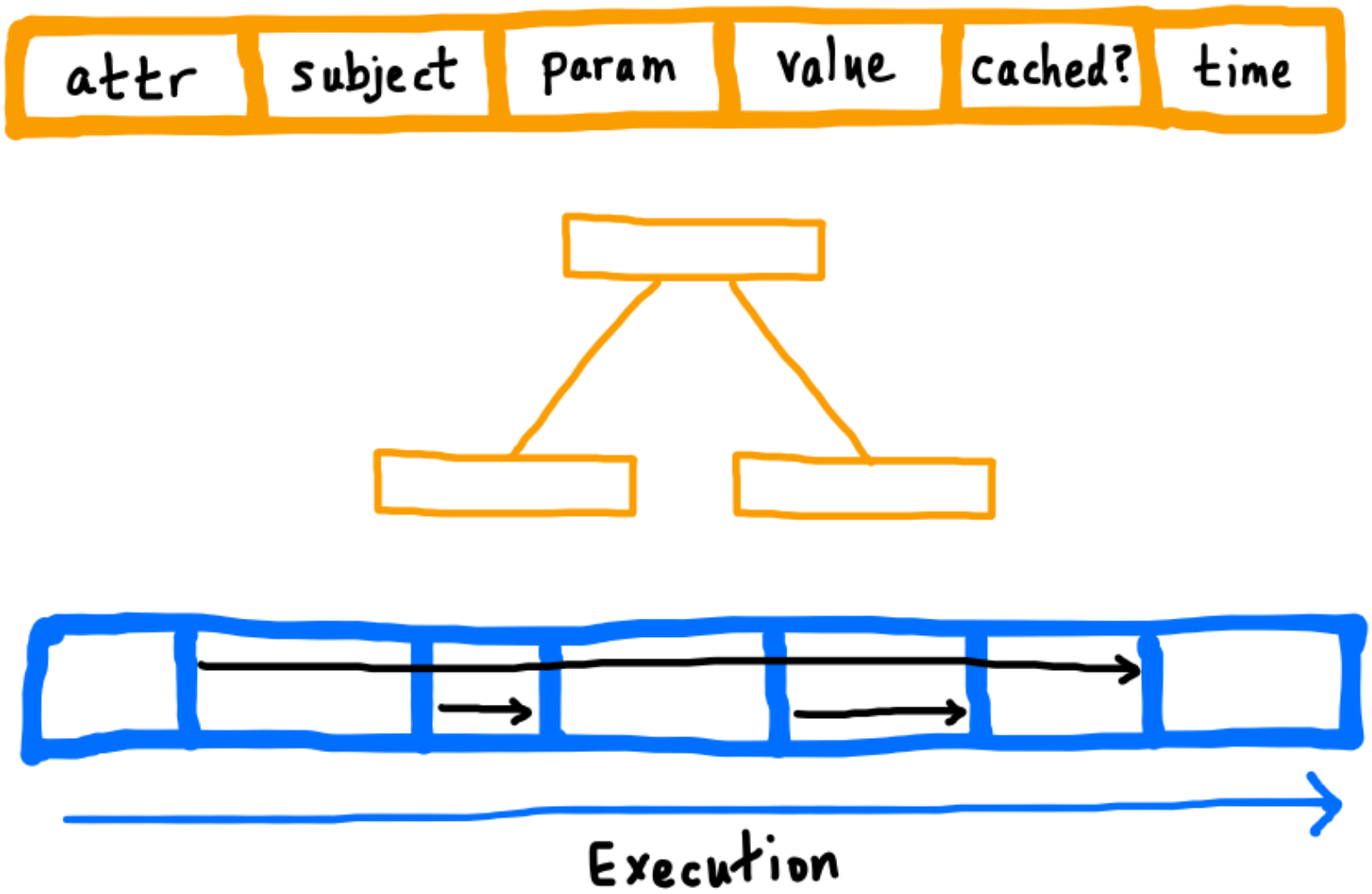


Figure: Event generation and execution model creation

Report writing

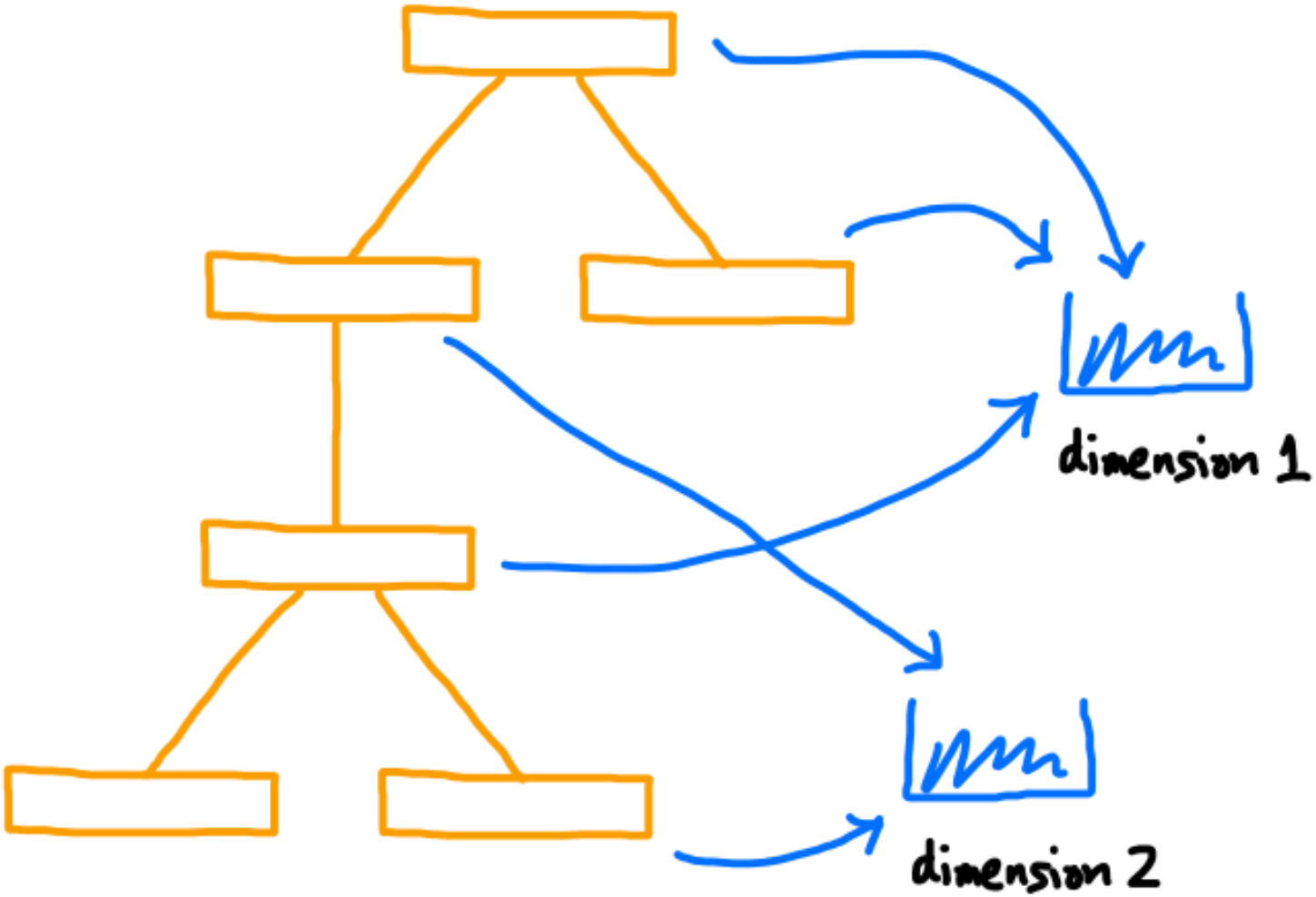


Figure: Aggregation by arbitrary dimension

Intrinsic dimension: attribute

```
   202 ms total time
    47 ms profiled time (23.5%)
   206 profile records

By attribute:

  Self   Self   Desc   Desc Count  Count
    ms      %     ms      %           %
    11  24.7     30  63.3     27  13.1  decl
    10  22.8     19  40.9     32  15.5  lookup
     9  19.0      6  13.9     19   9.2  localLookup
     0   1.8      4   8.8     18   8.7  unknownDecl
     4   8.9      0   0.0     33  16.0  declarationOf
     2   4.8      1   3.1      7   3.4  remoteLookup
     2   4.9      0   1.6      2   1.0  isSubtypeOf
     1   2.7      1   2.3     20   9.7  tipe
                    ...
```

Intrinsic dimension: value

```
By value for decl:

  Self   Self   Desc   Desc
    ms      %     ms      %
    12   23.5     15   29.7  PrimitiveDecl(int)
     0    0.1      1    2.4  [1]
     0    0.3      0    1.4  VarDecl(Use(int),x)
     0    0.6      0    0.5  [2]
     0    0.1      0    0.7  UnknownDecl(unknown)
     0    0.4      0    0.0  VarDecl(Use(BB),b)
     0    0.2      0    0.2  VarDecl(Use(AA),a)
     0    0.2      0    0.1  VarDecl(Use(int),y)
```

Intrinsic dimensions: attribute and subject node type

By type for lookup:

Self ms	Self %	Desc ms	Desc %	Count	Count %	
7	13.2	26	47.7	12	5.8	Use
1	3.2	15	28.6	4	1.9	VarDecl
2	4.2	2	4.1	5	2.4	Block
0	1.1	1	2.6	4	1.9	ClassDecl
0	0.7	0	1.5	3	1.5	Dot
0	0.8	0	1.3	4	1.9	AssignStmt

Intrinsic dimensions: attribute and parameter

```
By parameter for localLookup:

  Self   Self   Desc   Desc Count  Count
    ms      %     ms      %           %
     4   11.2      4   11.0      5    2.4  Some(int)
     2    5.9      1    3.8      1    0.5  Some(unknown)
     0    1.0      0    0.4      3    1.5  Some(BB)
     0    0.7      0    0.1      3    1.5  Some(y)
     0    0.6      0    0.1      3    1.5  Some(x)
     0    0.5      0    0.0      2    1.0  Some(AA)
     0    0.2      0    0.0      1    0.5  Some(a)
     0    0.2      0    0.0      1    0.5  Some(b)
```

Intrinsic dimensions: attribute, parameter and cache usage

```
By cached for lookup and Some(int):

  Self   Self   Desc   Desc Count  Count
    ms      %     ms      %              %
     4   10.2      0    0.5      9    4.4   false
     0    0.1      0    0.0      1    0.5   true

By cached for lookup and Some(a):

  Self   Self   Desc   Desc Count  Count
    ms      %     ms      %              %
     0    0.7      0    0.0      3    1.5   false
```

Derived dimension: subject node location in tree

```
By location for env.in:

 Self   Self   Desc   Desc Count  Count
   ms      %     ms      %           %
   52    7.5    132   18.9  3237   10.0  Leaf
   44    6.3     89   12.8  5455   16.8  Inner
   58    8.3      0    0.0    81    0.3  Root
```

Derived dimension: direct attribute dependence (1)

```
By depends-on for unknownDecl:

Count  Count
            %
    3    1.5  ClassDecl(^).unknownDecl
    3    1.5  Block(^).unknownDecl
    1    0.5  Program(^).unknownDecl
    1    0.5  Program(@).localLookup
   10    4.9
```

Derived dimension: direct attribute dependence (2)

```
By depends-on for idntype:

Count  Count
          %
  754    2.3  IdnUse(@).entity, NamedType(?).deftype
  155    0.5  IdnUse(@).entity
   13    0.0  IdnUse(@).entity, IntExp(?).tipe
   22    0.1  IdnUse(@).entity,
                 RecordTypeDef(?).deftype
    4    0.0  IdnUse(@).entity, AddExp(?).tipe
   24    0.1  IdnUse(@).entity,
                 ArrayTypeDef(?).deftype
    1    0.0  IdnUse(@).entity, DivExp(?).tipe
    1    0.0  IdnUse(@).entity, IdnExp(?).tipe
```

Derived dimension: indirect attribute dependence

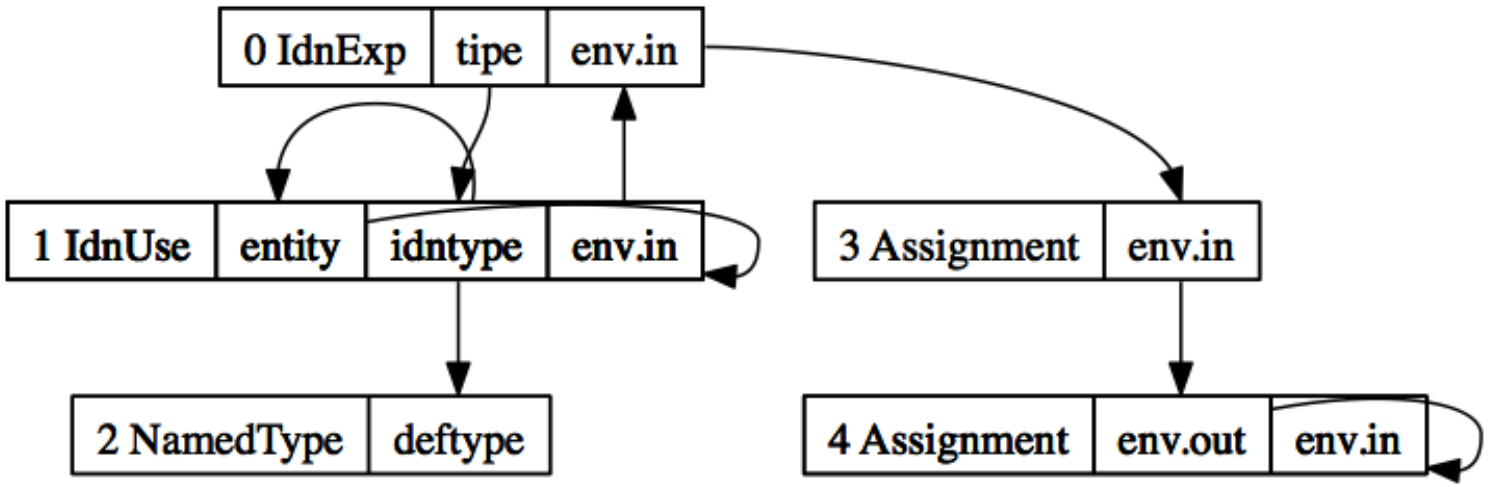


Figure: Attribute dependence graph

Conclusion

- A practical extension of execution profiling to a high-level programming domain.
- A general model of domain-specific events and aggregation that should apply to more domains.
- Experience with intrinsic and derived attribute dimensions that enhance our ability to understand how an attribute evaluation process works.

More Information about Kiama

- `http://kiama.googlecode.com/`

- Supported by Macquarie University, The Netherlands NWO, TU Eindhoven, TU Delft, Yourkit, LLC, and Cloudbees, Inc.