

Tickling Java With a Feather



Tristan Allwood
(tristan.allwood@imperial.ac.uk)

Susan Eisenbach
(s.eisenbach@imperial.ac.uk)

The Problem

The image shows two browser windows side-by-side. The left window displays the Sun Developer Network (SDN) bug database search results for 'java bug parade'. The right window displays the Haskell GHC report page for 'ghc'.

Left Window: Sun Developer Network (SDN) Bug Database Search

238 Results Returned, Sorted by Date | [Sort by Relevance](#) » [Previous 11 - 20 Next](#) » Bug Database Search

Bug ID: 6569789 Compiler test lang/TYPE/type153/type15304/type15304.html fails since jdk7 b05
java:compiler, Compiler test lang/TYPE/type153/type15304/type15304.html fails since jdk7 b05, State: In progress, bug, Reported: 14-JUN-2007, Reported against: b05
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569789 - 14-JUN-2007

Bug ID: 6569806 Negative lang/CLSS/class411/class41104/class41104a.html#class41104src fails since jdk7 b03
java:compiler, Negative lang/CLSS/class411/class41104/class41104a.html#class41104src fails since jdk7 b03, State: In progress, bug, Reported: 14-JUN-2007, Reported against: b03
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569806 - 14-JUN-2007

Bug ID: 6480588 No way to suppress deprecation warnings when implementing deprecated interface
java:compiler, No way to suppress deprecation warnings when implementing deprecated interface, State: In progress, bug, Reported: 11-OCT-2006, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6480588 - 13-JUN-2007

Bug ID: 6569057 Generics regression
java:compiler, Generics regression, State: In progress, bug, Reported: 13-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569057 - 13-JUN-2007

Bug ID: 6569074 Type inference fails with bounded type parameters
java:compiler, Type inference fails with bounded type parameters, State: In progress, bug, Reported: 13-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569074 - 13-JUN-2007

Bug ID: 6569079 Deprecation warnings for use within deprecated entities
java:compiler, Deprecation warnings for use within deprecated entities, State: In progress, bug, Reported: 13-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569079 - 13-JUN-2007

Bug ID: 6569404 Cannot instantiate an inner class of a type variable
java:compiler, Cannot instantiate an inner class of a type variable, State: In progress, bug, Reported: 13-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569404 - 13-JUN-2007

Bug ID: 6569091 javac 6 incompatibility or javac 5 bug
java:compiler, javac 6 incompatibility or javac 5 bug, State: In progress, bug, Reported: 13-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6569091 - 13-JUN-2007

Bug ID: 6568599 Failure of inference with generics
java:compiler, Failure of inference with generics, State: In progress, bug, Reported: 12-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6568599 - 12-JUN-2007

Bug ID: 6568290 KSL: problem updating resource bundles on Mac
java:compiler, KSL: problem updating resource bundles on Mac, State: In progress, bug, Reported: 11-JUN-2007, Reported against:
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6568290 - 11-JUN-2007

238 Results Returned, Sorted by Date | [Sort by Relevance](#) » [Previous 11 - 20 Next](#) »



































Right Window: Haskell GHC Report

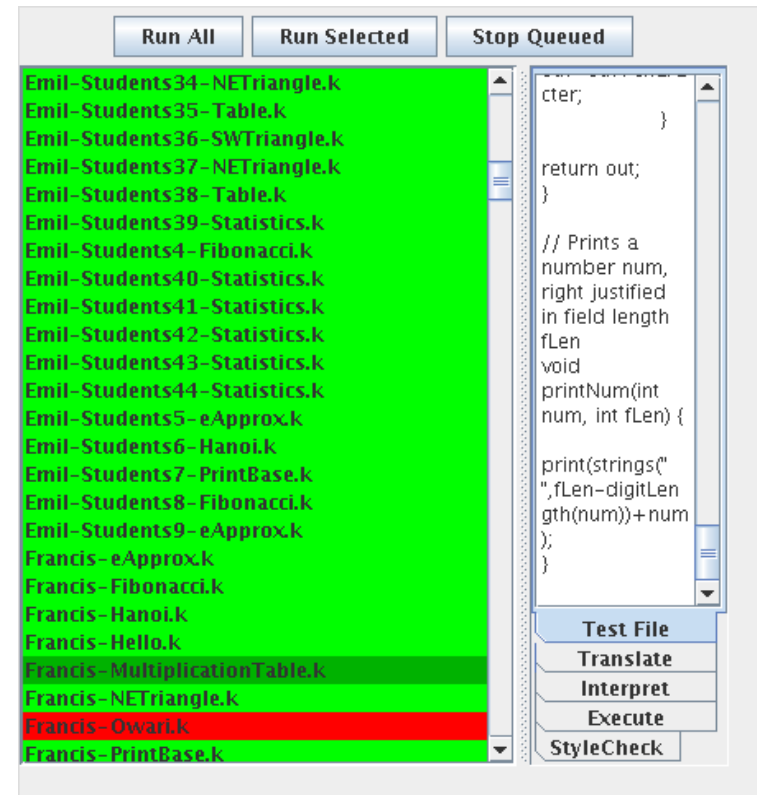
#	Issue	Component	Version	Priority	Severity	Platform	Assignee	Created
#1087	bang patterns with infix ops	Compiler (Parser)	6.6	normal	normal	_ _		01/07/07
#314	#line pragmas not respected inside nested comments	Compiler (Parser)	6.4	low	normal	6.8	simonmar *	02/07/07
#459	Bad parse error message	Compiler (Parser)	6.4.1	low	minor	6.8	simonmar	09/07/07
#1215	GHC fails to respect the maximal munch rule while lexing "qualified reservedids"	Compiler (Parser)	6.6	low	normal	6.8	simonmar	03/07/07
#984	Syntax error shows in the wrong position	Compiler (Parser)	6.4.2	low	minor	_ _		11/07/07
#999	Misattributed parse error in do block	Compiler (Parser)	6.4.2	low	normal	_ _		11/07/07
#1406	Constraint doesn't reduce in the presence of quantified type variables	Compiler (Type checker)	6.6.1	normal	normal			06/07/07
#344	double-panic with GADTs	Compiler (Type checker)	6.4	normal	normal	6.8	simonpj *	04/07/07
#345	GADT - fundep interaction	Compiler (Type checker)	6.4	normal	normal	6.8	simonpj *	04/07/07
#451	GHC poor type-checker error message	Compiler (Type checker)	6.4	normal	normal	6.8	simonpj	09/07/07
#816	Weird fundep behavior (with -fallow-undecidable-instances)	Compiler (Type checker)	6.4.2	normal	normal	6.8	simonpj	07/07/07
#919	Cryptic type error message (should be syntax error)	Compiler (Type checker)	6.5	normal	normal	6.8		10/07/07
#1220	Newtype deriving should only work if superclasses are newtype-derived	Compiler (Type checker)	6.6	normal	normal	6.8		03/07/07
#1241	Functional dependency Coverage Condition is lifted, and should not be	Compiler (Type checker)	6.6	normal	normal	6.8		03/07/07
#1330	Church2 test gives a rather confusing error with the HEAD	Compiler (Type checker)	6.7	normal	normal	6.8		05/07/07
#589	Various poor type error messages	Compiler (Type checker)	6.4.1	low	minor	6.8		12/07/07
#714	inconsistency between handling of class and signature constraints	Compiler (Type checker)	6.5	low	normal	6.8	simonpj	03/07/07
#1221	Types don't match expressions in type error	Compiler (Type checker)	6.6	low	normal	6.8		03/07/07
#565	overlapping instances & fundeps broken	Compiler (Type checker)	5.0.0	low	normal	_ _	nobody *	07/07/07
#323	Exponential behaviour with type synonyms	Compiler (Type checker)	6.4.1	low	normal	_ _	simonpj *	03/07/07
#1050	Using an inferred type as a type signature fails	Compiler (Type checker)	6.6	low	minor	_ _		12/07/07
#959	Debugging info(?) leaks out: "Urk! Inventing strangely-kinded void TyCon"	Compiler (Type checker)	6.6	lowest	minor	_ _		10/07/07

8/8 Top | [set guioptions=mT](#) | <http://bugs.sun.com/bugdatabase/search.do?process=1&category=java&subcategory=compiler&bugStatus=open&type=b>

9/9 54% | <http://hackage.haskell.org/trac/ghc/report/1?sort=component&asc=1> | [set guioptions=mT](#)

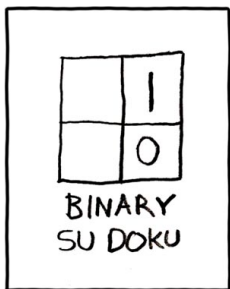
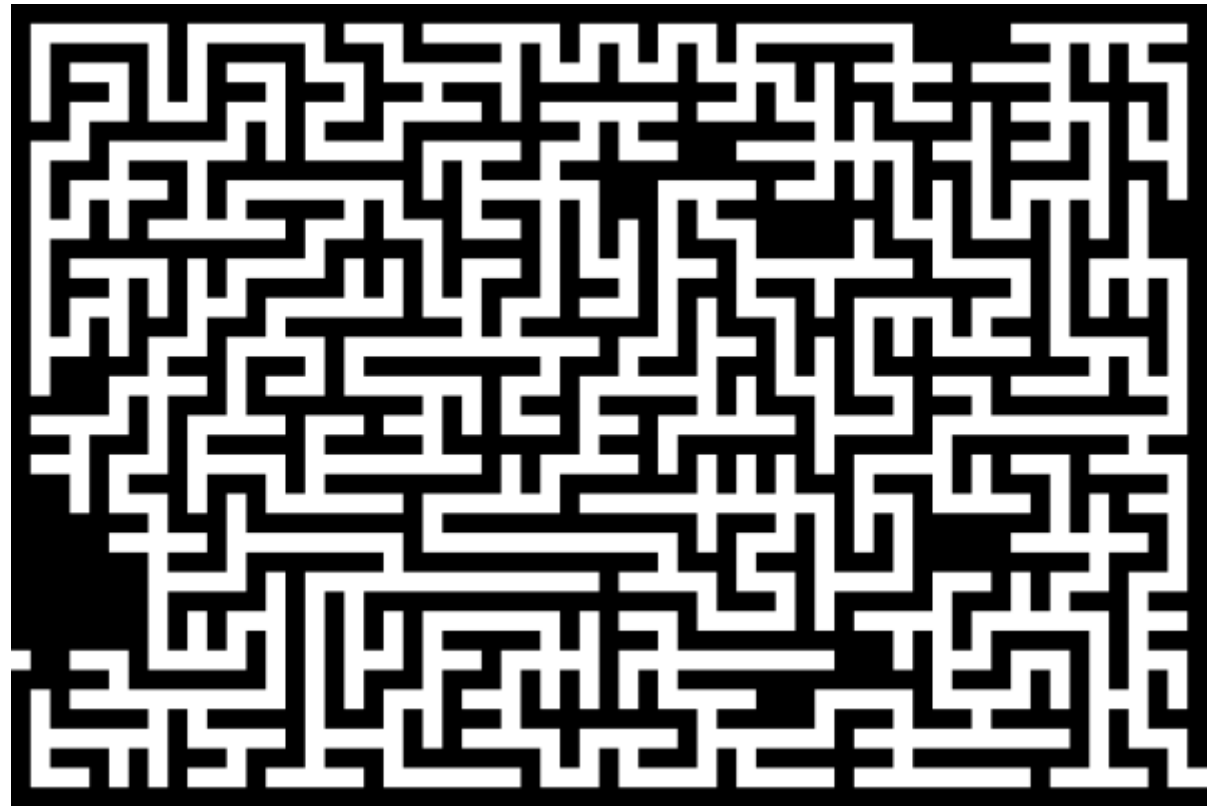
The Problem

	tcfail043.stderr-hugs	12-Nov-2004 03:56	125
	tcfail044.hs	21-Jun-2001 04:16	477
	tcfail044.stderr	01-Apr-2007 14:18	488
	tcfail044.stderr-hugs	12-Nov-2004 03:56	69
	tcfail046.hs	21-Jun-2001 04:16	725
	tcfail046.stderr	21-Jun-2007 09:34	556
	tcfail046.stderr-hugs	12-Nov-2004 03:56	97
	tcfail047.hs	21-Jun-2001 04:16	95
	tcfail047.stderr	01-Apr-2007 14:18	247
	tcfail047.stderr-hugs	12-Nov-2004 03:56	75
	tcfail048.hs	21-Jun-2001 04:16	65
	tcfail048.stderr	28-Nov-2005 07:44	63
	tcfail048.stderr-hugs	12-Nov-2004 03:56	45
	tcfail049.hs	21-Jun-2001 04:16	35
	tcfail049.stderr	28-Nov-2005 07:44	37
	tcfail049.stderr-hugs	12-Nov-2004 03:56	48
	tcfail050.hs	21-Jun-2001 04:16	35
	tcfail050.stderr	28-Nov-2005 07:44	54
	tcfail050.stderr-hugs	12-Nov-2004 03:56	56
	tcfail051.hs	21-Jun-2001 04:16	58
	tcfail051.stderr	28-Nov-2005 07:44	127
	tcfail051.stderr-hugs	12-Nov-2004 03:56	45
	tcfail052.hs	21-Jun-2001 04:16	42
	tcfail052.stderr	28-Nov-2005 07:44	52
	tcfail052.stderr-hugs	12-Nov-2004 03:56	53
	tcfail053.hs	21-Jun-2001 04:16	34
	tcfail053.stderr	28-Nov-2005 07:44	64
	tcfail053.stderr-hugs	12-Nov-2004 03:56	56
	tcfail054.hs	21-Jun-2001 04:16	40
	tcfail054.stderr	28-Nov-2005 07:44	54
	tcfail054.stderr-hugs	12-Nov-2004 03:56	56
	tcfail055.hs	21-Jun-2001 04:16	55
	tcfail055.stderr	22-Nov-2006 04:33	200
	tcfail055.stderr-hugs	12-Nov-2004 03:56	134
	tcfail056.hs	21-Jun-2001 04:16	200



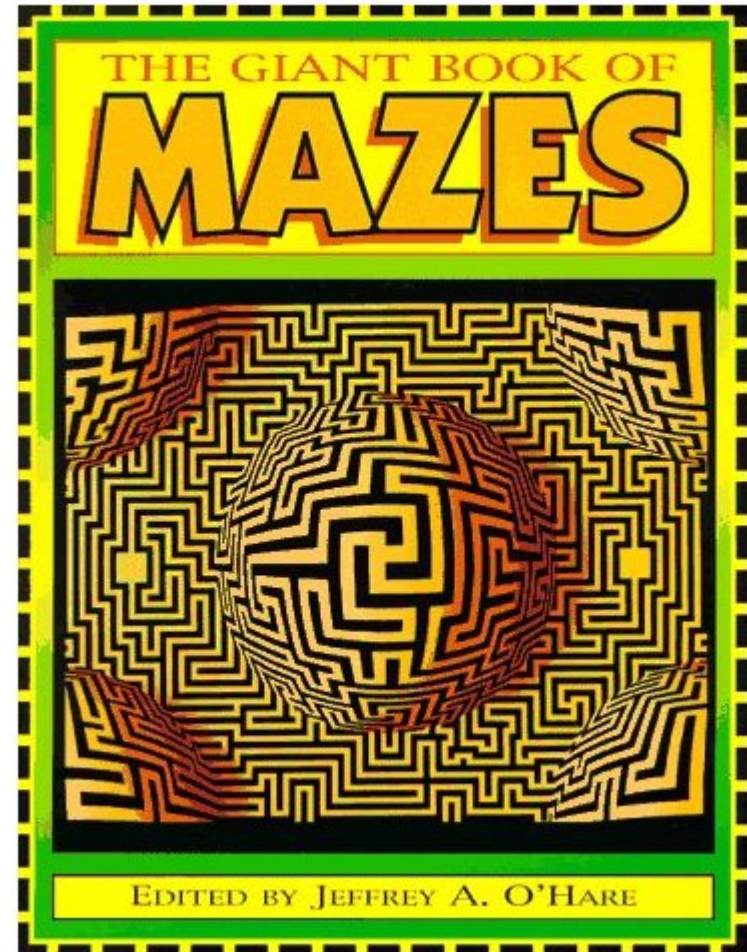
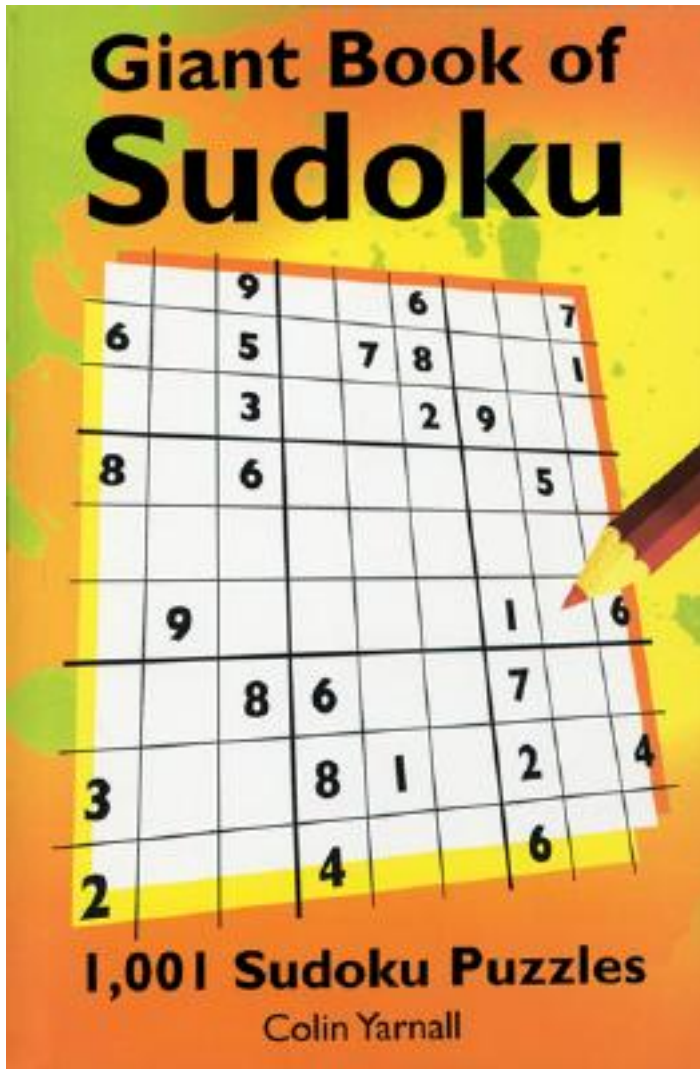
The Problem

		1						
		2		3				4
			5			6		7
5			1	4				
	7						2	
				7	8			9
8		7			9			
4				6		3		
						5		



(www.xkcd.com)

The Problem



Semantics To The Rescue

```
@Article{Igarashi-Pierce-Wadler01,
  author = "Atsushi Igarashi and
    Benjamin Pierce and Philip
    Wadler",
  title = "{Featherweight Java}: {A}
    Minimal Core Calculus for
    {Java} and {GJ}",
  month = may,
  journal = "TOPLAS",
  volume = "23",
  number = "3",
  pages = "396--459",
  year = "2001",
}
```

Expression typing:

$$\frac{}{\Gamma \vdash x : \Gamma(x)} \quad (\text{T-VAR})$$

$$\frac{\Gamma \vdash e_0 : C_0 \quad \text{fields}(C_0) = \bar{C} \bar{f}}{\Gamma \vdash e_0.f_i : C_i} \quad (\text{T-FIELD})$$

$$\frac{\Gamma \vdash e_0 : C_0 \quad \text{mtype}(m, C_0) = \bar{D} \rightarrow C \quad \Gamma \vdash \bar{e} : \bar{C} \quad \bar{C} \triangleleft \bar{D}}{\Gamma \vdash e_0.m(\bar{e}) : C} \quad (\text{T-INVK})$$

$$\frac{\text{fields}(C) = \bar{D} \bar{f} \quad \Gamma \vdash \bar{e} : \bar{C} \quad \bar{C} \triangleleft \bar{D}}{\Gamma \vdash \text{new } C(\bar{e}) : C} \quad (\text{T-NEW})$$

$$\frac{\Gamma \vdash e_0 : D \quad D \triangleleft C}{\Gamma \vdash (C)e_0 : C} \quad (\text{T-UCAST})$$

$$\frac{\Gamma \vdash e_0 : D \quad C \triangleleft D \quad C \neq D}{\Gamma \vdash (C)e_0 : C} \quad (\text{T-DCAST})$$

$$\frac{\Gamma \vdash e_0 : D \quad C \not\triangleleft D \quad D \not\triangleleft C \quad \text{stupid warning}}{\Gamma \vdash (C)e_0 : C} \quad (\text{T-SCAST})$$

Method typing:

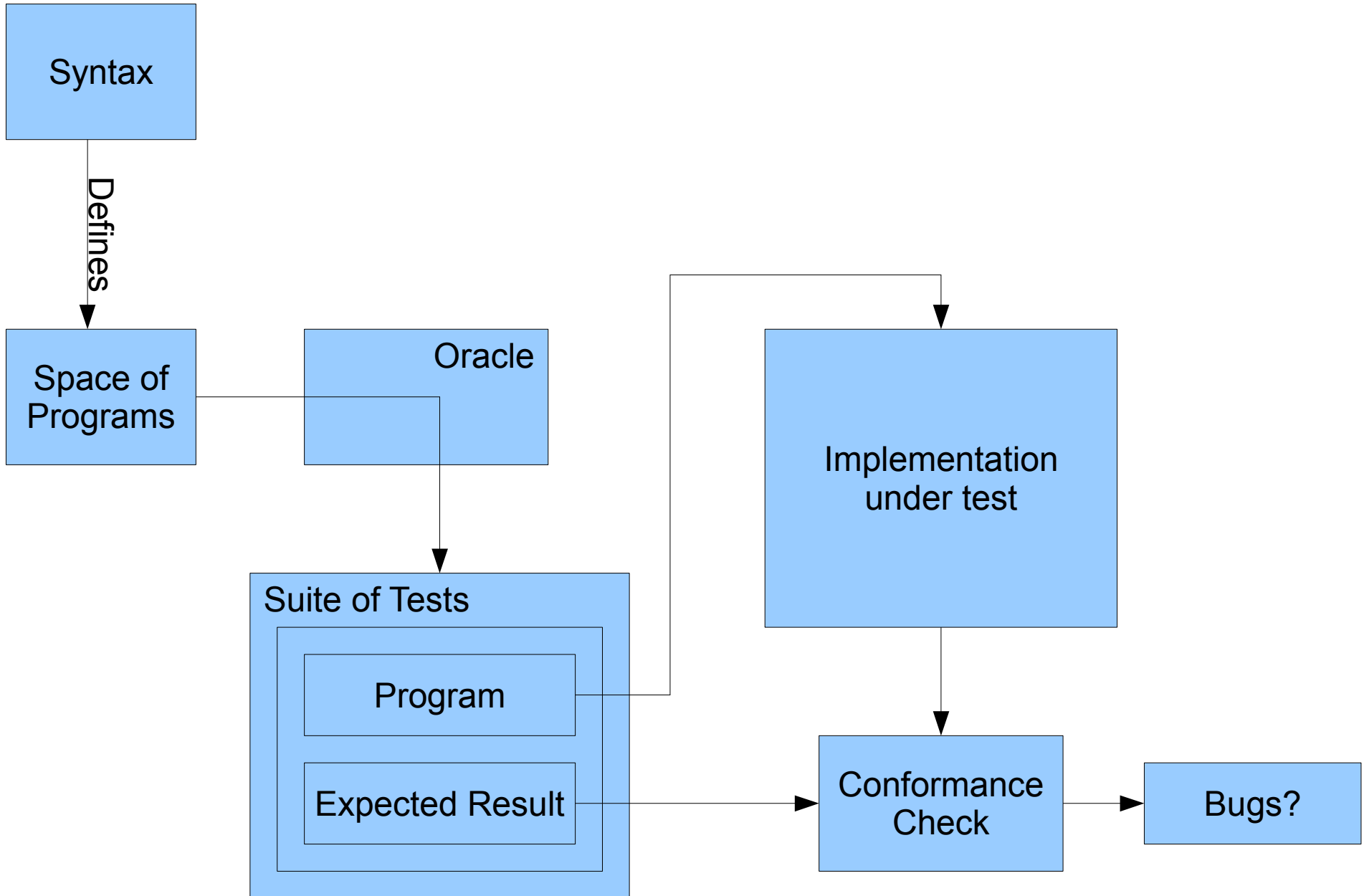
$$\frac{\bar{x} : \bar{C}, \text{this} : C \vdash e_0 : E_0 \quad E_0 \triangleleft C_0 \quad \text{class } C \text{ extends } D \{ \dots \} \quad \text{if } \text{mtype}(m, D) = \bar{D} \rightarrow D_0, \text{ then } \bar{C} = \bar{D} \text{ and } C_0 = D_0}{C_0 \text{ m}(\bar{C} \bar{x}) \{ \text{return } e_0; \} \text{ OK IN } C} \quad (\text{T-METHOD})$$

Class typing:

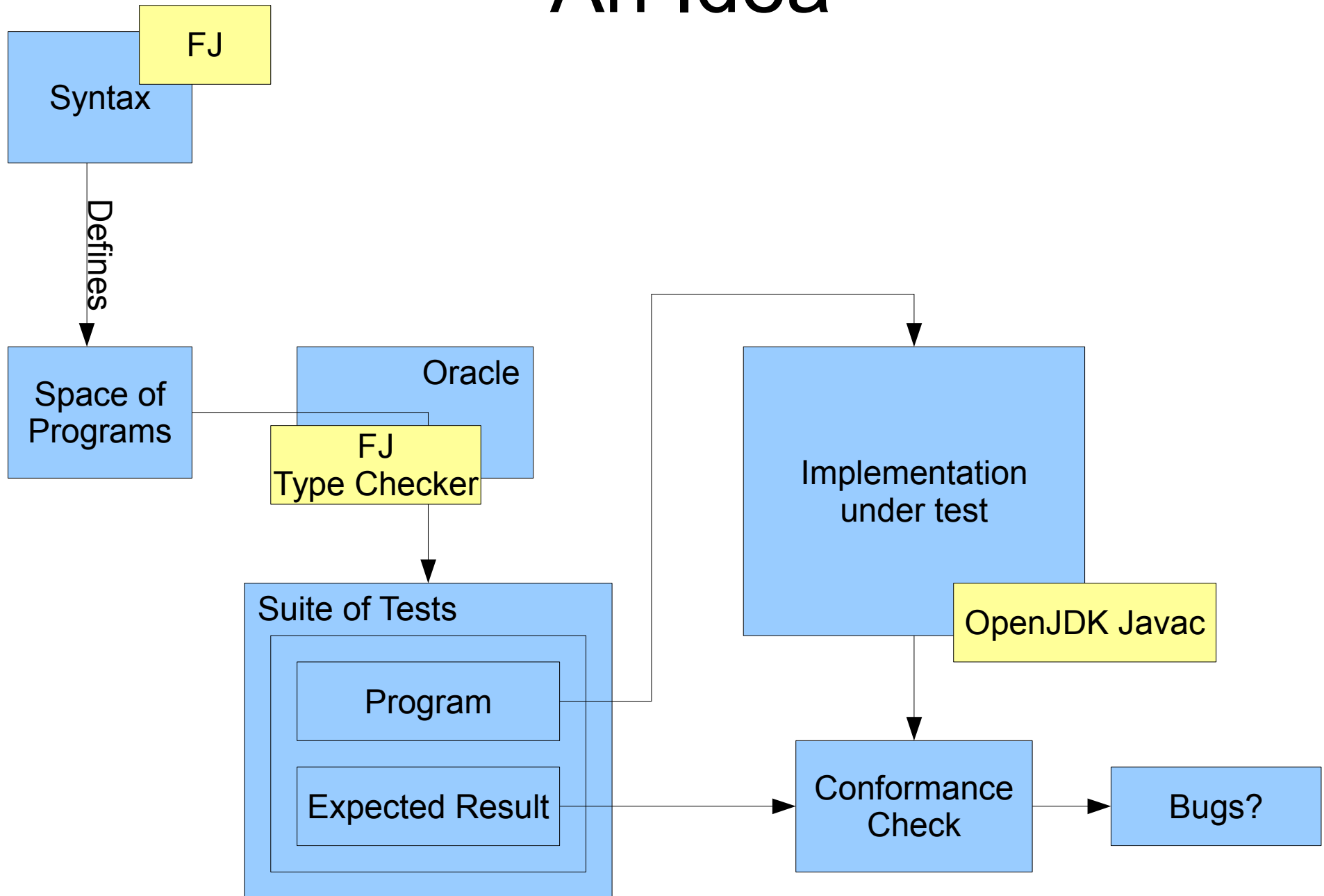
$$\frac{K = C(\bar{D} \bar{g}, \bar{C} \bar{f}) \{ \text{super}(\bar{g}); \text{this.f} = \bar{f}; \} \quad \text{fields}(D) = \bar{D} \bar{g} \quad \bar{M} \text{ OK IN } C}{\text{class } C \text{ extends } D \{ \bar{C} \bar{f}; K \bar{M} \} \text{ OK}} \quad (\text{T-CLASS})$$

Fig. 2. FJ: Typing rules.

An Idea



An Idea



Syntax

Syntax:

$L ::= \text{class } C \text{ extends } C \{ \bar{C} \bar{f}; K \bar{M} \}$

$K ::= C(\bar{C} \bar{f}) \{ \text{super}(\bar{f}); \text{this}.\bar{f}=\bar{f}; \}$

$M ::= C \ m(\bar{C} \ \bar{x}) \{ \text{return } e; \}$

$e ::= x \mid e.f \mid e.m(\bar{e}) \mid \text{new } C(\bar{e}) \mid (C)e$

Generating Programs

Syntax:

$L ::= \text{class } C \text{ extends } C \{ \bar{C} \bar{f}; K \bar{M} \}$

$K ::= C(\bar{C} \bar{f}) \{ \text{super}(\bar{f}); \text{this}.\bar{f}=\bar{f}; \}$

$M ::= C m(\bar{C} \bar{x}) \{ \text{return } e; \}$

$e ::= x \mid e.f \mid e.m(\bar{e}) \mid \text{new } C(\bar{e}) \mid (C)e$

```
class Object extends C_0{
  C_0() {
    super();
  }
}
```

```
class C_0 extends C_0{
  C_0() {
    super();
  }
}
```

```
class C_0 extends Object{
  C_0() {
    super();
  }
}
```

```
class Object extends C_0{
  Object() {
    super();
  }
}
```

```
class C_0 extends C_0{
  Object() {
    super();
  }
}
```

```
class C_0 extends Object{
  Object() {
    super();
  }

  Object m_0() {
    return (this);
  }
}
```

```
class Object extends Object{
  C_0() {
    super();
  }
}
```

```
class C_0 extends Object{
  Object() {
    super();
  }
}
```

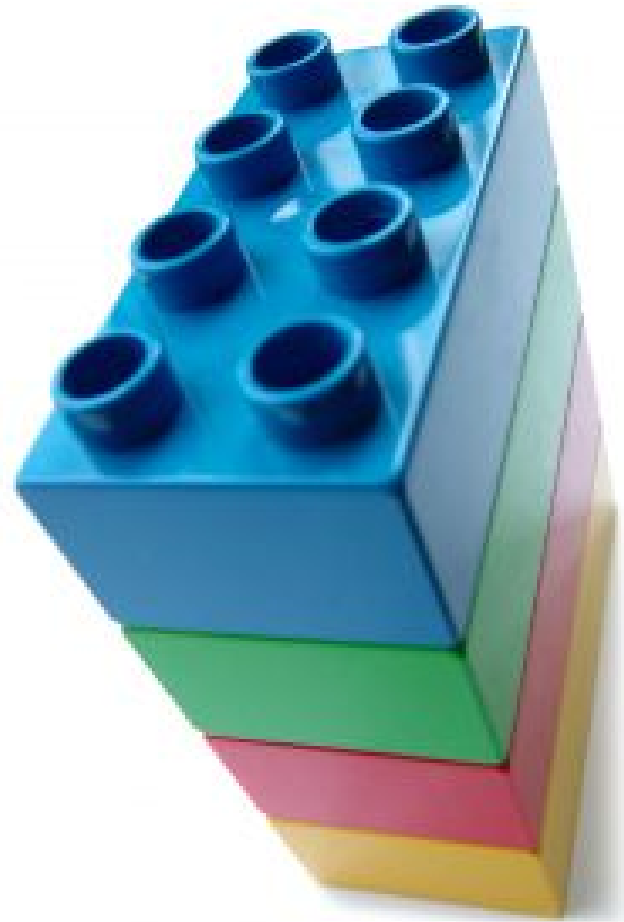
```
class Object extends Object{
  Object() {
    super();
  }
}
```

```
class C_0 extends Object{
  Object() {
    super();
  }

  C_0 m_0() {
    return (this);
  }
}
```

Generating Programs

- Bounded, depth-first generation
- What about names?



Generating Programs

ClassNames = { Object, Foo, Bar }

```
class Object extends Object {  
}
```

```
class Foo extends Object {  
}
```

```
class Bar extends Object {  
}
```

```
class Foo extends Foo {  
}
```

```
class Bar extends Bar {  
}
```

```
class Bar extends Foo {  
}
```

```
class Foo extends Bar {  
}
```

Binding

Syntax:

$L ::= \text{class } C \text{ extends } C \{ \bar{C} \bar{f}; K \bar{M} \}$

$K ::= C(\bar{C} \bar{f}) \{ \text{super}(\bar{f}); \text{this}.\bar{f}=\bar{f}; \}$

$M ::= C \bar{m}(\bar{C} \bar{x}) \{ \text{return } e; \}$

$e ::= x \mid e.f \mid e.m(\bar{e}) \mid \text{new } C(\bar{e}) \mid (C)e$

Skeletons

```
class C1 {  
    field f1;  
    method m1;  
}
```

ClassNames = { Object, C1, C2 }
FieldNames = { f1 }
MethodNames = { m1 }

```
class C2 {  
}
```

```
class C1 extends C1 {  
    C1 (Object f1) {  
        super ();  
        this.f1 = f1;  
    }  
  
    Object f1;  
    Object m1() { return this; }  
}
```

```
class C2 extends Object {  
    Object() { super(); }  
}
```

```
class C1 extends Object {  
    C1 () { super (); }  
    C2 f1;  
    C1 m1() { return new C2(); }  
}
```

```
class C2 extends C1 {  
    C2() { super(); }  
}
```



Classify Tests

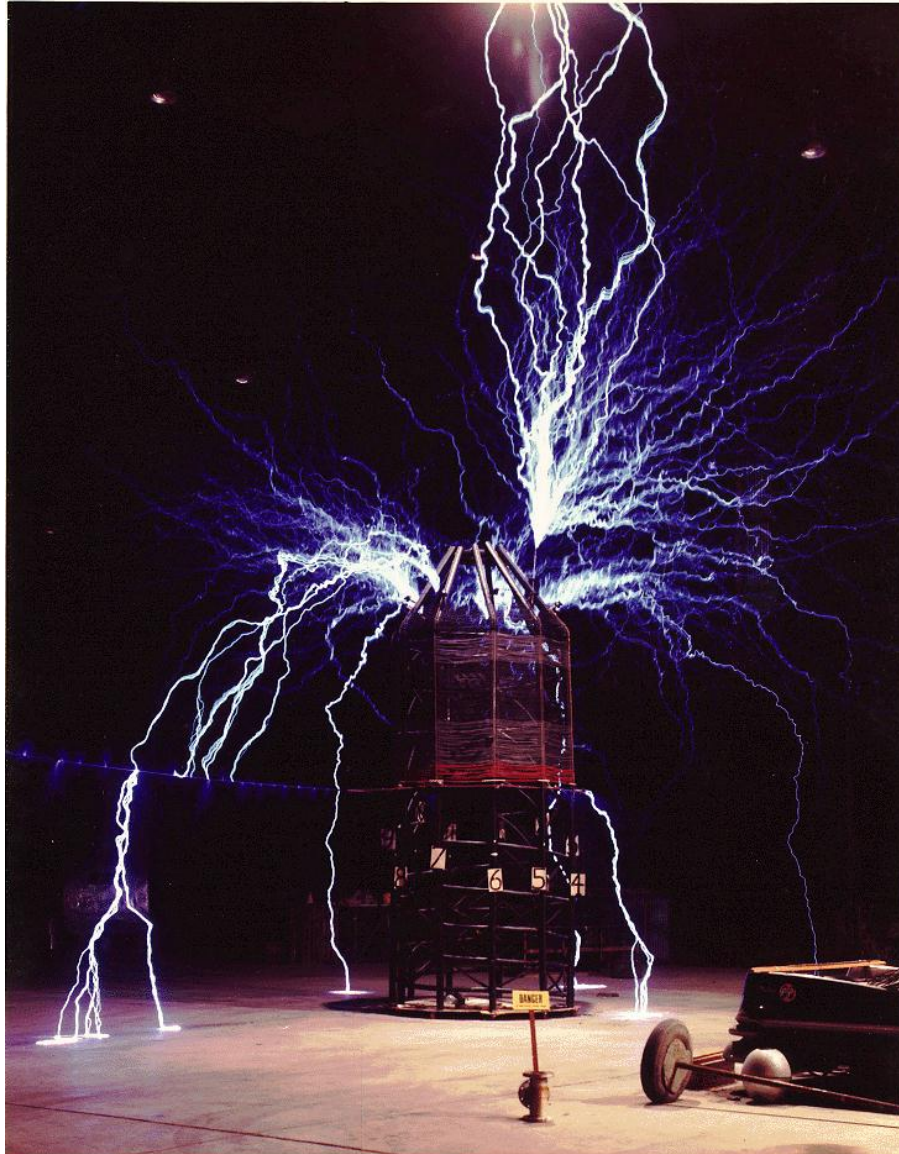


Classify Tests

- Yes
 - Javac should accept this program
- No
 - Javac should reject this program
- Maybe
 - We don't know!
 - Can't use this program as a test!



Experiment



Results

EMMA Coverage Report (generated Sat Dec 08 18:10:06 GMT 2007)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	58% (86/149)	30% (366/1222)	31% (5462/17341)	30% (1520.6/4990)

OVERALL STATS SUMMARY

total packages: 11
total executable files: 133
total classes: 149
total methods: 1222
total executable lines: 4990

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
uk.co.zonetora.javacRunner	0% (0/7)	0% (0/20)	0% (0/267)	0% (0/51)
uk.co.zonetora.fj.ast.analysis	67% (2/3)	12% (26/221)	7% (177/2405)	8% (68/844)
uk.co.zonetora.fj.ast.node	56% (54/97)	26% (177/686)	19% (1596/8292)	22% (588.6/2668)
uk.co.zonetora.fj	100% (3/3)	39% (7/18)	37% (139/376)	42% (31.6/75)
uk.co.zonetora.fj.util	75% (3/4)	38% (8/21)	41% (80/196)	37% (20.6/55)
uk.co.zonetora.fj.passes	100% (3/3)	69% (18/26)	53% (374/702)	55% (75.8/139)
uk.co.zonetora.fj.typecheck	50% (5/10)	55% (22/40)	54% (431/800)	63% (102.6/164)
uk.co.zonetora.fj.ast.parser	100% (4/4)	54% (42/78)	56% (1227/2210)	57% (257/451)
uk.co.zonetora.fj.model	64% (9/14)	51% (40/78)	58% (587/1013)	58% (140.6/244)
uk.co.zonetora.fj.ast.lexer	67% (2/3)	78% (25/32)	78% (803/1029)	78% (224/286)
uk.co.zonetora	100% (1/1)	50% (1/2)	94% (48/51)	92% (12/13)

[all classes]

EMMA 2.0.5312 (C) Vladimir Roubtsov

```
fieldDecl(AFieldDecl afd) {  
    fd.getClassName().getText().trim();  
    name(afd.getFieldname().getText().trim());  
    fieldName>(type, fieldName);  
}
```

```
145  
146     private Constructor buildConstructor(AConstructorDecl constructorDecl) {  
147         ClassName returnType = new ClassName(constructorDecl.getIdentifier().getText().trim());  
148         Constructor constructor = new Constructor(returnType);  
149  
150         List<Tuple<ClassName,ArgumentName>> parameters = getFormalParameters(constructorDecl.getParamDe  
151         for(Tuple<ClassName, ArgumentName> param : parameters) {  
152             constructor.addArgument(param.getX(), new FieldName(param.getY().getArgName()));  
153         }  
154  
155  
156         List<FieldName> superFields = getSuperFieldNames(constructorDecl.getFieldList());  
157         for(FieldName f : superFields) {  
158             constructor.addSuperField(f);  
159         }  
160  
161         for(Object o : constructorDecl.getThisFieldAssig()) {  
162             AThisFieldAssig atfa = (AThisFieldAssig) o;  
163             FieldName f;  
164             try {  
165                 f = getThisFieldAssig(atfa);  
166             } catch (FJException e) {  
167                 System.err.println(e.getMessage());  
168                 addException(e);  
169                 continue;  
170             }  
171             constructor.addLocalField(f);  
172         }
```

Results

Max #Classes	0	1	1	1	1	2	1	1	1	1	2	1	1	1
Max #Methods/Class	0	0	1	0	1	0	1	1	1	1	1	2	1	1
Max #Fields/Class	0	0	0	1	0	0	0	1	1	1	0	1	1	1
Max #Variables/Method	0	0	0	0	1	0	1	1	1	0	0	1	1	0
Max Expr. Complexity	0	0	1	0	1	0	2	1	2	3	1	1	3	4
Total # Programs	1	9	57	201	281	333	1641	7001	47529	62025	54813	238201	360137	542361
# Usable programs	1	9	57	157	281	333	1553	6141	41957	50037	54813	217757	317509	437735
UP That TC (%)	100.00%	22.22%	12.28%	2.55%	7.12%	1.50%	3.67%	0.75%	0.41%	0.37%	0.46%	0.30%	0.20%	0.15%

Entire javac	12.27%	23.36%	25.16%	25.49%	25.81%	23.44%	27.13%	27.07%	28.04%	27.97%	25.25%	27.07%	28.09%	27.97%
code package	19.04%	28.93%	31.87%	30.81%	32.10%	29.03%	35.96%	32.97%	36.28%	36.33%	31.98%	32.97%	36.33%	36.33%
comp package	6.21%	22.28%	25.21%	26.45%	26.37%	22.44%	28.46%	28.77%	30.37%	30.48%	25.36%	28.79%	30.48%	30.49%

Entire FJ	6.17%	18.38%	28.44%	25.95%	30.46%	19.18%	35.67%	36.56%	42.67%	43.85%	29.25%	37.23%	44.65%	44.43%
model package	1.64%	25.37%	54.02%	34.84%	57.62%	25.78%	72.95%	66.68%	86.93%	83.89%	57.30%	66.68%	87.46%	83.89%
passes package	3.60%	17.77%	41.08%	44.96%	54.53%	17.77%	65.76%	71.29%	86.12%	84.03%	41.08%	71.29%	87.12%	84.03%
typecheck package	15.61%	31.46%	62.56%	31.46%	62.56%	35.79%	82.13%	62.56%	87.56%	88.17%	64.45%	64.39%	91.22%	88.17%

# Type Check	1	2	7	4	20	5	57	46	171	186	251	658	633	675
---------------------	---	---	---	---	----	---	----	----	-----	-----	-----	-----	-----	-----

javac Line Count	3210.1	6112.1	6583.2	6669.3	6752.7	6133.7	7096.8	7082.4	7337	7318.4	6605.6	7083.4	7348.7	7318.8
Code Line Count	547.5	832	916.7	886.1	923.2	834.8	1034.3	948.3	1043.3	1044.8	919.8	948.3	1044.8	1044.8
Comp Line Count	372.9	1338.3	1514	1588.5	1584	1347.9	1709.6	1728.2	1823.8	1830.9	1523	1729.2	1830.9	1831.3

FJ Line Count	307.7	917.1	1419.4	1295.1	1520	957.2	1780	1824.4	2129.2	2187.9	1459.5	1857.8	2227.8	2216.9
Model Line Count	4	61.9	131.8	85	140.6	62.9	178	162.7	212.1	204.7	139.8	162.7	213.4	204.7
Passes Line Count	5	24.7	57.1	62.5	75.8	24.7	91.4	99.1	119.7	116.8	57.1	99.1	121.1	116.8
TypeCheck Line Count	25.6	51.6	102.6	51.6	102.6	58.7	134.7	102.6	143.6	144.6	105.7	105.6	149.6	144.6

Results

Max #Classes	0	1	1	1	1	2	1	1	1	1	2	1	1	1
Max #Methods/Class	0	0	1	0	1	0	1	1	1	1	1	2	1	1
Max #Fields/Class	0	0	0	1	0	0	0	1	1	1	0	1	1	1
Max #Variables/Method	0	0	0	0	1	0	1	1	1	0	0	1	1	0
Max Expr. Complexity	0	0	1	0	1	0	2	1	2	3	1	1	3	4
Total # Programs	1	9	57	201	281	333	1641	7001	47529	62025	54813	238201	360137	542364
# Usable programs	1	9	57	157	281	333	1553	6141	41957	50037	54813	217757	317509	437735
UP That TC (%)	100.00%	22.22%	12.28%	2.55%	7.12%	1.50%	3.67%	0.75%	0.41%	0.37%	0.46%	0.30%	0.20%	0.15%

Entire javac	12.27%	23.36%	25.16%	25.49%	25.81%	23.44%	27.13%	27.07%	28.04%	27.97%	25.25%	27.07%	28.09%	27.97%
code package	19.04%	28.93%	31.87%	30.81%	32.10%	29.03%	35.96%	32.97%	36.28%	36.33%	31.98%	32.97%	36.33%	36.33%
comp package	6.21%	22.28%	25.21%	26.45%	26.37%	22.44%	28.46%	28.77%	30.37%	30.48%	25.36%	28.79%	30.48%	30.49%

Entire FJ	6.17%	18.38%	28.44%	25.95%	30.46%	19.18%	35.67%	36.56%	42.67%	43.85%	29.25%	37.23%	44.65%	44.43%
model package	1.64%	25.37%	54.02%	34.84%	57.62%	25.78%	72.95%	66.68%	86.93%	83.89%	57.30%	66.68%	87.46%	83.89%
passes package	3.60%	17.77%	41.08%	44.96%	54.53%	17.77%	65.76%	71.29%	86.12%	84.03%	41.08%	71.29%	87.12%	84.03%
typecheck package	15.61%	31.46%	62.56%	31.46%	62.56%	35.79%	82.13%	62.56%	87.56%	88.17%	64.45%	64.39%	91.22%	88.17%

# Type Check	1	2	7	4	20	5	57	46	171	186	251	658	633	675
javac Line Count	3210.1	6112.1	6583.2	6669.3	6752.7	6133.7	7096.8	7082.4	7337	7318.4	6605.6	7083.4	7348.7	7318.8
Code Line Count	547.5	832	916.7	886.1	923.2	834.8	1034.3	948.3	1043.3	1044.8	919.8	948.3	1044.8	1044.8
Comp Line Count	372.9	1338.3	1514	1588.5	1584	1347.9	1709.6	1728.2	1823.8	1830.9	1523	1729.2	1830.9	1831.3
FJ Line Count	307.7	917.1	1419.4	1295.1	1520	957.2	1780	1824.4	2129.2	2187.9	1459.5	1857.8	2227.8	2216.9
Model Line Count	4	61.9	131.8	85	140.6	62.9	178	162.7	212.1	204.7	139.8	162.7	213.4	204.7
Passes Line Count	5	24.7	57.1	62.5	75.8	24.7	91.4	99.1	119.7	116.8	57.1	99.1	121.1	116.8
TypeCheck Line Count	25.6	51.6	102.6	51.6	102.6	58.7	134.7	102.6	143.6	144.6	105.7	105.6	149.6	144.6

Results

Max #Classes	0	1	1	1	1	2	1	1	1	1	2	1	1	1
Max #Methods/Class	0	0	1	0	1	0	1	1	1	1	1	2	1	1
Max #Fields/Class	0	0	0	1	0	0	0	1	1	1	0	1	1	1
Max #Variables/Method	0	0	0	0	1	0	1	1	1	0	0	1	1	0
Max Expr. Complexity	0	0	1	0	1	0	2	1	2	3	1	1	3	4
Total # Programs	1	9	57	201	281	333	1641	7001	47529	62025	54813	238201	360137	542364
# Usable programs	1	9	57	157	281	333	1553	6141	41957	50037	54813	217757	317509	437735
UP That TC (%)	100.00%	22.22%	12.28%	2.55%	7.12%	1.50%	3.67%	0.75%	0.41%	0.37%	0.46%	0.30%	0.20%	0.15%

Entire javac	12.27%	23.36%	25.16%	25.49%	25.81%	23.44%	27.13%	27.07%	28.04%	27.97%	25.25%	27.07%	28.09%	27.97%
code package	19.04%	28.93%	31.87%	30.81%	32.10%	29.03%	35.96%	32.97%	36.28%	36.33%	31.98%	32.97%	36.33%	36.33%
comp package	6.21%	22.28%	25.21%	26.45%	26.37%	22.44%	28.46%	28.77%	30.37%	30.48%	25.36%	28.79%	30.48%	30.49%

Entire FJ	6.17%	18.38%	28.44%	25.95%	30.46%	19.18%	35.67%	36.56%	42.67%	43.85%	29.25%	37.23%	44.65%	44.43%
model package	1.64%	25.37%	54.02%	34.84%	57.62%	25.78%	72.95%	66.68%	86.93%	83.89%	57.30%	66.68%	87.46%	83.89%
passes package	3.60%	17.77%	41.08%	44.96%	54.53%	17.77%	65.76%	71.29%	86.12%	84.03%	41.08%	71.29%	87.12%	84.03%
typecheck package	15.61%	31.46%	62.56%	31.46%	62.56%	35.79%	82.13%	62.56%	87.56%	88.17%	64.45%	64.39%	91.22%	88.17%

# Type Check	1	2	7	4	20	5	57	46	171	186	251	658	633	675
javac Line Count	3210.1	6112.1	6583.2	6669.3	6752.7	6133.7	7096.8	7082.4	7337	7318.4	6605.6	7083.4	7348.7	7318.8
Code Line Count	547.5	832	916.7	886.1	923.2	834.8	1034.3	948.3	1043.3	1044.8	919.8	948.3	1044.8	1044.8
Comp Line Count	372.9	1338.3	1514	1588.5	1584	1347.9	1709.6	1728.2	1823.8	1830.9	1523	1729.2	1830.9	1831.3
FJ Line Count	307.7	917.1	1419.4	1295.1	1520	957.2	1780	1824.4	2129.2	2187.9	1459.5	1857.8	2227.8	2216.9
Model Line Count	4	61.9	131.8	85	140.6	62.9	178	162.7	212.1	204.7	139.8	162.7	213.4	204.7
Passes Line Count	5	24.7	57.1	62.5	75.8	24.7	91.4	99.1	119.7	116.8	57.1	99.1	121.1	116.8
TypeCheck Line Count	25.6	51.6	102.6	51.6	102.6	58.7	134.7	102.6	143.6	144.6	105.7	105.6	149.6	144.6

Results

Max #Classes	0	1	1	1	1	2	1	1	1	1	2	1	1	1
Max #Methods/Class	0	0	1	0	1	0	1	1	1	1	1	2	1	1
Max #Fields/Class	0	0	0	1	0	0	0	1	1	1	0	1	1	1
Max #Variables/Method	0	0	0	0	1	0	1	1	1	0	0	1	1	0
Max Expr. Complexity	0	0	1	0	1	0	2	1	2	3	1	1	3	4
Total # Programs	1	9	57	201	281	333	1641	7001	47529	62025	54813	238201	360137	542361
# Usable programs	1	9	57	157	281	333	1553	6141	41957	50037	54813	217757	317509	437735
UP That TC (%)	100.00%	22.22%	12.28%	2.55%	7.12%	1.50%	3.67%	0.75%	0.41%	0.37%	0.46%	0.30%	0.20%	0.15%

Entire javac	12.27%	23.36%	25.16%	25.49%	25.81%	23.44%	27.13%	27.07%	28.04%	27.97%	25.25%	27.07%	28.09%	27.97%
code package	19.04%	28.93%	31.87%	30.81%	32.10%	29.03%	35.96%	32.97%	36.28%	36.33%	31.98%	32.97%	36.33%	36.33%
comp package	6.21%	22.28%	25.21%	26.45%	26.37%	22.44%	28.46%	28.77%	30.37%	30.48%	25.36%	28.79%	30.48%	30.49%

Entire FJ	6.17%	18.38%	28.44%	25.95%	30.46%	19.18%	35.67%	36.56%	42.67%	43.85%	29.25%	37.23%	44.65%	44.43%
model package	1.64%	25.37%	54.02%	34.84%	57.62%	25.78%	72.95%	66.68%	86.93%	83.89%	57.30%	66.68%	87.46%	83.89%
passes package	3.60%	17.77%	41.08%	44.96%	54.53%	17.77%	65.76%	71.29%	86.12%	84.03%	41.08%	71.29%	87.12%	84.03%
typecheck package	15.61%	31.46%	62.56%	31.46%	62.56%	35.79%	82.13%	62.56%	87.56%	88.17%	64.45%	64.39%	91.22%	88.17%

# Type Check	1	2	7	4	20	5	57	46	171	186	251	658	633	675
---------------------	---	---	---	---	----	---	----	----	-----	-----	-----	-----	-----	-----

javac Line Count	3210.1	6112.1	6583.2	6669.3	6752.7	6133.7	7096.8	7082.4	7337	7318.4	6605.6	7083.4	7348.7	7318.8
Code Line Count	547.5	832	916.7	886.1	923.2	834.8	1034.3	948.3	1043.3	1044.8	919.8	948.3	1044.8	1044.8
Comp Line Count	372.9	1338.3	1514	1588.5	1584	1347.9	1709.6	1728.2	1823.8	1830.9	1523	1729.2	1830.9	1831.3

FJ Line Count	307.7	917.1	1419.4	1295.1	1520	957.2	1780	1824.4	2129.2	2187.9	1459.5	1857.8	2227.8	2216.9
Model Line Count	4	61.9	131.8	85	140.6	62.9	178	162.7	212.1	204.7	139.8	162.7	213.4	204.7
Passes Line Count	5	24.7	57.1	62.5	75.8	24.7	91.4	99.1	119.7	116.8	57.1	99.1	121.1	116.8
TypeCheck Line Count	25.6	51.6	102.6	51.6	102.6	58.7	134.7	102.6	143.6	144.6	105.7	105.6	149.6	144.6

Results

Max #Classes	0	1	1	1	1	2	1	1	1	1	2	1	1	1
Max #Methods/Class	0	0	1	0	1	0	1	1	1	1	1	2	1	1
Max #Fields/Class	0	0	0	1	0	0	0	1	1	1	0	1	1	1
Max #Variables/Method	0	0	0	0	1	0	1	1	1	0	0	1	1	0
Max Expr. Complexity	0	0	1	0	1	0	2	1	2	3	1	1	3	4
Total # Programs	1	9	57	201	281	333	1641	7001	47529	62025	54813	238201	360137	542361
# Usable programs	1	9	57	157	281	333	1553	6141	41957	50037	54813	217757	317509	437735
UP That TC (%)	100.00%	22.22%	12.28%	2.55%	7.12%	1.50%	3.67%	0.75%	0.41%	0.37%	0.46%	0.30%	0.20%	0.15%

Entire javac	12.27%	23.36%	25.16%	25.49%	25.81%	23.44%	27.13%	27.07%	28.04%	27.97%	25.25%	27.07%	28.09%	27.97%
code package	19.04%	28.93%	31.87%	30.81%	32.10%	29.03%	35.96%	32.97%	36.28%	36.33%	31.98%	32.97%	36.33%	36.33%
comp package	6.21%	22.28%	25.21%	26.45%	26.37%	22.44%	28.46%	28.77%	30.37%	30.48%	25.36%	28.79%	30.48%	30.49%

Entire FJ	6.17%	18.38%	28.44%	25.95%	30.46%	19.18%	35.67%	36.56%	42.67%	43.85%	29.25%	37.23%	44.65%	44.43%
model package	1.64%	25.37%	54.02%	34.84%	57.62%	25.78%	72.95%	66.68%	86.93%	83.89%	57.30%	66.68%	87.46%	83.89%
passes package	3.60%	17.77%	41.08%	44.96%	54.53%	17.77%	65.76%	71.29%	86.12%	84.03%	41.08%	71.29%	87.12%	84.03%
typecheck package	15.61%	31.46%	62.56%	31.46%	62.56%	35.79%	82.13%	62.56%	87.56%	88.17%	64.45%	64.39%	91.22%	88.17%

# Type Check	1	2	7	4	20	5	57	46	171	186	251	658	633	675
---------------------	---	---	---	---	----	---	----	----	-----	-----	-----	-----	-----	-----

javac Line Count	3210.1	6112.1	6583.2	6669.3	6752.7	6133.7	7096.8	7082.4	7337	7318.4	6605.6	7083.4	7348.7	7318.8
Code Line Count	547.5	832	916.7	886.1	923.2	834.8	1034.3	948.3	1043.3	1044.8	919.8	948.3	1044.8	1044.8
Comp Line Count	372.9	1338.3	1514	1588.5	1584	1347.9	1709.6	1728.2	1823.8	1830.9	1523	1729.2	1830.9	1831.3

FJ Line Count	307.7	917.1	1419.4	1295.1	1520	957.2	1780	1824.4	2129.2	2187.9	1459.5	1857.8	2227.8	2216.9
Model Line Count	4	61.9	131.8	85	140.6	62.9	178	162.7	212.1	204.7	139.8	162.7	213.4	204.7
Passes Line Count	5	24.7	57.1	62.5	75.8	24.7	91.4	99.1	119.7	116.8	57.1	99.1	121.1	116.8
TypeCheck Line Count	25.6	51.6	102.6	51.6	102.6	58.7	134.7	102.6	143.6	144.6	105.7	105.6	149.6	144.6

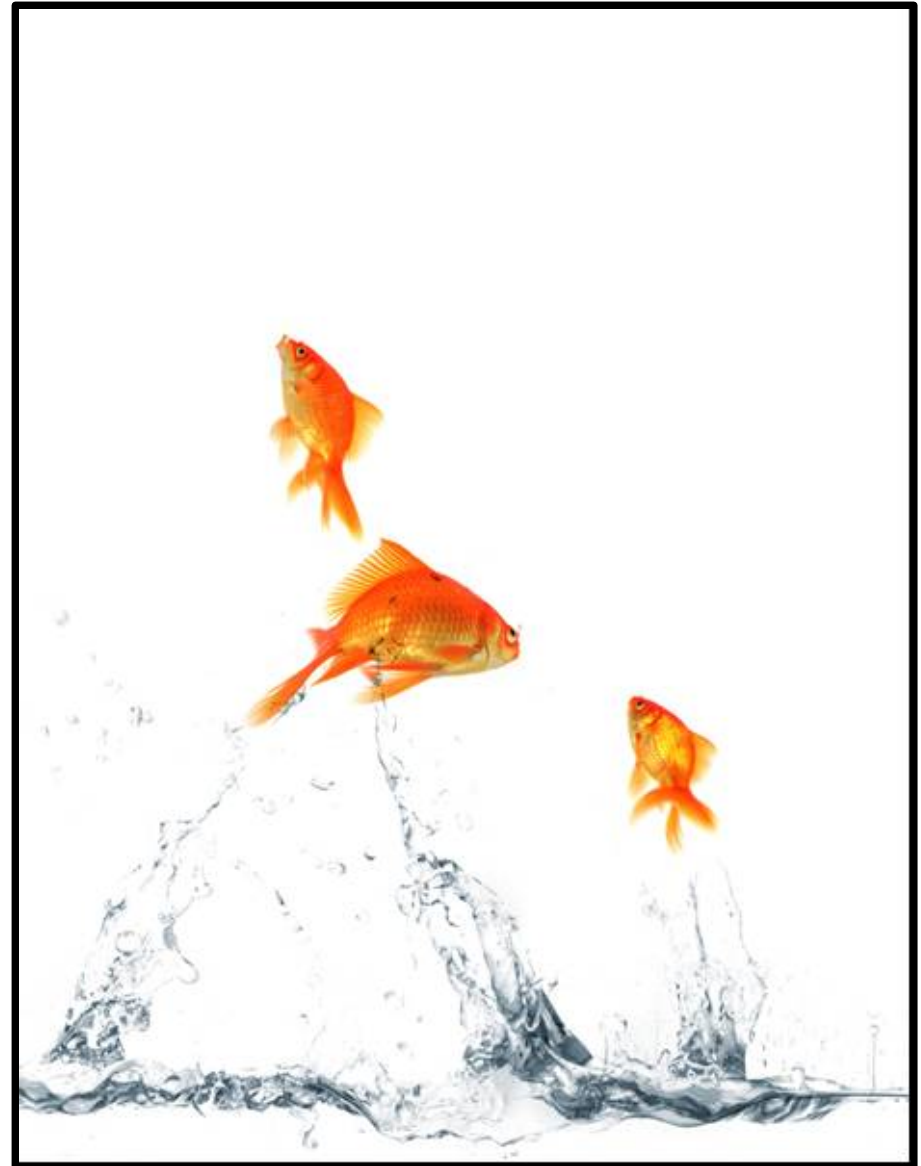
Related Work

- Automated test data generation
- Model checking
- Binding



Conclusions

- Possible to generate large numbers of tests cheaply & easily
- Can naively achieve 80%-90% code coverage of a FJ type checker
- Corresponds to 25%-30% code coverage of Javac





Thank-you for listening!

Results

Max #Classes	0	1	1	1	1	2	1	1	1	1	2	1	1	1
Max #Methods/Class	0	0	1	0	1	0	1	1	1	1	1	2	1	1
Max #Fields/Class	0	0	0	1	0	0	0	1	1	1	0	1	1	1
Max #Variables/Method	0	0	0	0	1	0	1	1	1	0	0	1	1	0
Max Expr. Complexity	0	0	1	0	1	0	2	1	2	3	1	1	3	4
Total # Programs	1	9	57	201	281	333	1641	7001	47529	62025	54813	238201	360137	542361
# Usable programs	1	9	57	157	281	333	1553	6141	41957	50037	54813	217757	317509	437735
UP That TC (%)	100.00%	22.22%	12.28%	2.55%	7.12%	1.50%	3.67%	0.75%	0.41%	0.37%	0.46%	0.30%	0.20%	0.15%

Entire javac	12.27%	23.36%	25.16%	25.49%	25.81%	23.44%	27.13%	27.07%	28.04%	27.97%	25.25%	27.07%	28.09%	27.97%
code package	19.04%	28.93%	31.87%	30.81%	32.10%	29.03%	35.96%	32.97%	36.28%	36.33%	31.98%	32.97%	36.33%	36.33%
comp package	6.21%	22.28%	25.21%	26.45%	26.37%	22.44%	28.46%	28.77%	30.37%	30.48%	25.36%	28.79%	30.48%	30.49%

Entire FJ	6.17%	18.38%	28.44%	25.95%	30.46%	19.18%	35.67%	36.56%	42.67%	43.85%	29.25%	37.23%	44.65%	44.43%
model package	1.64%	25.37%	54.02%	34.84%	57.62%	25.78%	72.95%	66.68%	86.93%	83.89%	57.30%	66.68%	87.46%	83.89%
passes package	3.60%	17.77%	41.08%	44.96%	54.53%	17.77%	65.76%	71.29%	86.12%	84.03%	41.08%	71.29%	87.12%	84.03%
typecheck package	15.61%	31.46%	62.56%	31.46%	62.56%	35.79%	82.13%	62.56%	87.56%	88.17%	64.45%	64.39%	91.22%	88.17%

# Type Check	1	2	7	4	20	5	57	46	171	186	251	658	633	675
---------------------	---	---	---	---	----	---	----	----	-----	-----	-----	-----	-----	-----

javac Line Count	3210.1	6112.1	6583.2	6669.3	6752.7	6133.7	7096.8	7082.4	7337	7318.4	6605.6	7083.4	7348.7	7318.8
Code Line Count	547.5	832	916.7	886.1	923.2	834.8	1034.3	948.3	1043.3	1044.8	919.8	948.3	1044.8	1044.8
Comp Line Count	372.9	1338.3	1514	1588.5	1584	1347.9	1709.6	1728.2	1823.8	1830.9	1523	1729.2	1830.9	1831.3

FJ Line Count	307.7	917.1	1419.4	1295.1	1520	957.2	1780	1824.4	2129.2	2187.9	1459.5	1857.8	2227.8	2216.9
Model Line Count	4	61.9	131.8	85	140.6	62.9	178	162.7	212.1	204.7	139.8	162.7	213.4	204.7
Passes Line Count	5	24.7	57.1	62.5	75.8	24.7	91.4	99.1	119.7	116.8	57.1	99.1	121.1	116.8
TypeCheck Line Count	25.6	51.6	102.6	51.6	102.6	58.7	134.7	102.6	143.6	144.6	105.7	105.6	149.6	144.6