

1,000 Testers in a Shoe Box

Advances in Automated Test Generation

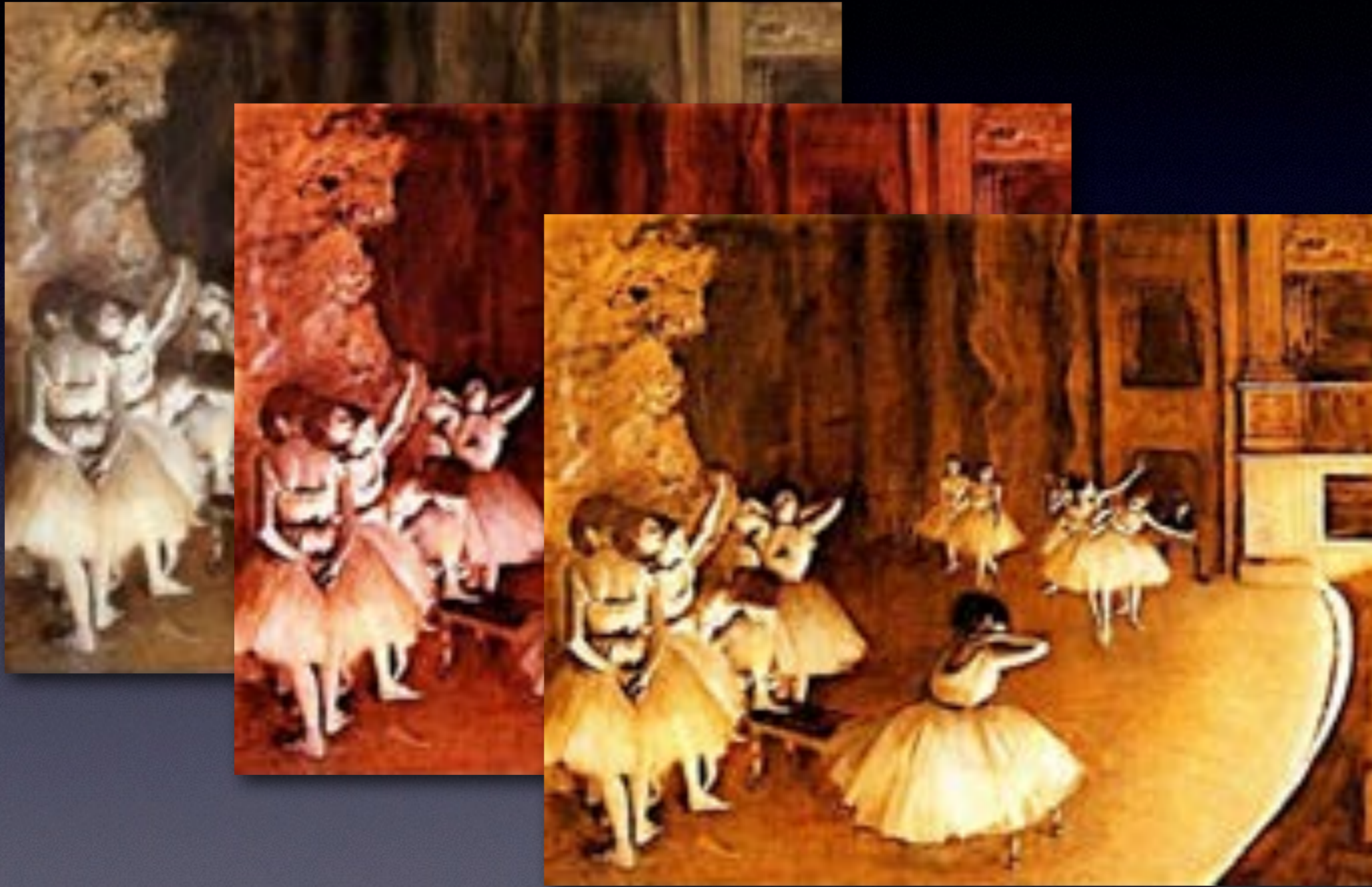
Andreas Zeller
Saarland University, Saarbrücken, Germany



Testing



Software is manifold



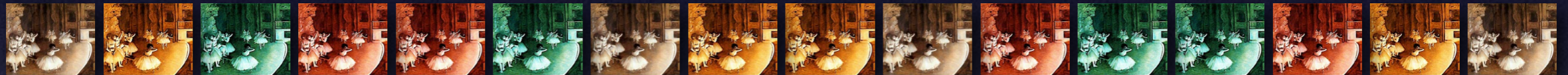
Software is manifold



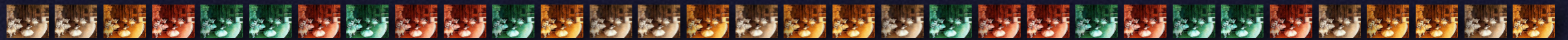
Software is manifold



Software is manifold



Software is manifold



Software is manifold

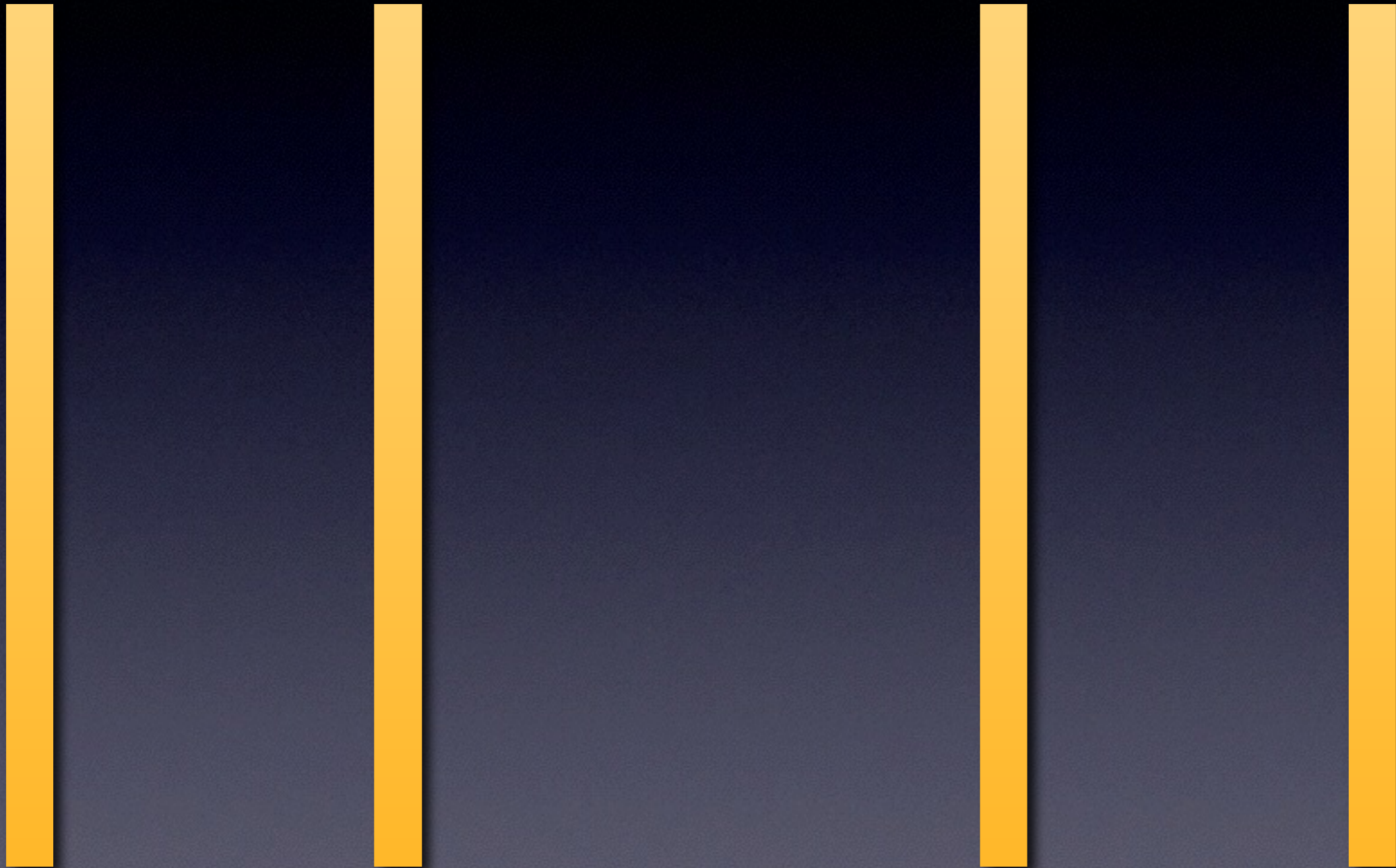


Testing



Configurations

Testing



Configurations

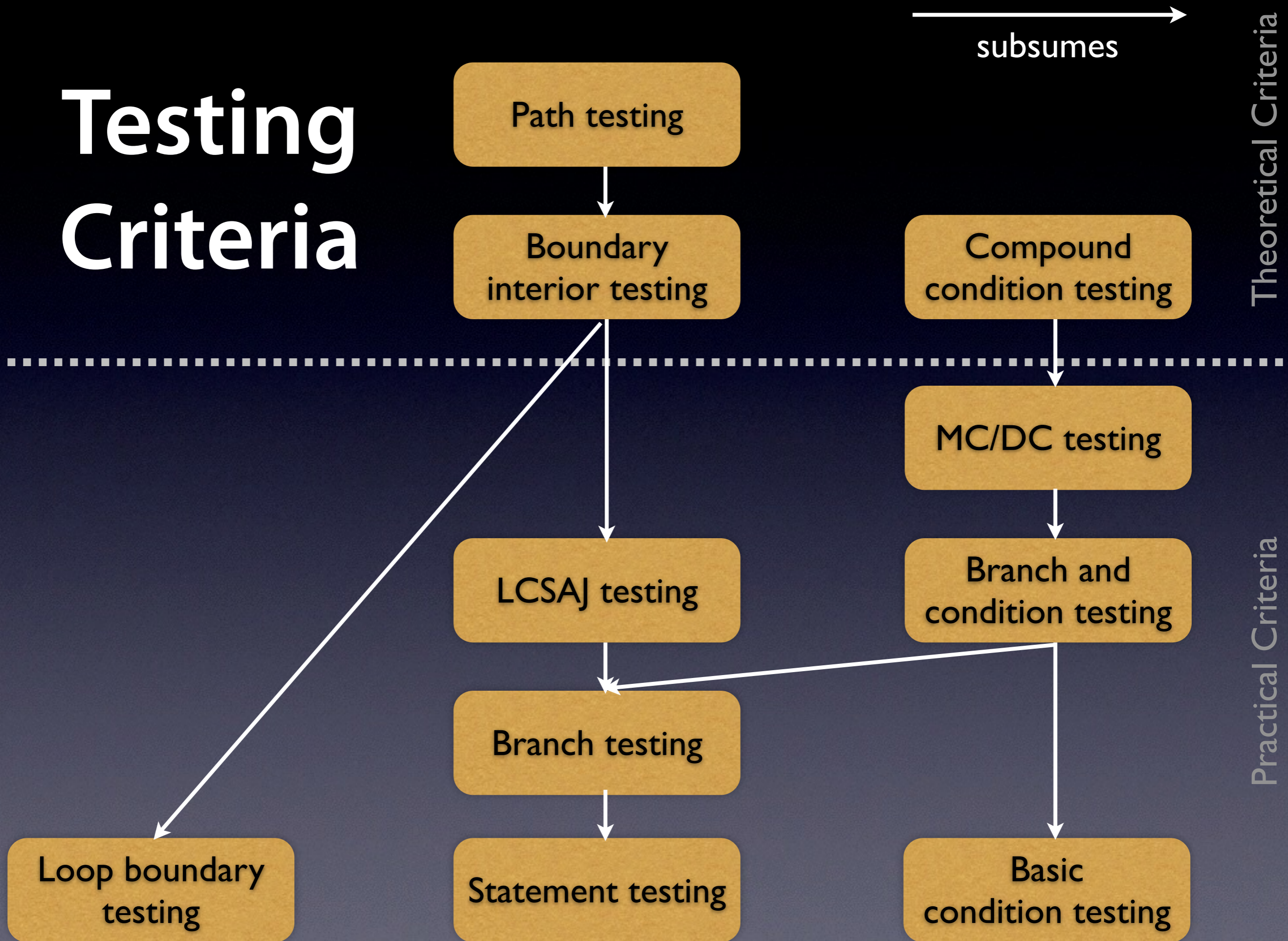
What to test?

How do we cover as much behavior as possible?

Configurations



Testing Criteria



Dijkstra's Curse

Testing can only find the
presence of errors,
not their *absence*

Configurations →

Formal Verification



Configurations →

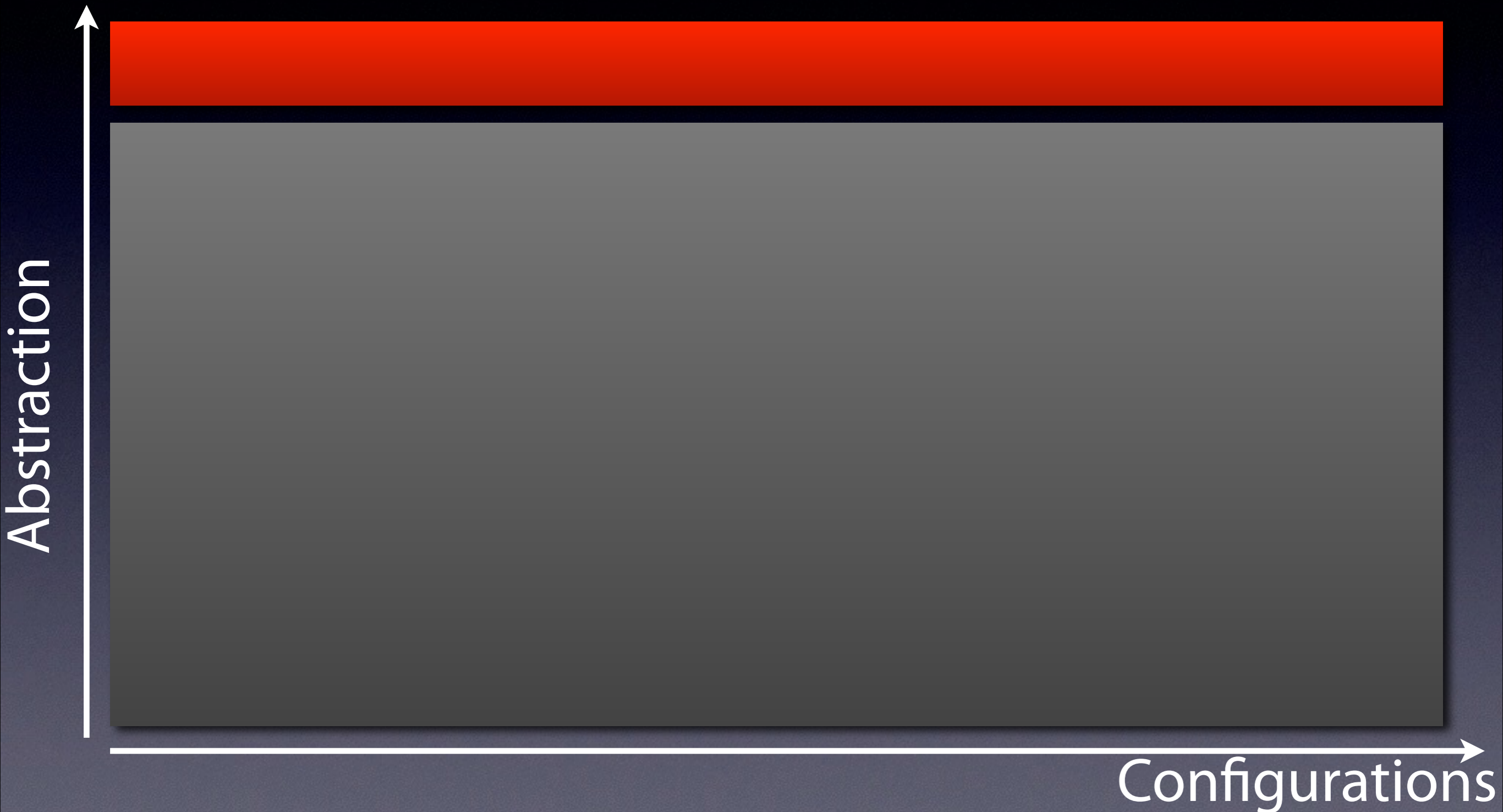
Formal Verification

Abstraction



Configurations

Formal Verification



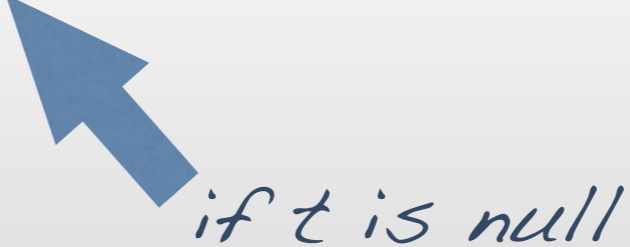
Formal Verification

Abstraction

Configurations

“This might crash”

```
static tree
fold_negate_expr (location_t loc, tree t)
{
    // 200 Lines...
    tem = TREE_OPERAND (t, 0);
    if ((INTEGRAL_TYPE_P (type)
        && (TREE_CODE (tem) == NEGATE_EXPR
            || (TREE_CODE (tem) == INTEGER_CST
                && may_negate_without_overflow_p (tem))))
        || !INTEGRAL_TYPE_P (type))
        return fold_build2_loc (loc, TREE_CODE (t), type,
                                negate_expr (tem), TREE_OPERAND (t, 1));
}
```

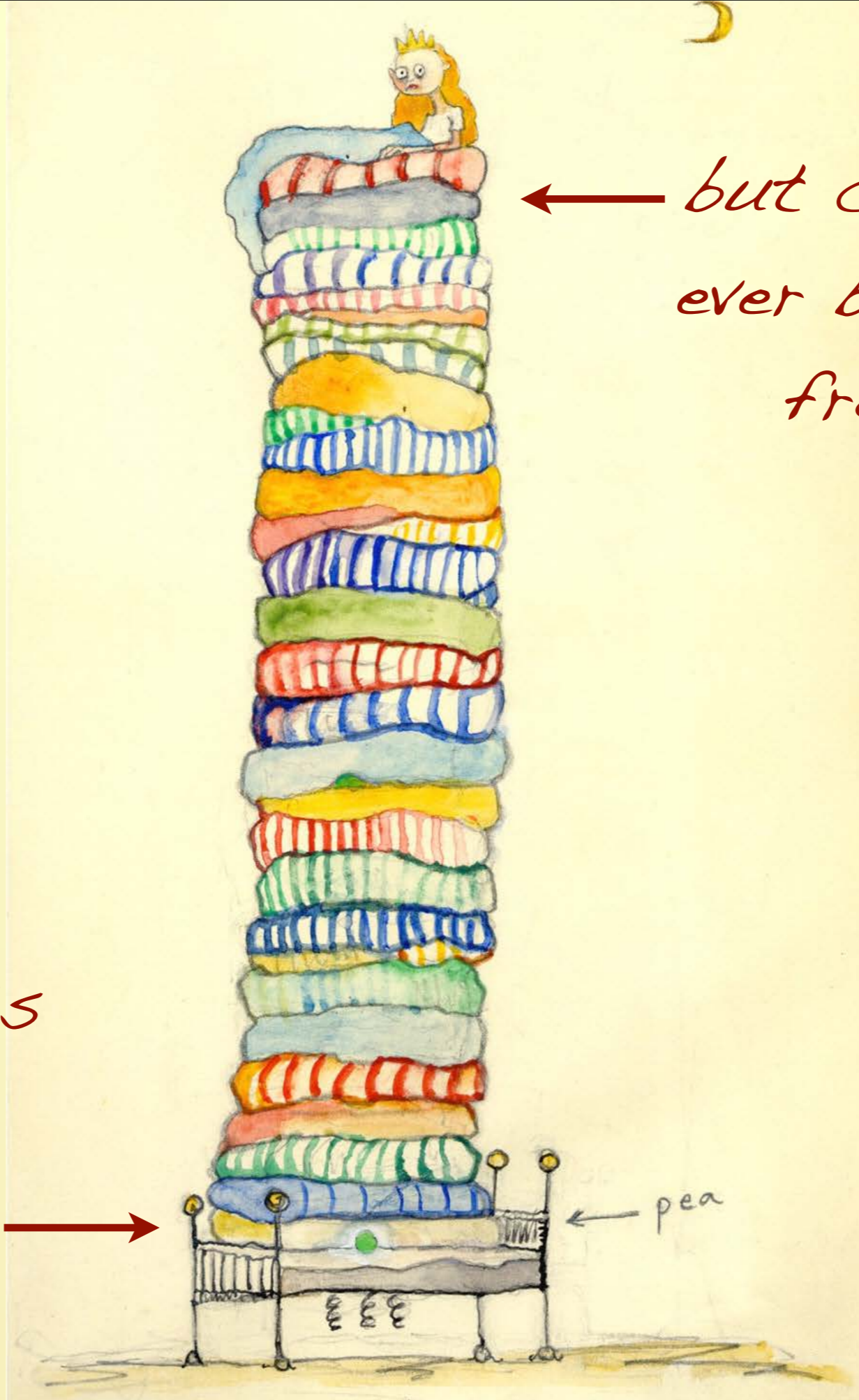


if t is null



← pea

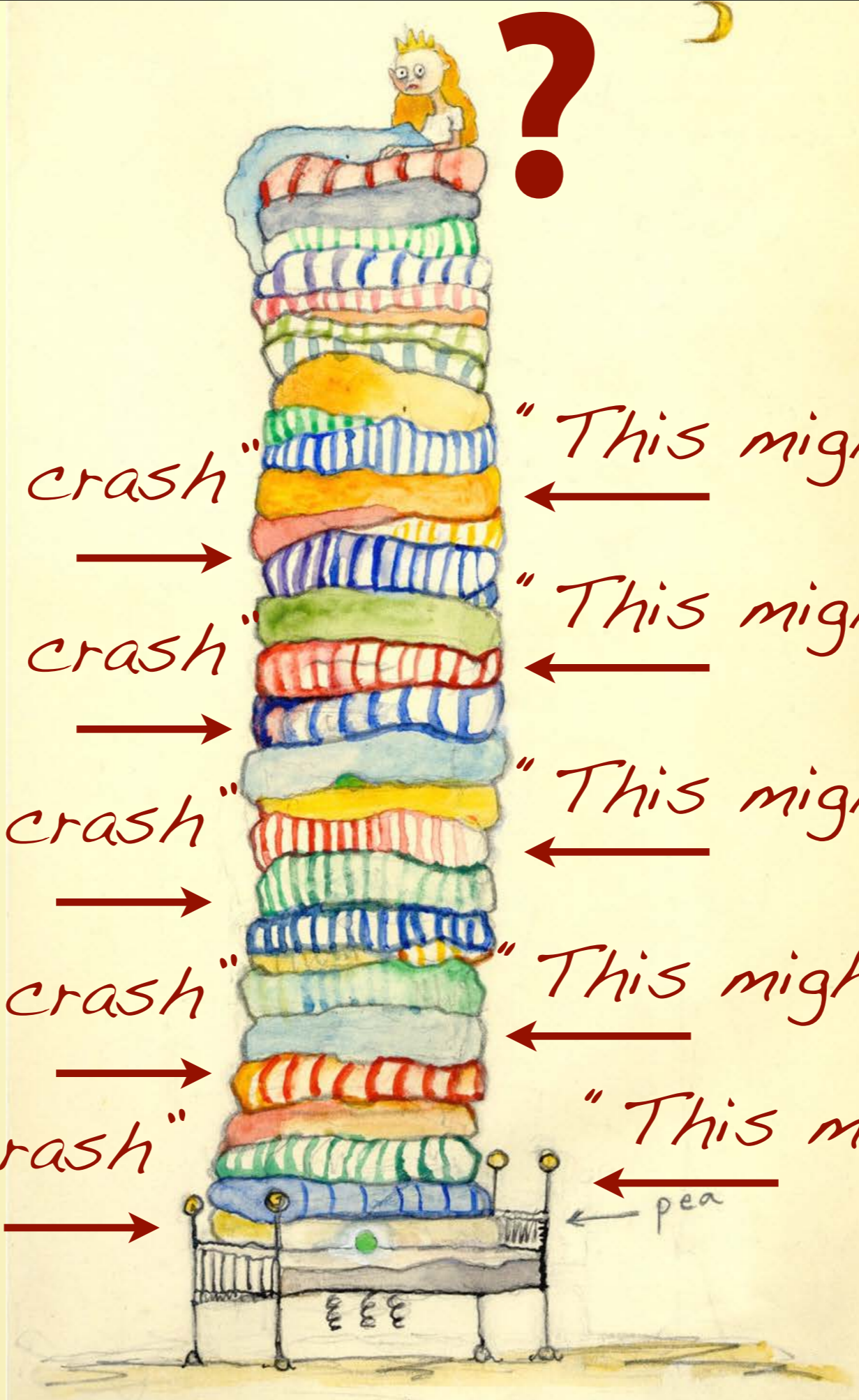




← but can these
ever be invoked
from here?

static analysis
may detect
errors here... →

← pea



?

"This might crash"

"This might crash"

"This might crash"

"This might crash"

"This might crash"

"This might crash"

"This might crash"

"This might crash"

"This might crash"

"This might crash"

← pea

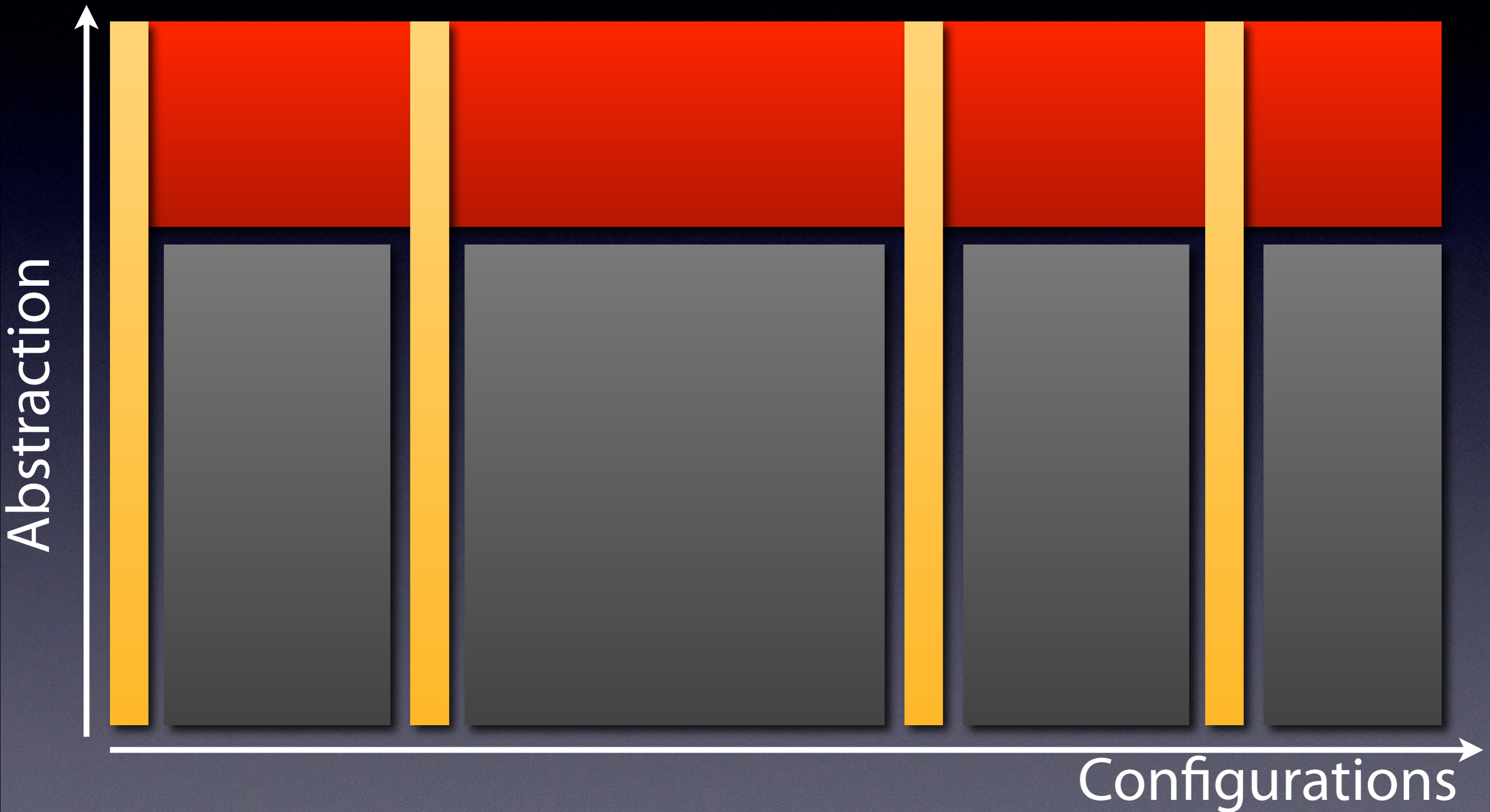
Zeller's Variation on Dijkstra

Abstraction

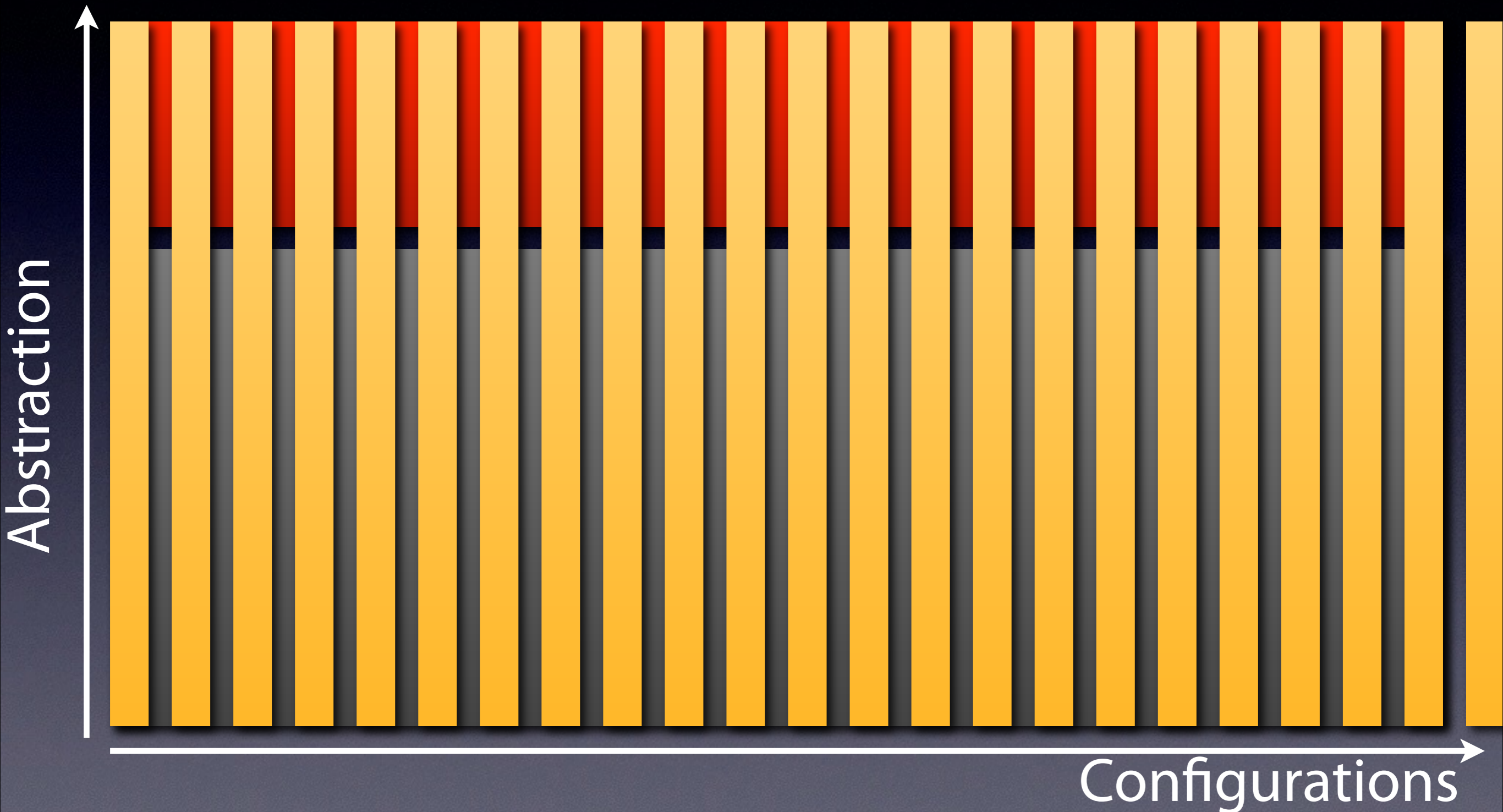
Verification can only find
the *absence* of errors,
but never their *presence*

Configurations

The Best of two Worlds



Generating Tests



Address Book

New contact
New category

First name	Last name	E-mail	Phone	Mobile
James S.	Roebuck	JamesSRoe...	561-888-...	561-888-...
Naomi D.	Long	NaomiDLo...	390-12-5...	390-12-1...
Karen L.	Lloyd	KarenLLo...	228-76-1...	228-76-...
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

- All

 - Contractors
 - Customers
 - Employees
 - ▼

Suppliers
 - Europe
 - U.S.

First name

Last name

Phone

Mobile

Notes

E-Mail

Second e-mail

URL

Generated Unit Test

```
public class RandoopTest0 extends TestCase {
```

```
...
```

```
public void test8() throws Throwable {
```

```
    if (debug) System.out.printf("%nRandoopTest0.test8");
```

```
    AddressBook var0 = new AddressBook();
```

```
    EventHandler var1 = var0.getEventHandler();
```

```
    Category var2 = var0.getRootCategory();
```

```
    Contact var3 = new Contact();
```

```
    AddressBook var4 = new AddressBook();
```

```
    EventHandler var5 = var4.getEventHandler();
```

```
    Category var6 = var4.getRootCategory();
```

```
    String var7 = var6.getName();
```

```
    var0.addCategory(var3, var6);
```

```
    SelectionHandler var9 = new SelectionHandler();
```

```
    AddressBook var10 = new AddressBook();
```

```
    EventHandler var11 = var10.getEventHandler();
```

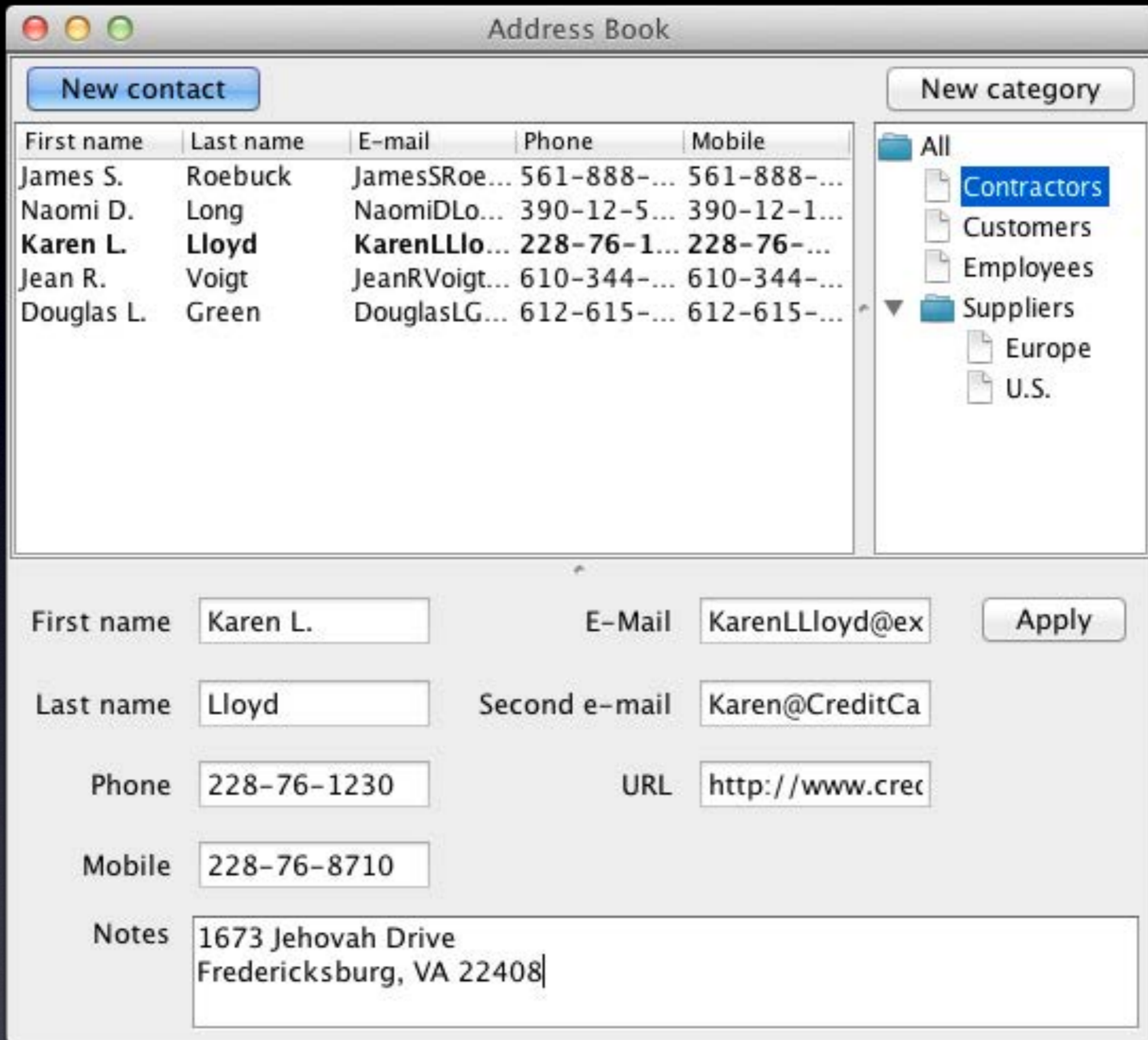
```
    Category var12 = var10.getRootCategory();
```

```
mainWindow var61 = new MainWindow(var0);
AddressBook var65 = new AddressBook();
EventHandler var66 = var65.getEventHandler();
Category var67 = var65.getRootCategory();
Contact var68 = new Contact();
Category[] var69 = var68.getCategories();
var65.removeContact(var68);
java.util.List var71 = var65.getContacts();
AddressBook var72 = new AddressBook();
EventHandler var73 = var72.getEventHandler();
Category var74 = var72.getRootCategory();
EventHandler var75 = var72.getEventHandler();
SelectionHandler var76 = new SelectionHandler();
actions.CreateContactAction var77 = new actions.CreateContactAction(var72, var76);
boolean var78 = var77.isEnabled();
AddressBook var79 = new AddressBook();
EventHandler var80 = var79.getEventHandler();
Category var81 = var79.getRootCategory();
String var82 = var81.getName();
var77.categorySelected(var81);
Category var85 = var65.createCategory(var81, "hi!");
String var86 = var85.toString();
Category var88 = var0.createCategory(var85, "exceptions.NameAlreadyInUseException");
}
```

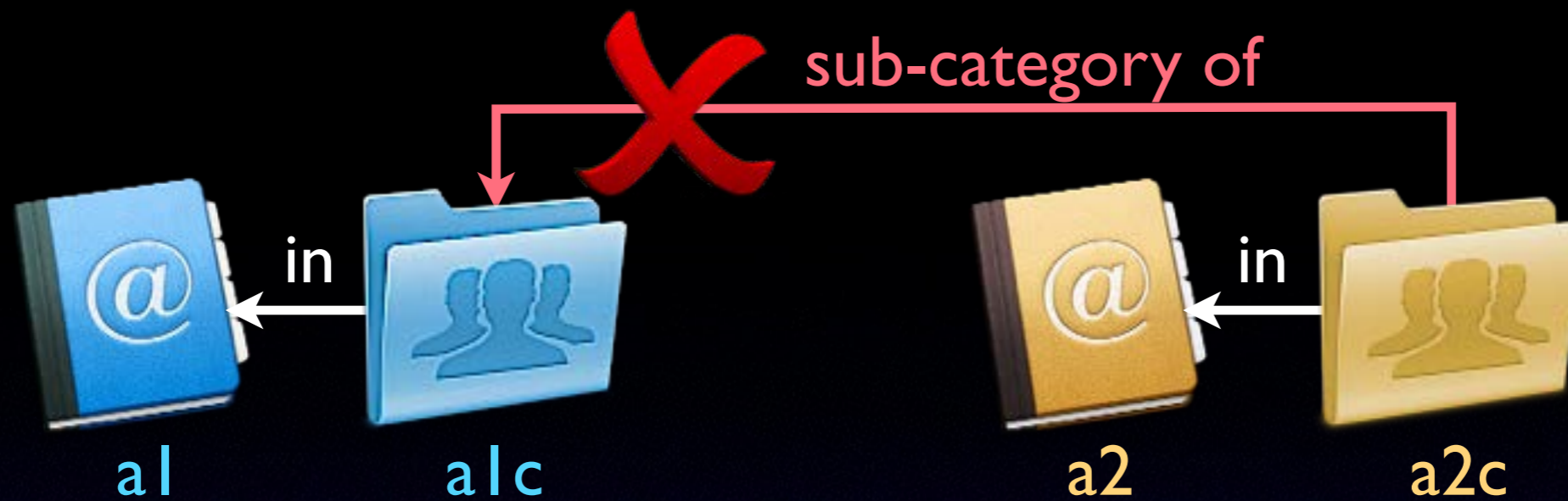


Simplified Test Case

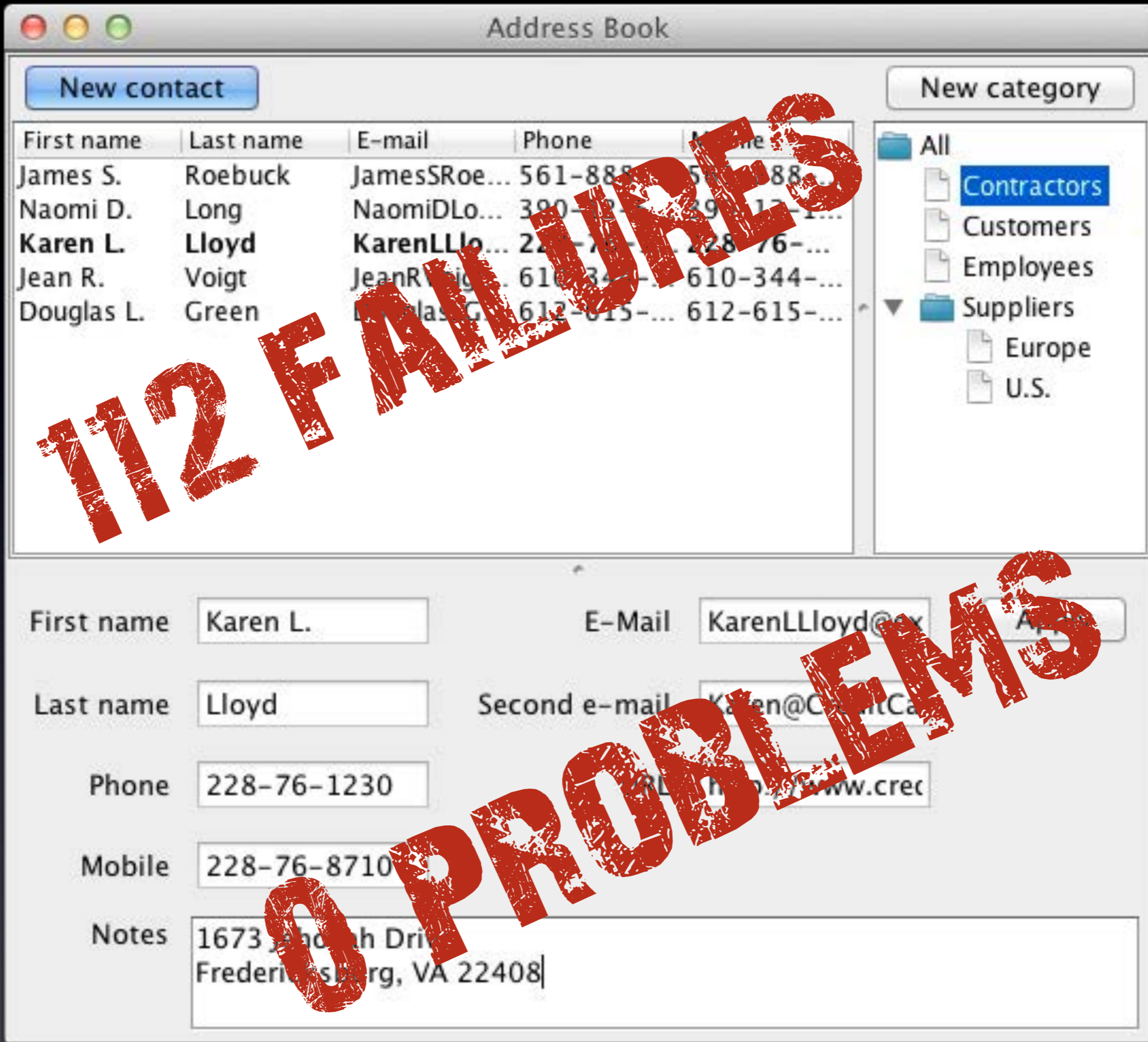
```
public class RandoopTest0 extends TestCase {  
    public void test8() throws Throwable {  
        if (debug) System.out.printf("%nRandoopTest0.test8");  
  
        AddressBook a1 = new AddressBook();  
        AddressBook a2 = new AddressBook();  
        Category a1c = a1.createCategory(a1.getRootCategory(), "a1c");  
        Category a2c = a2.createCategory(a1c, "a2c");  
    }  
}
```



how many addressbooks?



```
public class RandoopTest0 extends TestCase {  
    public void test8() throws Throwable {  
        if (debug) System.out.printf("%nRandoopTest0.test8");  
  
        AddressBook a1 = new AddressBook();  
        AddressBook a2 = new AddressBook();  
        Category a1c = a1.createCategory(a1.getRootCategory(), "a1c");  
        Category a2c = a2.createCategory(a1c, "a2c");  
    }  
}
```



New contact

New category

First name	Last name	E-mail	Phone	Mobile
James S.	Roebuck	JamesSRoe...	561-888-...	561-888-...
Naomi D.	Long	NaomiDLo...	390-822-...	29-112-1...
Karen L.	Lloyd	KarenLLo...	228-76-...	228-76-...
Jean R.	Voigt	JeanRVoig...	610-344-...	610-344-...
Douglas L.	Green	DouglasG...	612-615-...	612-615-...

- All
 - Contractors
 - Customers
 - Employees
 - Suppliers
 - Europe
 - U.S.

First name
 E-Mail

Last name
 Second e-mail

Phone
 Home

Mobile

Notes

112 FAILURES

0 PROBLEMS

Address Book

New contact New category

First name	Last name	E-mail	Phone	Mobile
James S.	Roebuck	JamesSRoe...	561-888-...	561-888-...
Naomi D.	Long	NaomiDLo...	390-12-5...	390-12-1...
Karen L.	Lloyd	KarenLLo...	228-76-1...	228-76-...
Jean R.	Voigt	JeanRVoigt...	610-34...	610-34...
Douglas L.	Green	DouglasLG...	612-61...	612-61...

Categories: All, Contractors, Customers, Employees

Form fields: First name: Karen L., Last name: Lloyd

Calculator

1,245 C

7 8 9 +

4 5 6 -

TerpWord - quotes2.html

File Edit View Font Format Search Insert Table Help

- New Document ^N
- Open Document... ^O
- Print... ^P
- Recently Opened
 - quotes2.html
 - quotes.html
- Save ^S
- Save As...
- Save RTF...

Quotes

TerpSpreadsheet Graph

OPEN PROBLEMS

TerpP...

Datei Bearbeiten Ansicht Format TPBook Zelle Markierung

Neuer Graph*

Slide 1

Today's Keynote:

Search-Based Program Analysis

e

Am = 9

Ri = 44

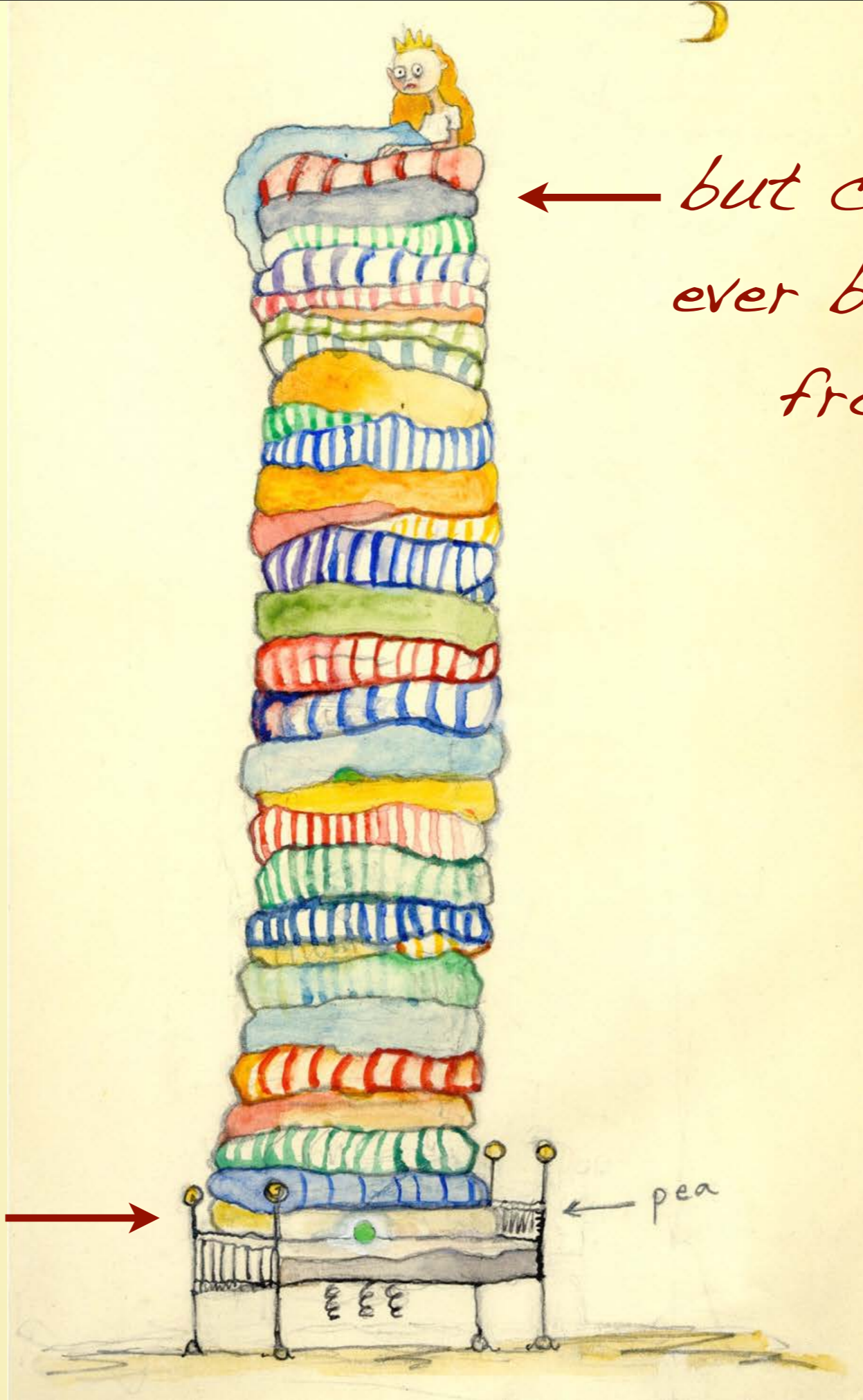
Je = 45

Mic Rob

ram = 44,91 Jeffrey B. M

Lester = 55,33 James M

Plays Pleasant and



← but can these
ever be invoked
from here?

testing
may detect
errors here...

← pea



← test generation
at the system level



← test generation
at the system level

anything that
happens here
is real

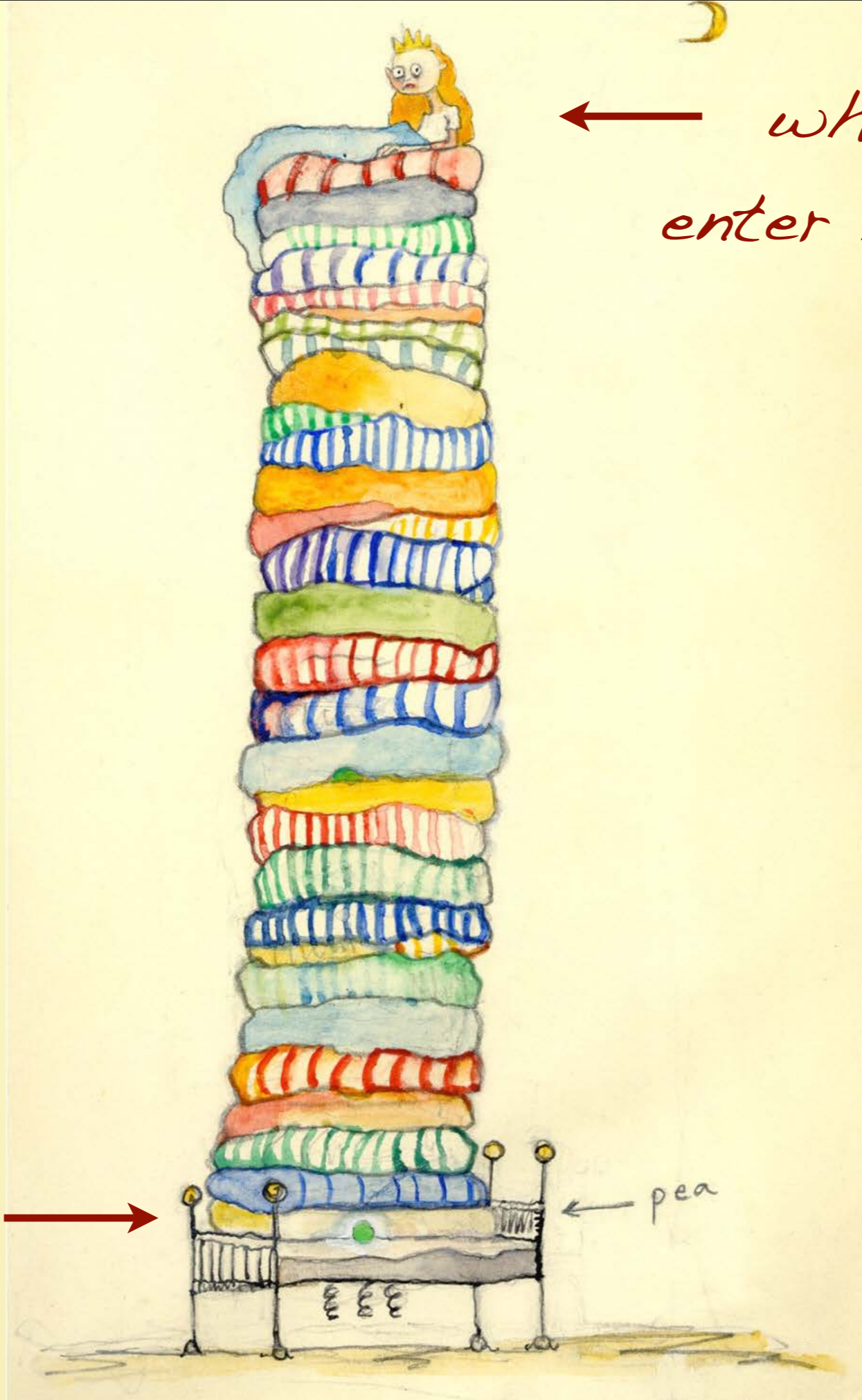
← pea

Getting Coverage

```
static tree
fold_negate_expr (location_t loc, tree t)
{
    // 200 Lines...
    tem = TREE_OPERAND (t, 0);
    if ((INTEGRAL_TYPE_P (type)
        && (TREE_CODE (tem) == NEGATE_EXPR
            || (TREE_CODE (tem) == INTEGER_CST
                && may_negate_without_overflow_p (tem))))
        || !INTEGRAL_TYPE_P (type))
        return fold_build2_loc (loc, TREE_CODE (t), type,
                                negate_expr (tem), TREE_OPERAND (t, 1));
}
```



get me here



← what to enter here?

get me here →

← pea

Infinite Monkey Theorem



Test Generation

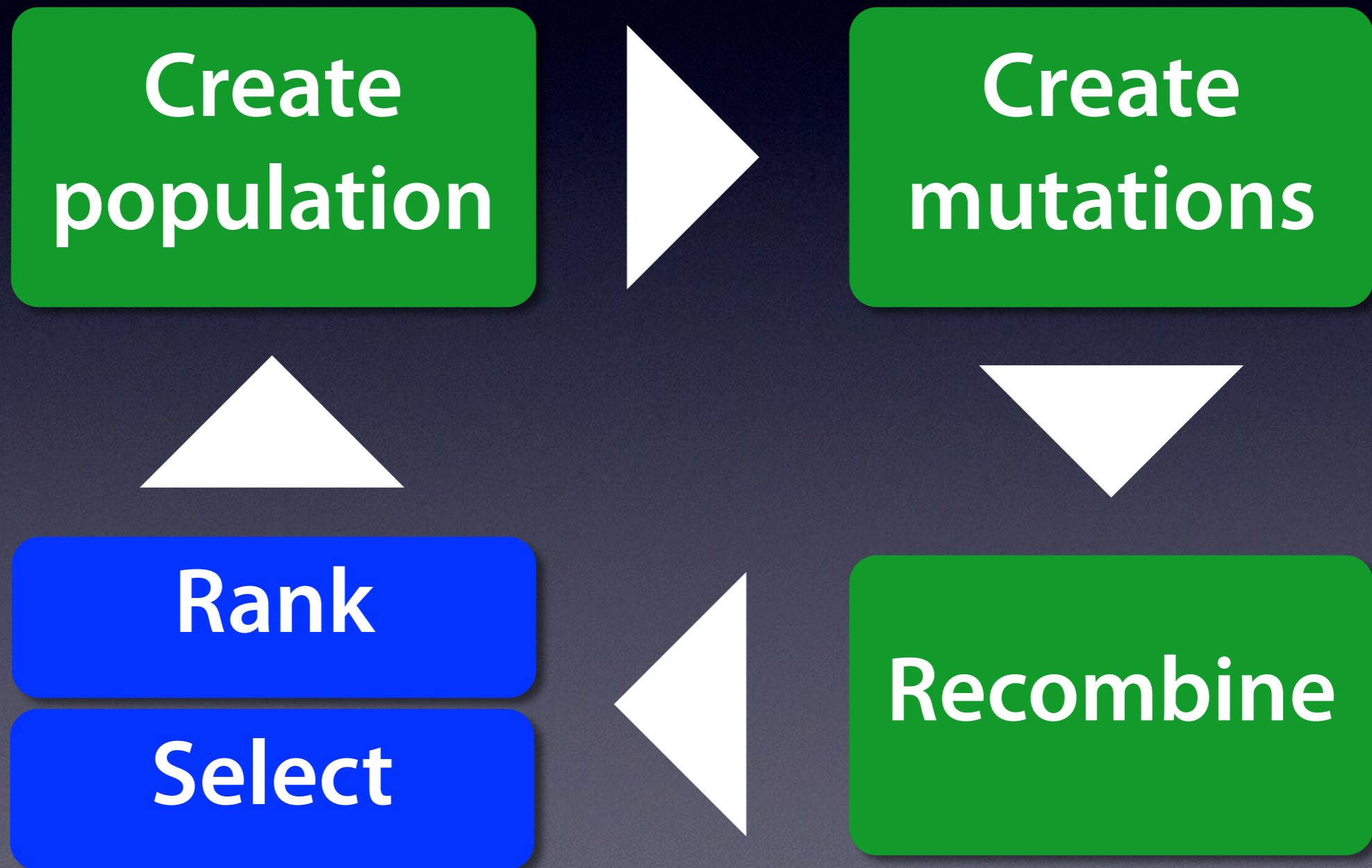
Static analysis and symbolic test generation

- ...work well at the unit level
- ...but miss the surrounding context

System test generation

- ...generates real executions
- ...but misses features at the code level

Evolutionary Algorithms



Create population

“fdsakfh+ew%3gfhdi%4f”

“fwe8^ru786234jä”

Mutation

“fdsakfh+br%3gfhdi%4f”

“fdsakfh+ew%4gfhdi%4f”

“fwe8^ru&26234jä”

“xb3#ru786234jä”

Recombine

“fdsakfh+ew%4gfhdi%4f”

“xb3#ru786234jä”

Create population

“fdsakfh+ew%3gfhdi%4f”

“fwe8^ru786234jä”

Mutation

“fdsakfh+**br**%3gfhdi%4f”

“fdsakfh+ew%**4**gfhdi%4f”

“fwe8^ru**&2**6234jä”

“**xb3**#ru786234jä”

Recombine

“fdsakfh+ew%**4**gfhdi%4f”

“**xb3**#ru786234jä”

“**xb3**#ru7%**4**gfhdi%4f”

Selection and Ranking

if (angle = 47 \wedge power = 532) { ... }

“fdsakfh+ew%4gfhdi%4f”

angle = 31

“xb3#ru786234jä”

angle = 48

“xb3#ru7%4gfhdi%4f”

angle = 65

Selection and Ranking

if (angle = 47 \wedge power = 532) { ... }

angle = 51

“xb3#ru786234jä”

angle = 48

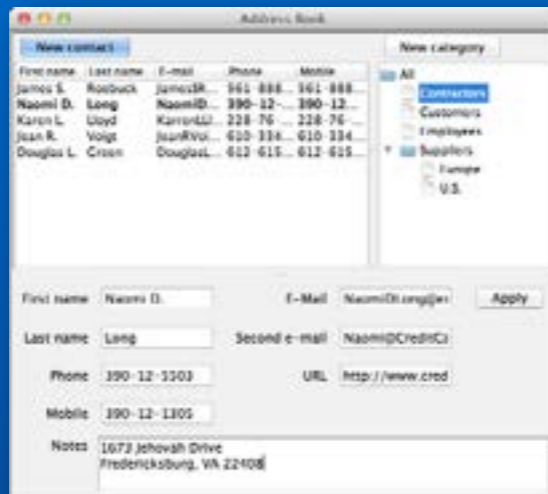
angle = 47

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



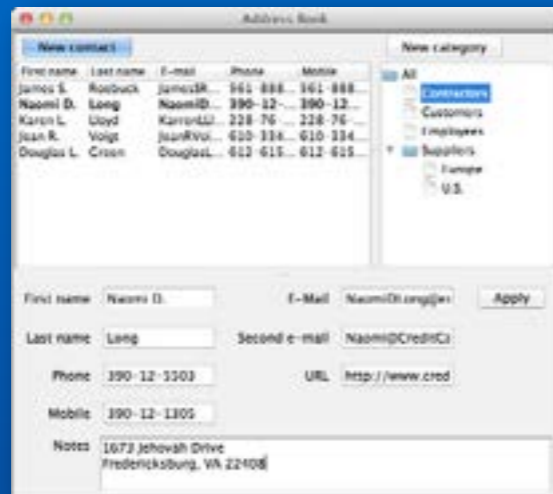
Web Tests

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



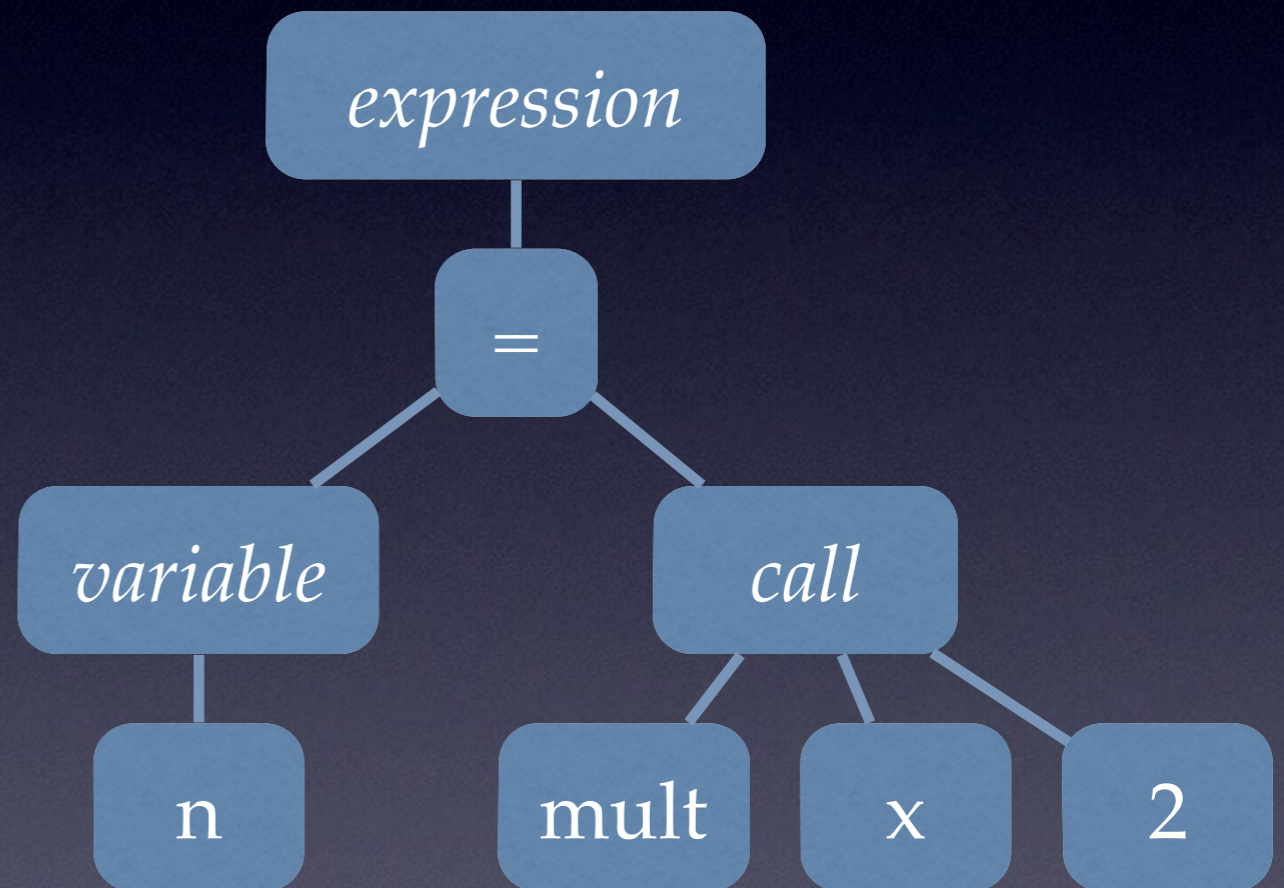
Web Tests

Testing a C Compiler



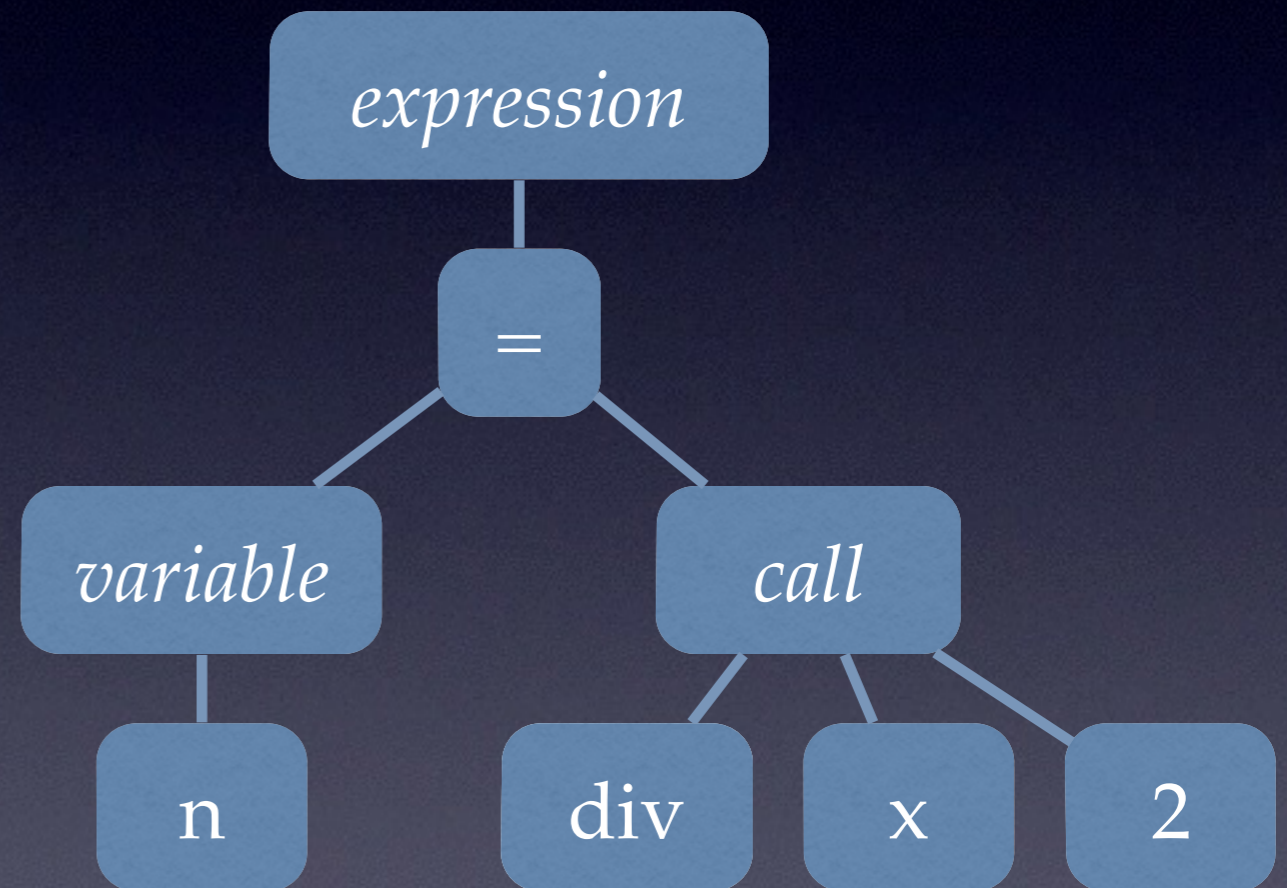
Parsing C Files

`n = mult(x, 2);`



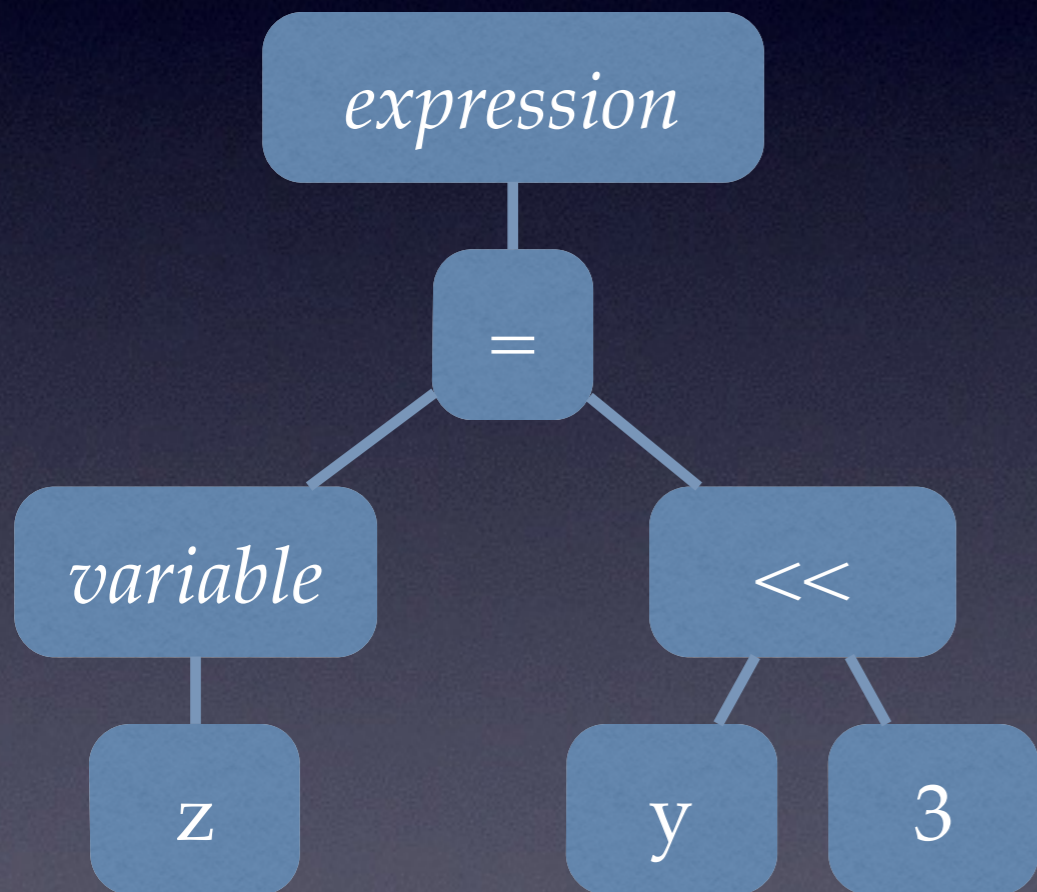
Mutation

`n = div(x, 2);`

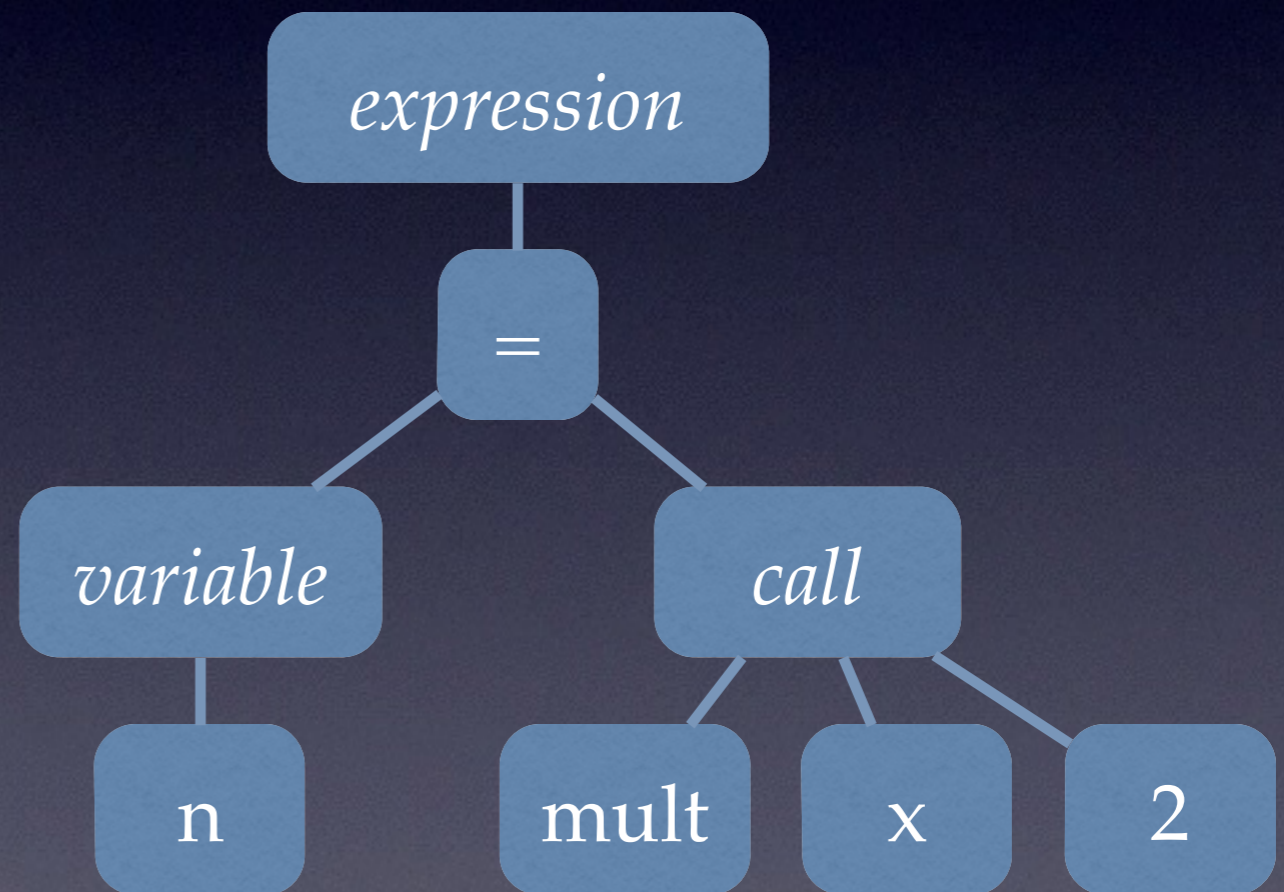


Recombination

`z = y << 3;`

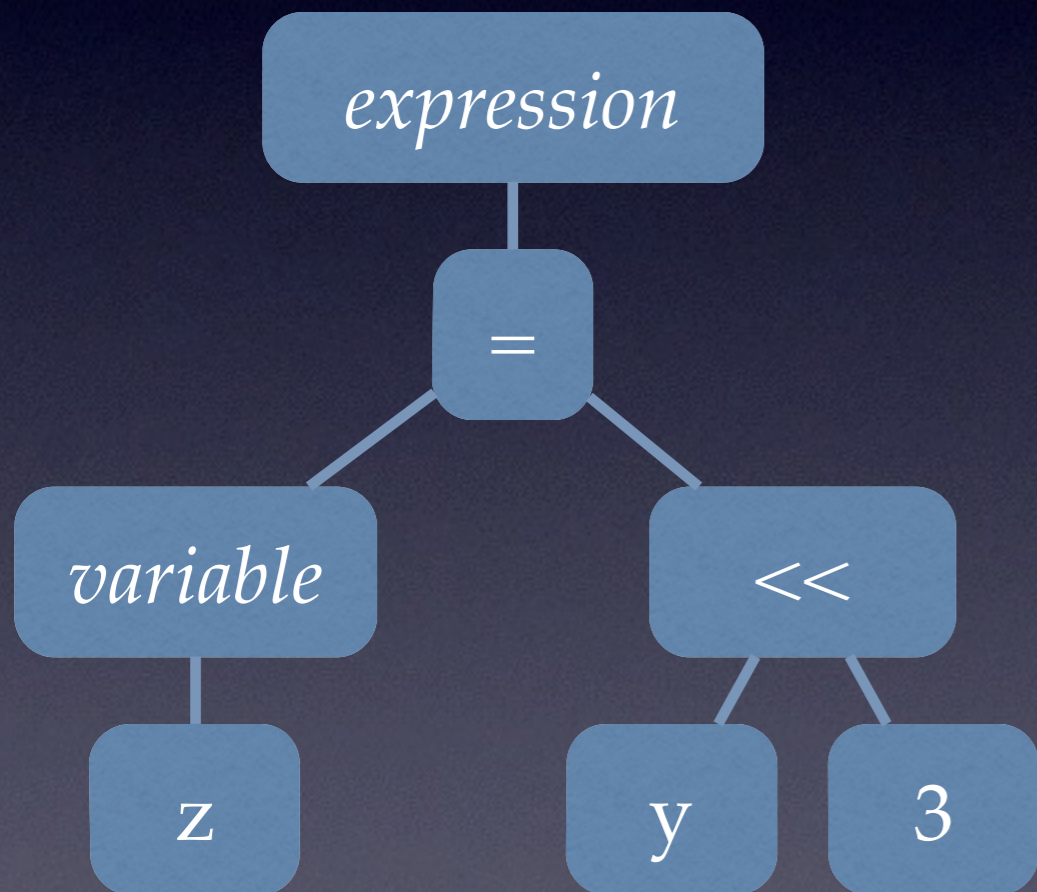


`n = mult(x, 2);`

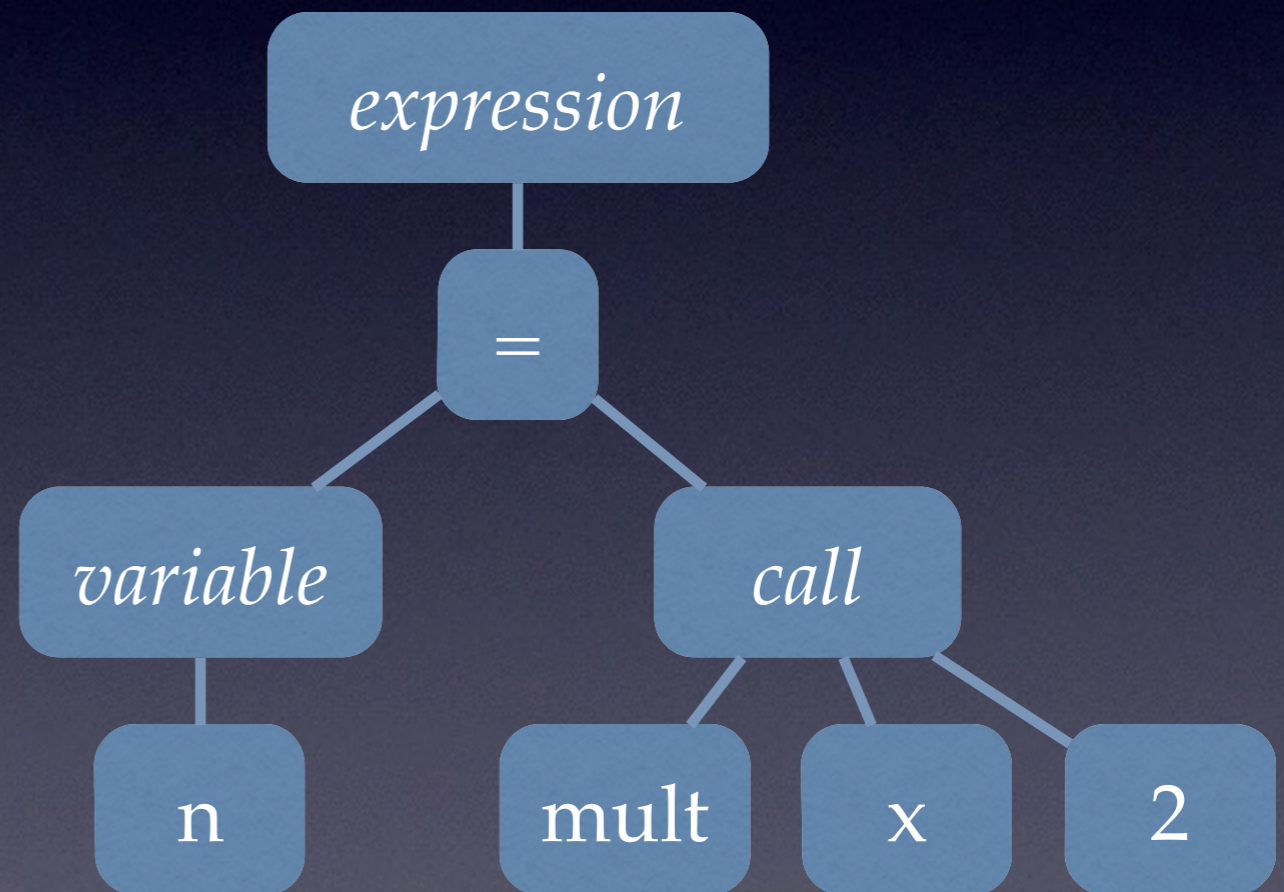


Recombination

```
z = y << 3;
```

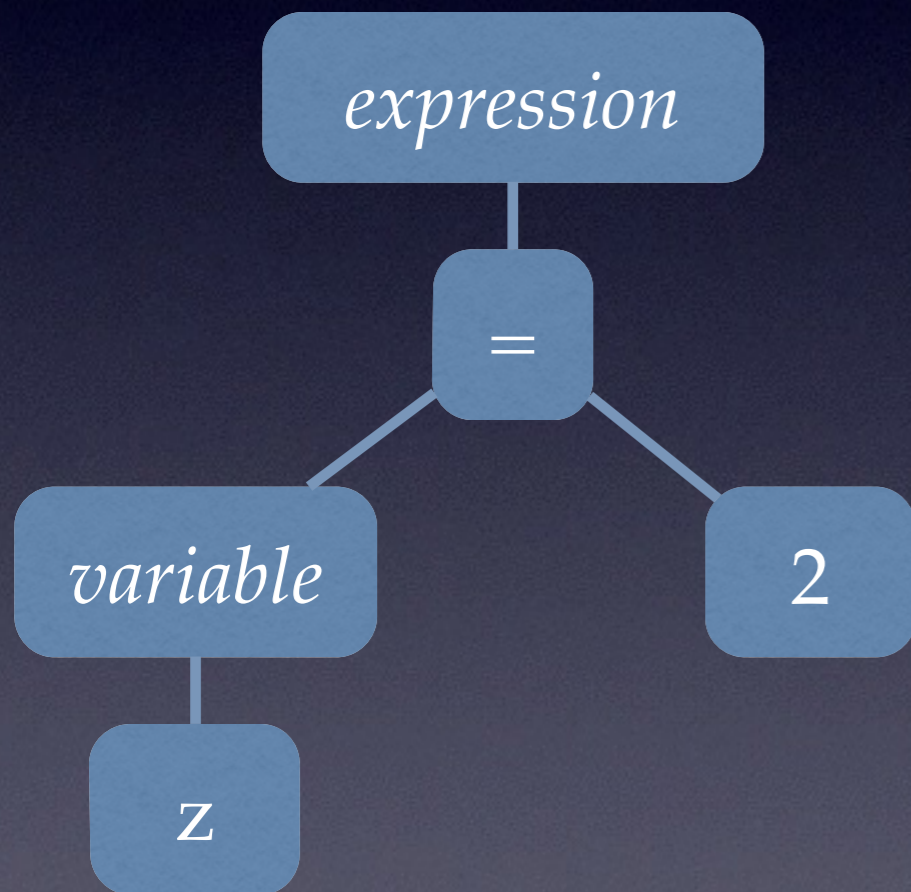


```
n = mult(x, 2);
```

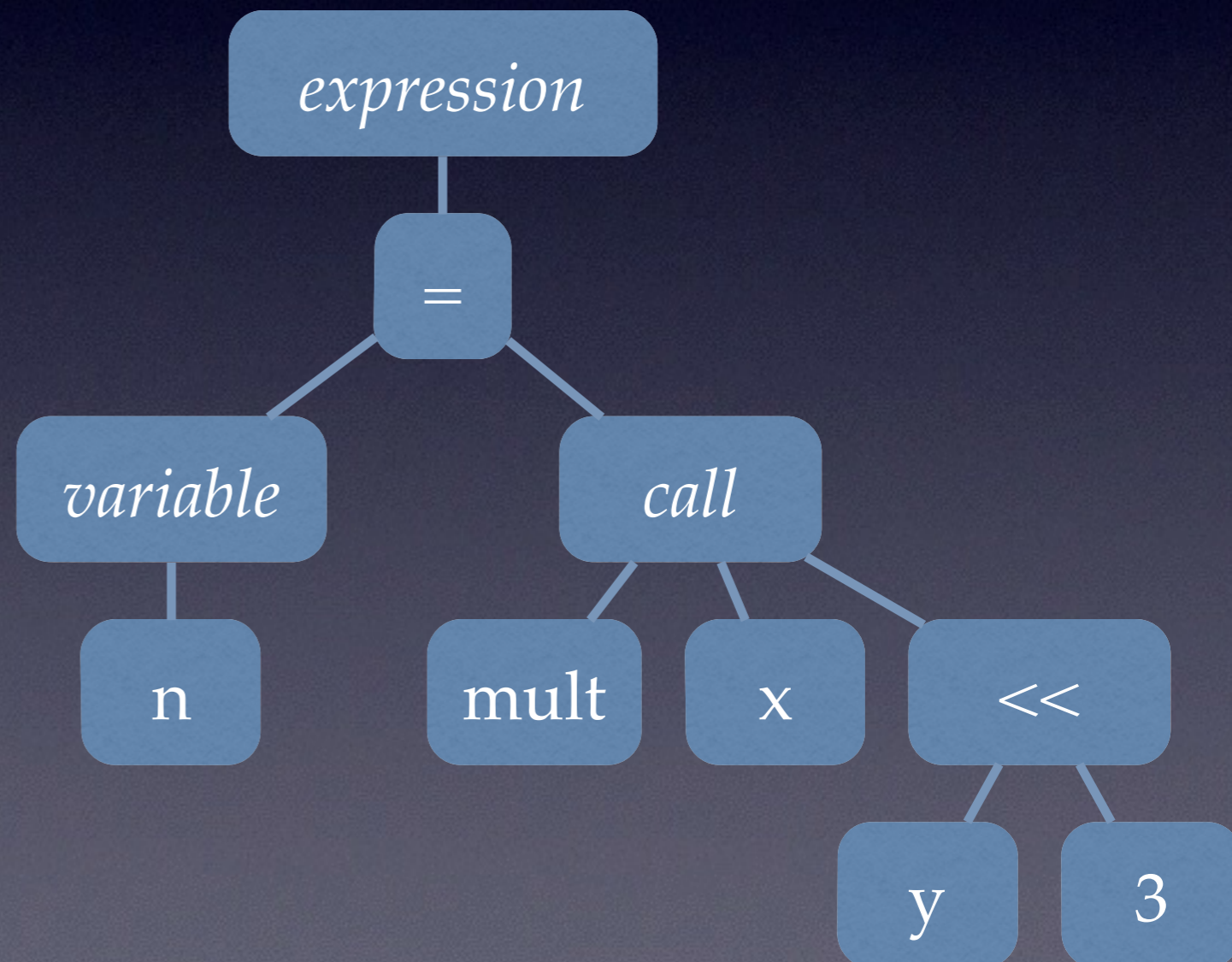


Recombination

`z = 2;`



`n = mult(x, y << 3);`



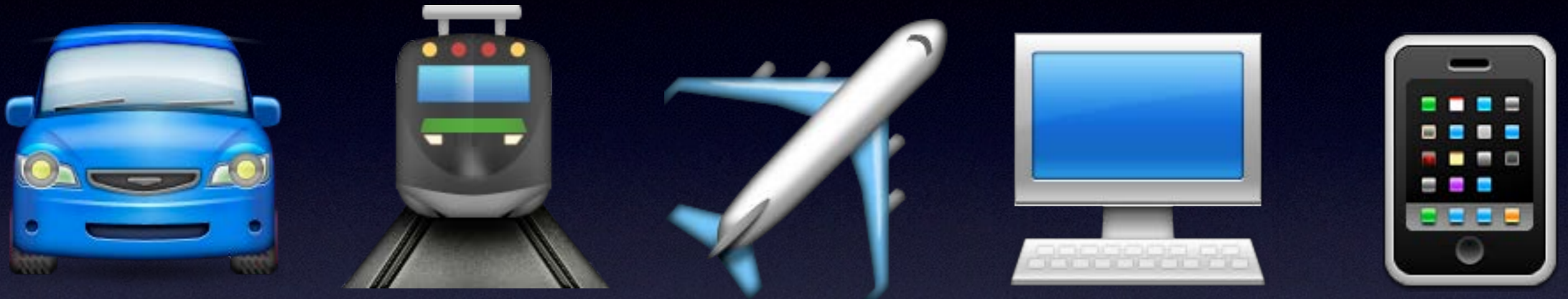
Testing a C Compiler



Evolved C Input

```
int main() {
    double var0 = -474411016.5983948;
    double var1 = var0;
    var1 /= (var0 + 842286918.0743376 * var0) + 1757571303008385195L;
    unsigned long var2 = 7608230537870159477UL;
    var2 *= var2 * var2;
    int var3 = -1522999176;
    for (; var3 <= -1522999137; ++var3) {
        var1 = var1 / -3315559715349753910LL /
            (var3 & 7709094732231684709ULL) == -1854837480 / 2075989423U;
        unsigned long long var4 = 17274394483946351480ULL;
        int var5 = 1690262972;
        var5 += (4 / var4 % var2 & 8151340956687106979ULL) >= 36423 / -24124 % 120;
    }
    int var4 = 1979470929;
    var4 |= var3 - -4708;
    signed char var5 = 22;
    unsigned char var6 = 7;
    var5 = var5 + var6;
}
```

Domain Code

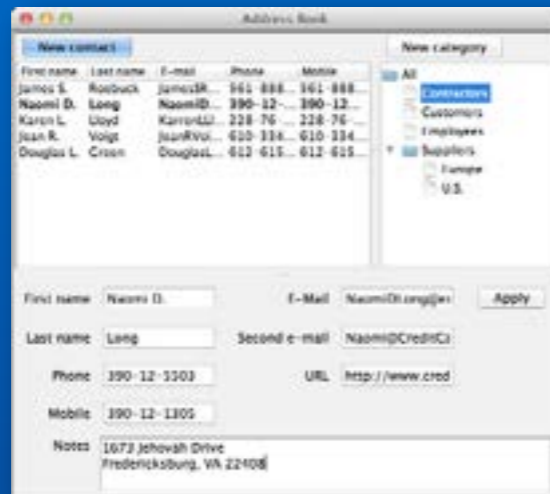



```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



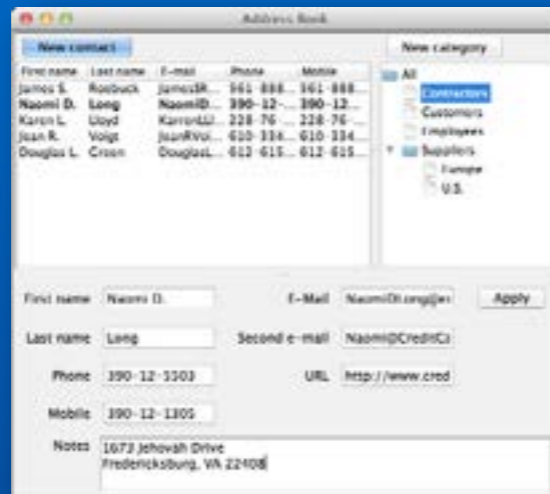
Web Tests

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



Web Tests

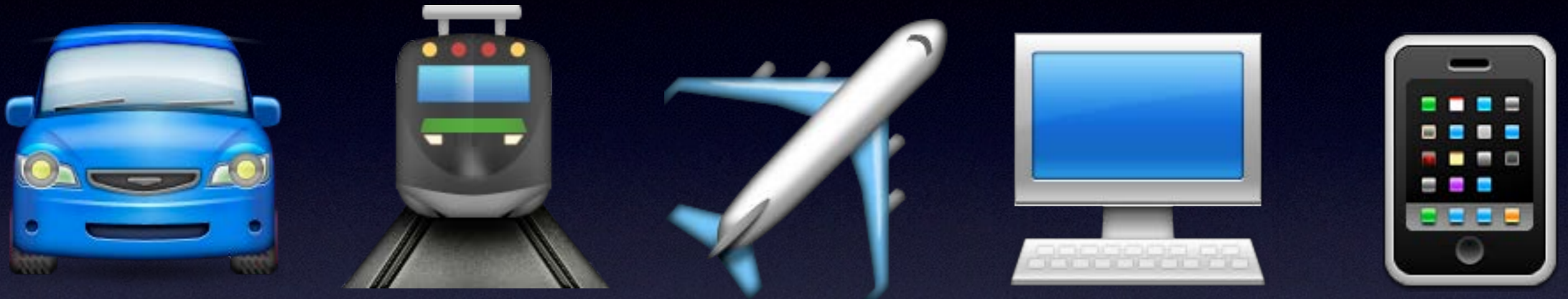
XML Tests

```
<math>
<mrow>
  <munderover>
    <mo>&sum;</mo>
    <mrow>
      <mi>x</mi>
      <mo>=</mo>
      <mn>1</mn>
    </mrow>
    <mi>n</mi>
  </munderover>
  <mrow>
    <apply>
      <power/>
      <ci>x</ci>
      <cn>2</cn>
    </apply>
  </mrow>
</mrow>
</math>
```



$$\sum_{x=1}^n x^2$$

Domain Inputs



XMLTest



XML Input



Tested System

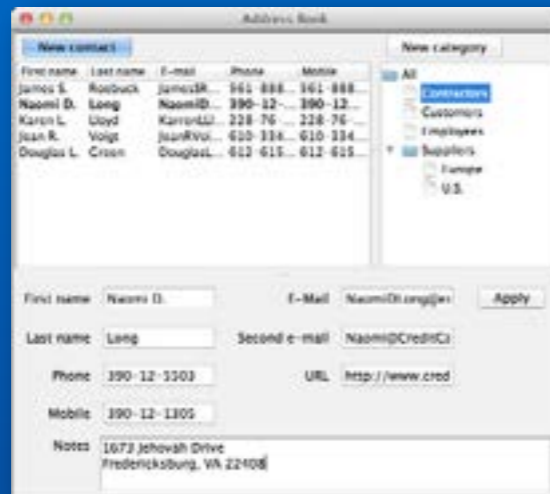


```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



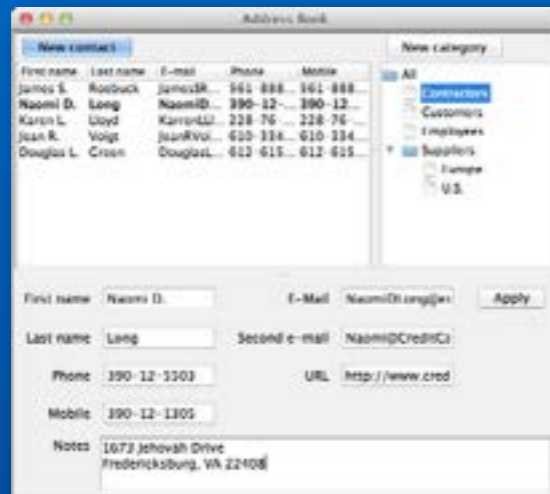
Web Tests

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests

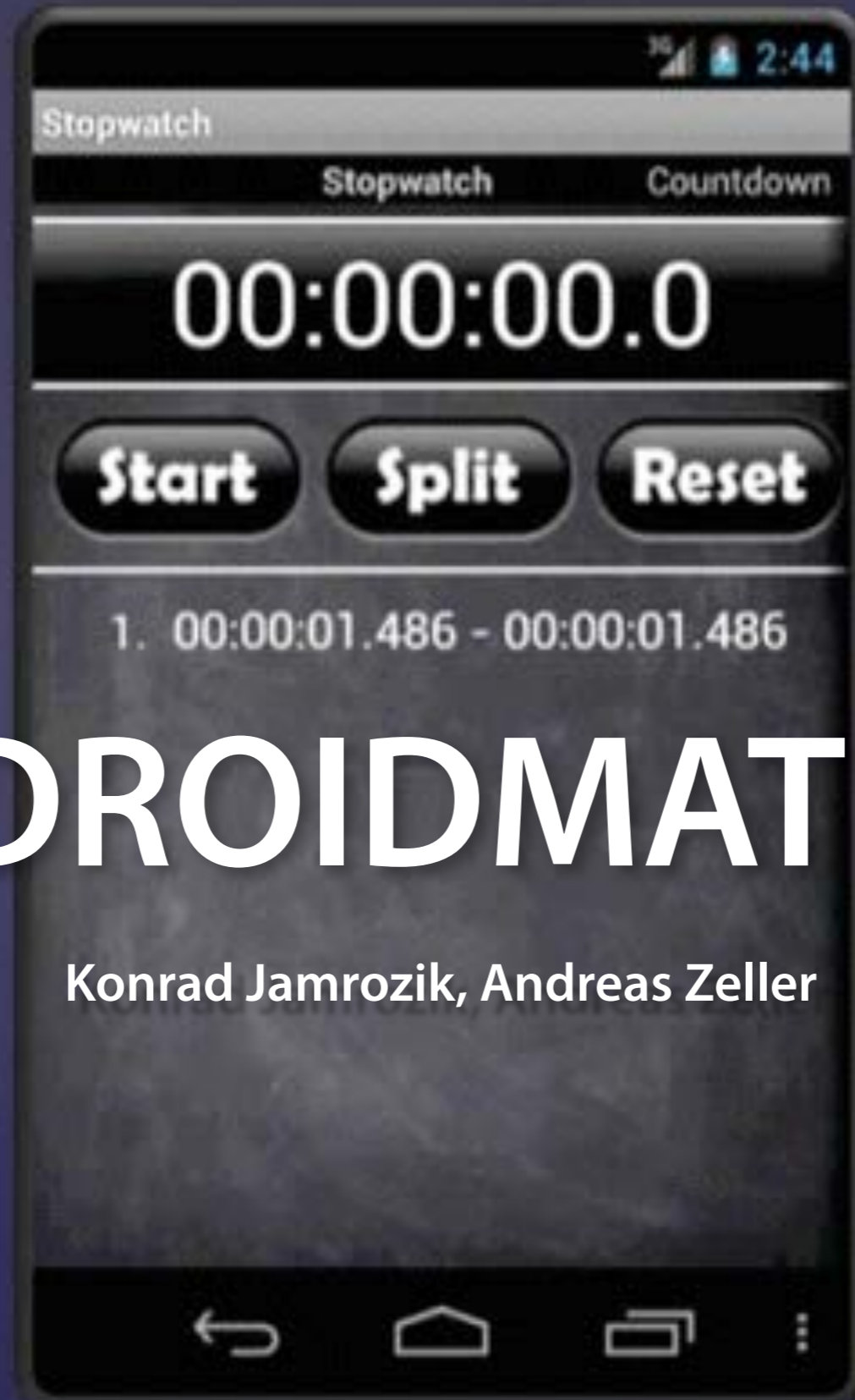


Web Tests



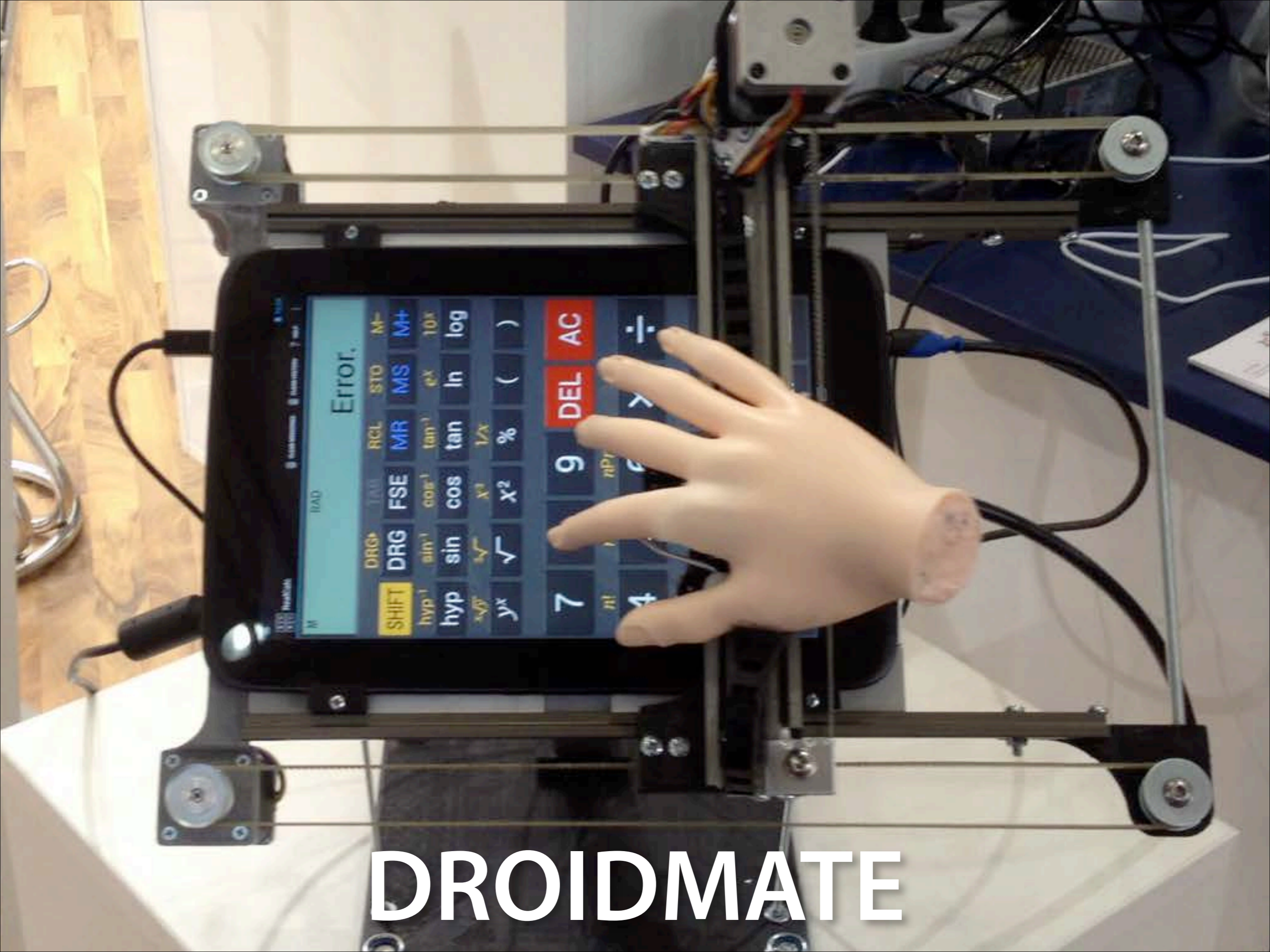
EXSYST

Florian Groß, Andreas Zeller



DROIDMATE

Konrad Jamrozik, Andreas Zeller



DROIDMATE

Test Coverage

Unit Test Generators

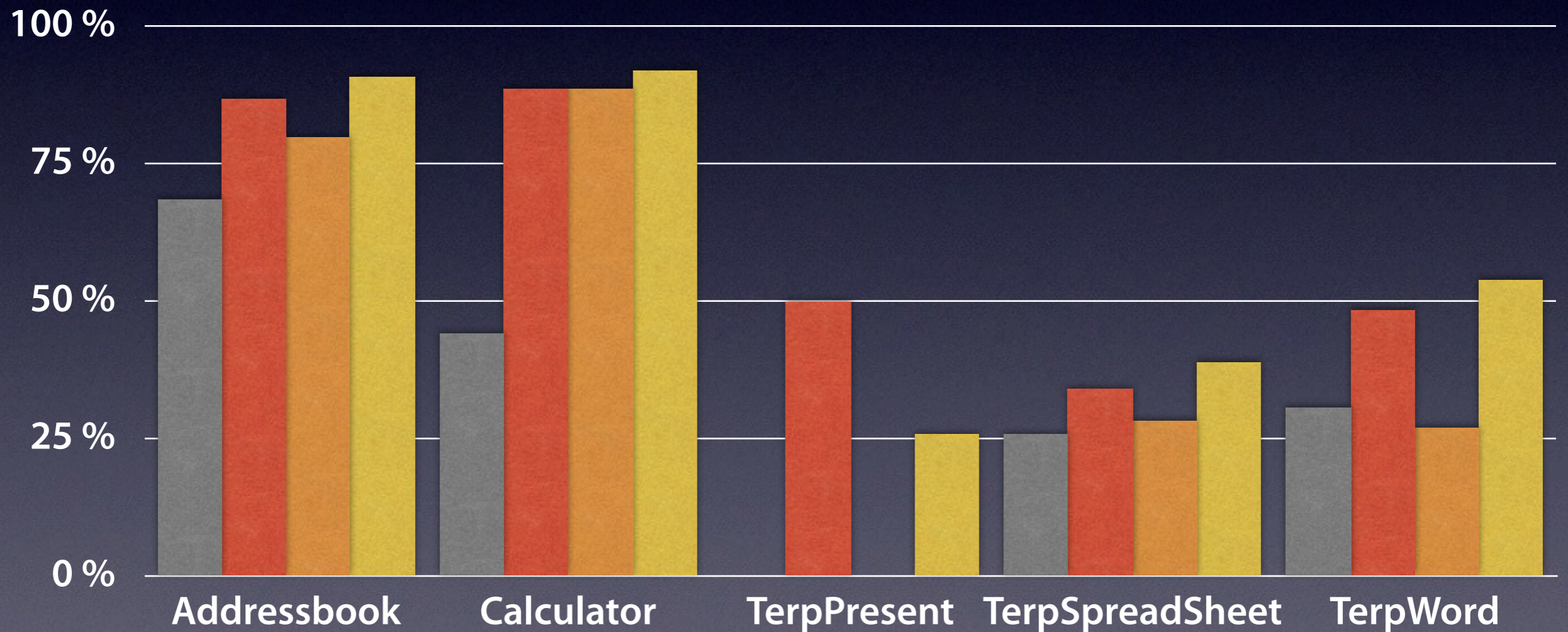
■ Randoop

■ Evosuite

GUI Test Generators

■ GUITar

■ Exsyst





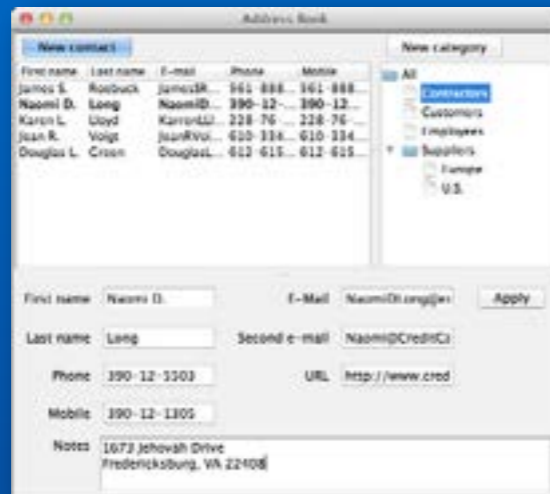


```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



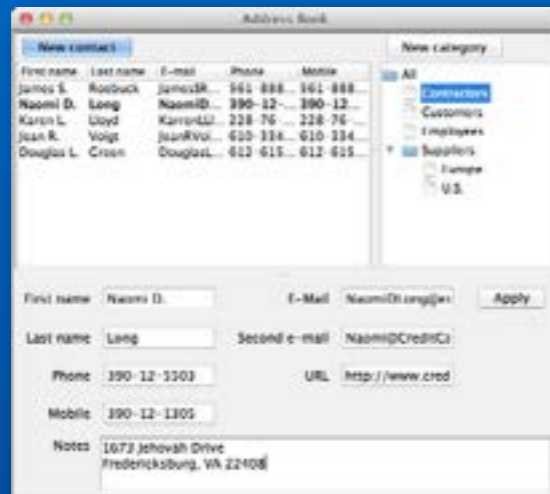
Web Tests

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



Web Tests



facebook

E-Mail Passwort

Angemeldet bleiben [Passwort vergessen?](#)

Facebook ermöglicht es dir, mit den Menschen in deinem Leben in Verbindung zu treten und Inhalte mit diesen zu teilen.

WEBMATE

Martin Burger, Valentin Dallmeier, Andreas Zeller



Registrieren

Facebook ist und bleibt kostenlos.

Vorname:

Nachname:

Diese E-Mail-Adresse:

E-Mail nochmals eingeben:

Neues Passwort:

Ich bin:

Geburtsdag: Tag: Monat: Jahr:

Warum muss ich meinen Geburtstag angeben?

Wenn du auf „Registrieren“ klickst, akzeptierst du unsere Nutzungsbedingungen und erklärst unsere Datenverwendungsrichtlinien gelesen und verstanden zu haben.





Startseite

Softwaretesten für Web 2.0-Anwendungen

Die Testfabrik AG ist Ihr Partner, wenn es um das automatische Testen von Webanwendungen geht.

WebMate ist derzeit das weltweit einzige Werkzeug zur vollständigen und vollautomatischen Prüfung auf Cross-Browser-Kompatibilität von Web 2.0 Anwendungen. WebMate entsteht direkt aus der Spitzenforschung des Lehrstuhles für Softwaretechnik von Prof. Dr. Andreas Zeller in Saarbrücken.

Wenn Sie und Ihr Business darauf angewiesen sind, dass Ihre Webanwendung fehlerfrei funktioniert, bieten wir Ihnen die richtigen Testlösungen dazu an – maßgeschneidert auf Ihre Bedürfnisse und mit einem Return on Investment, der Sie positiv überraschen wird. Informieren Sie sich [vorab online](#) oder [sprechen Sie uns direkt an](#).

Wir machen das Testen im Web 2.0 rentabel!



Dr. Martin Burger, Dr. Valentin Dallmeier, Michael Mirolid, Bernd Pohl und Prof. Dr. Andreas Zeller erklären die Geschäftsidee hinter der Testfabrik AG.

Wir stellen ein Verstärkung gesucht

Wir suchen *Softwareentwickler (w/m)*, *Software-Testberater (w/m)* und *Studentische Hilfskräfte (w/m)* zur Verstärkung unserer Produktentwicklung sowie unseres Berater-Teams. Interesse? Unter [Karriere](#) gibt es weiterführende Informationen.

CISPA
Center for Information Security

Über CISPA | Forschung | Existenzstudien | Ausbildung | Events | Kontakt

Saarbrücker Informatik
Zusätzlich zu den an CISPA durchgeführten Instituten (ZIT, IPRIT, ITH-DWS, DPK) kooperiert das CISPA mit einer Reihe von Max-Planck-Forschungsinstituten.

Exzellenzcluster Multimodal Computing and Interaction
Die DFG geförderte Exzellenzcluster Multimodal Computing and Interaction (MCI) kooperiert mit der Grundlagenforschung und Technologieentwicklung in allen Bereichen der multimodalen Kommunikation und Interaktion zwischen Mensch und Maschine. IT-Sicherheit ist ein zentraler Bestandteil des Clusters.
Weitere Infos finden Sie unter: www.mci.uni-saarland.de

DFG Graduiertenschule Saarbrücken Graduate School of Computer Science
Die Saarbrücker Graduiertenschule für Informatik (SaarIT) fördert herausragende Talente in diesem Forschungsbereich und stellt für Spitzenleistungen auf internationaler Ebene ein. Studierende erhalten eine forschungsorientierte Ausbildung in einer wissensintensiven, interdisziplinären und anpassungsstarken Atmosphäre und werden von international renommierten Wissenschaftlern betreut. Das Doktorandenprogramm der Graduiertenschule wird auf Englisch angeboten und von der deutschen Exzellenzinitiative unterstützt. Neben dem Fachbereich Informatik der Universität des Saarlandes ist international hervorragende Forschungsinfrastruktur an der Leitung der Graduiertenschule beteiligt.
Weitere Informationen dazu finden Sie unter: www.graduiertenschule.uni-saarland.de

Intel Visual Computing Institute
Das Intel Visual Computing Institute (IVI) ist ein Kooperationsprojekt zwischen Intel, der Universität des Saarlandes, der Leibniz-Universität Hannover (LUNH), der RWTH Aachen und dem IPRIT mit dem Schwerpunkt auf Visual Computing, also Erstellung, Modifizierung, Verarbeitung, Übertragung, Rendering und Darstellung von 2D- und 3D-Computergrafiken.
Weitere Infos finden Sie unter: www.intel-ivi.uni-saarland.de

Intel Visual Computing Institute
Das Intel Visual Computing Institute (IVI) ist ein Kooperationsprojekt zwischen Intel, der Universität des Saarlandes, der Leibniz-Universität Hannover (LUNH), der RWTH Aachen und dem IPRIT mit dem Schwerpunkt auf Visual Computing, also Erstellung, Modifizierung, Verarbeitung, Übertragung, Rendering und Darstellung von 2D- und 3D-Computergrafiken.
Weitere Infos finden Sie unter: www.intel-ivi.uni-saarland.de

Calendar: 1st week of 2012, 1st week of 2012.

CISPA
Center for Information Security

Über CISPA | Forschung | Existenzstudien | Ausbildung | Events | Kontakt

Saarbrücker Informatik
Zusätzlich zu den an CISPA durchgeführten Instituten (ZIT, IPRIT, ITH-DWS, DPK) kooperiert das CISPA mit einer Reihe von Max-Planck-Forschungsinstituten.

Exzellenzcluster Multimodal Computing and Interaction
Die DFG geförderte Exzellenzcluster Multimodal Computing and Interaction (MCI) kooperiert mit der Grundlagenforschung und Technologieentwicklung in allen Bereichen der multimodalen Kommunikation und Interaktion zwischen Mensch und Maschine. IT-Sicherheit ist ein zentraler Bestandteil des Clusters.
Weitere Infos finden Sie unter: www.mci.uni-saarland.de

DFG Graduiertenschule Saarbrücken Graduate School of Computer Science
Die Saarbrücker Graduiertenschule für Informatik (SaarIT) fördert herausragende Talente in diesem Forschungsbereich und stellt für Spitzenleistungen auf internationaler Ebene ein. Studierende erhalten eine forschungsorientierte Ausbildung in einer wissensintensiven, interdisziplinären und anpassungsstarken Atmosphäre und werden von international renommierten Wissenschaftlern betreut. Das Doktorandenprogramm der Graduiertenschule wird auf Englisch angeboten und von der deutschen Exzellenzinitiative unterstützt. Neben dem Fachbereich Informatik der Universität des Saarlandes ist international hervorragende Forschungsinfrastruktur an der Leitung der Graduiertenschule beteiligt.
Weitere Informationen dazu finden Sie unter: www.graduiertenschule.uni-saarland.de

Intel Visual Computing Institute
Das Intel Visual Computing Institute (IVI) ist ein Kooperationsprojekt zwischen Intel, der Universität des Saarlandes, der Leibniz-Universität Hannover (LUNH), der RWTH Aachen und dem IPRIT mit dem Schwerpunkt auf Visual Computing, also Erstellung, Modifizierung, Verarbeitung, Übertragung, Rendering und Darstellung von 2D- und 3D-Computergrafiken.
Weitere Infos finden Sie unter: www.intel-ivi.uni-saarland.de

Intel Visual Computing Institute
Das Intel Visual Computing Institute (IVI) ist ein Kooperationsprojekt zwischen Intel, der Universität des Saarlandes, der Leibniz-Universität Hannover (LUNH), der RWTH Aachen und dem IPRIT mit dem Schwerpunkt auf Visual Computing, also Erstellung, Modifizierung, Verarbeitung, Übertragung, Rendering und Darstellung von 2D- und 3D-Computergrafiken.
Weitere Infos finden Sie unter: www.intel-ivi.uni-saarland.de



Test this!

www.facebook.com

Test

4309 2341 6539 231

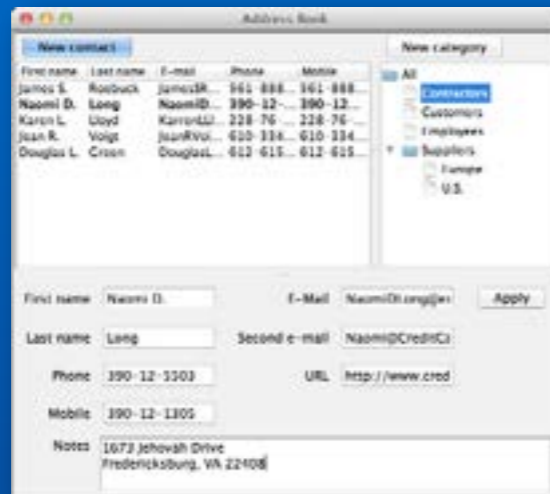
Subscribe

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



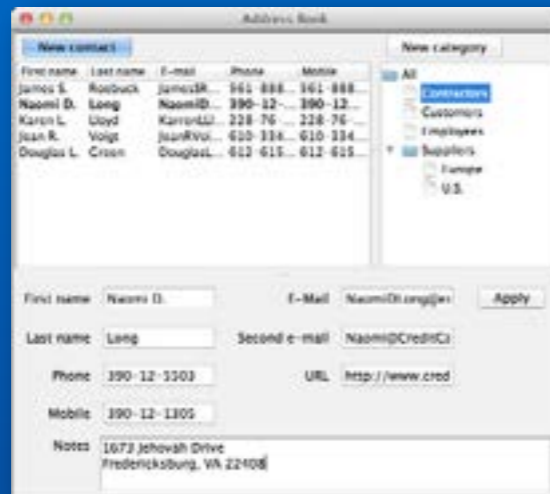
Web Tests

```
int main() { ... }
```

CC Tests



XML Tests



GUI Tests



Web Tests

Checking Results

Crashes

Assertions

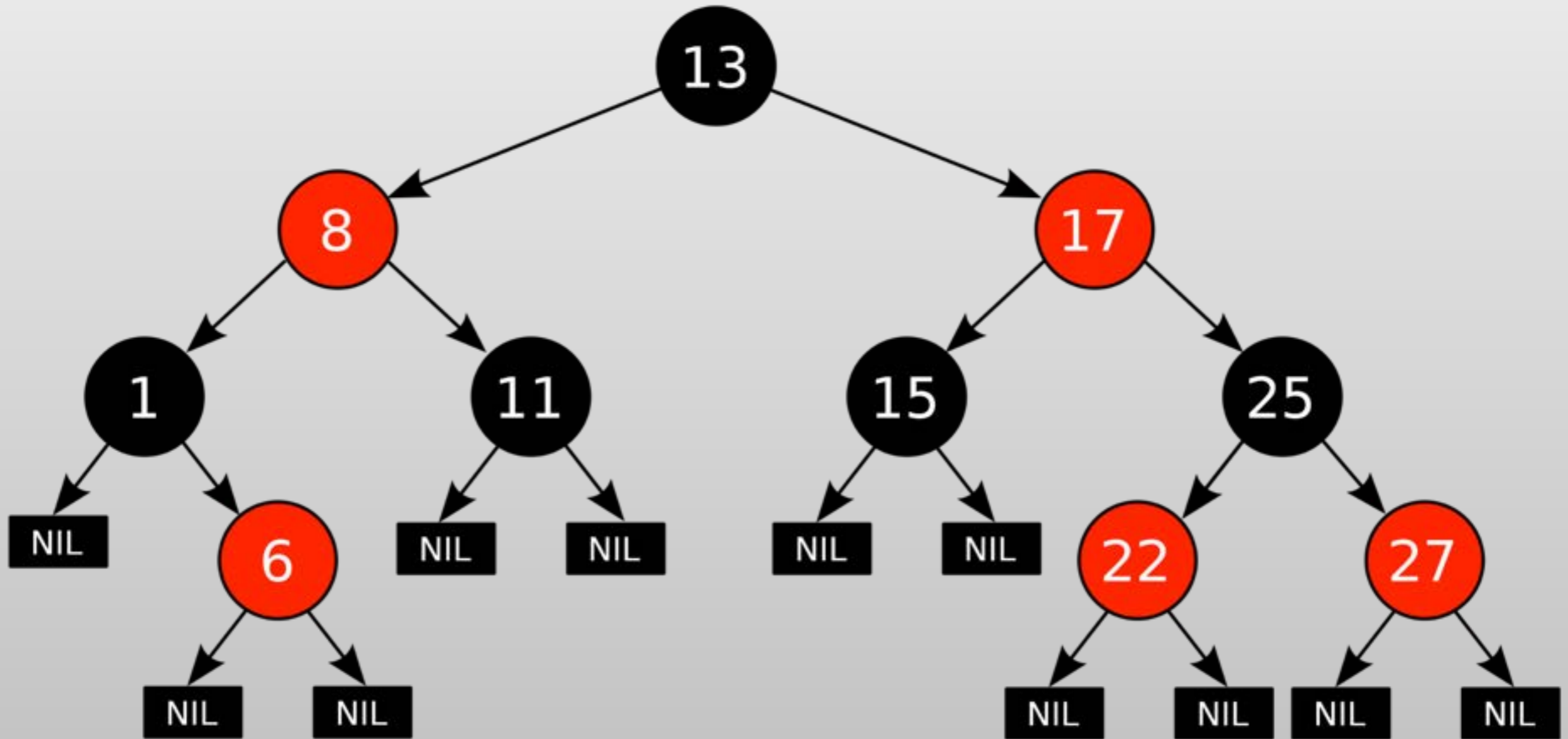
Regressions

Crashes



If the system detects an error, we do so, too

Assertions



Assertions

```
class RedBlackTree {  
    ...  
    boolean repOk() {  
        assert (rootHasNoParent());  
        assert (rootIsBlack());  
        assert (redNodesHaveOnlyBlackChildren());  
        assert (equalNumberOfBlackNodesOnSubtrees());  
        assert (treeIsAcyclic());  
        assert (parentsAreConsistent());  
  
        return true;  
    }  
}
```

Test generators aim for *branch coverage* –
and thus aim for falsifying assertions

Regression

$$\sum_{x=1}^n x^2$$

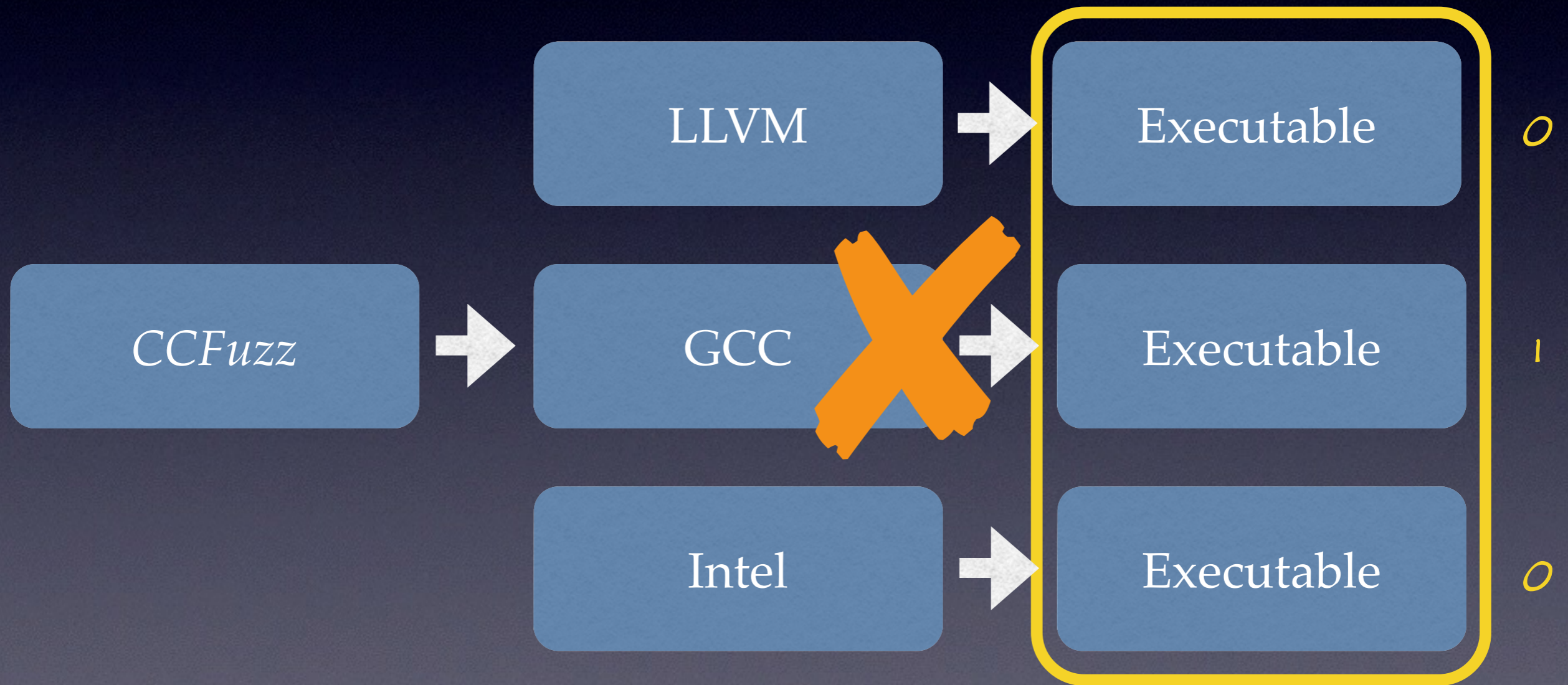
$$x=1$$

$$\sum_{x=1}^n x^3$$

$$x=1$$

Compare with *earlier* or *alternate* version

Compare Compilers



App Mining



Layer Oracles

System interface (XML, etc.)

Java code

C code

Layer Oracles

System interface (XML, etc.)

in = "<x>20</x>"

Java code

sx = "20"

```
int f(int x) { return 2 * x; }  
void h(int x, int y) {  
    if (x != y && f(x) == x + 10)  
        abort();  
}
```

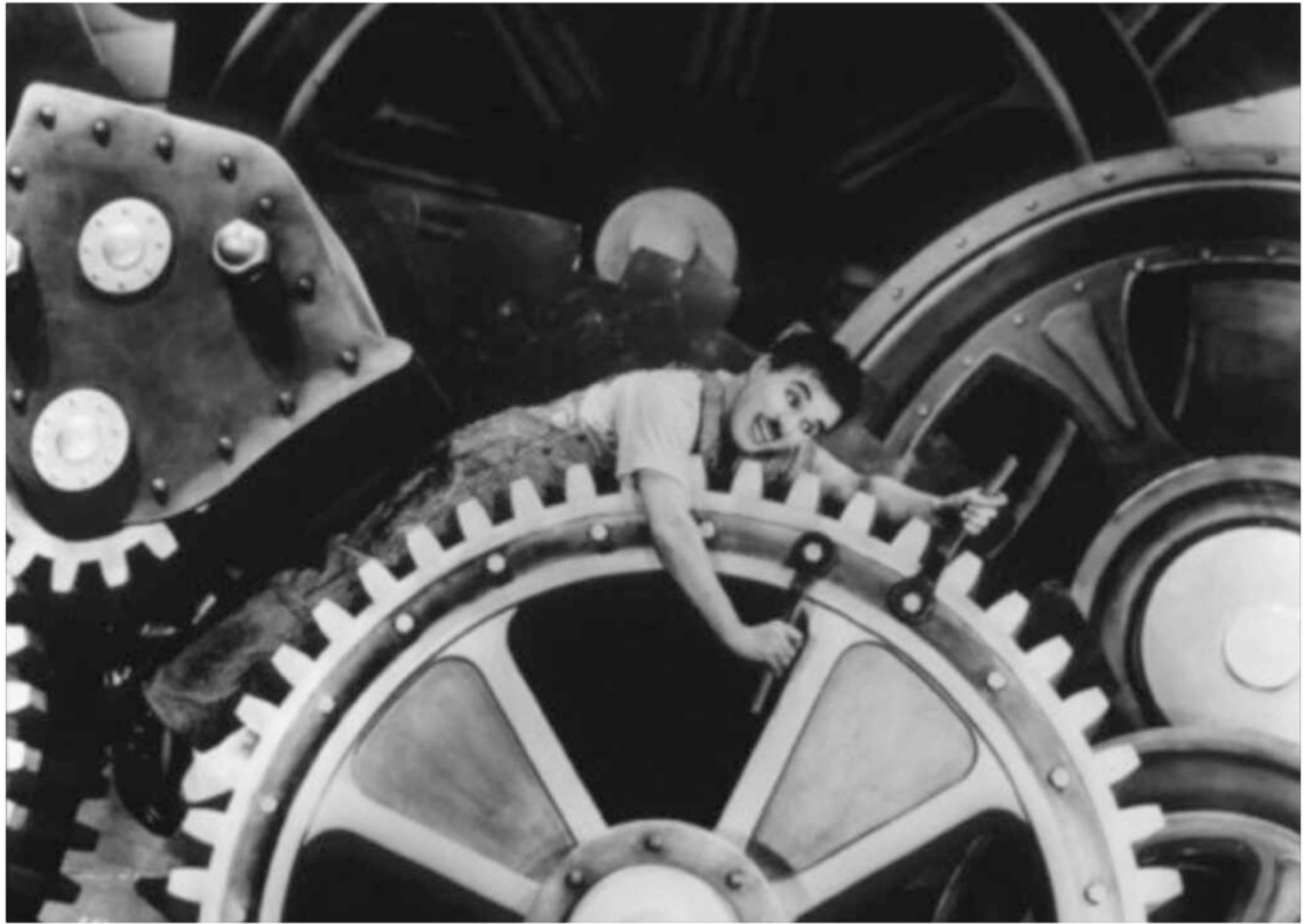
x = 20, y = 1
x = 20, y != x

Layer Oracles

System interface (XML, etc.)

Java code

C code







A Rau

V Dallmeier

S Just

E May

F Gross

K Herzig

C Hammacher

A Zeller

M Burger

A Tarasevich

K Streit

M Höschele

B Pohl

M Mirolid

I Tavecchia

G Reibold

J Rößler

K Jamrozik

JP Galeotti

K Becker

A Gorla



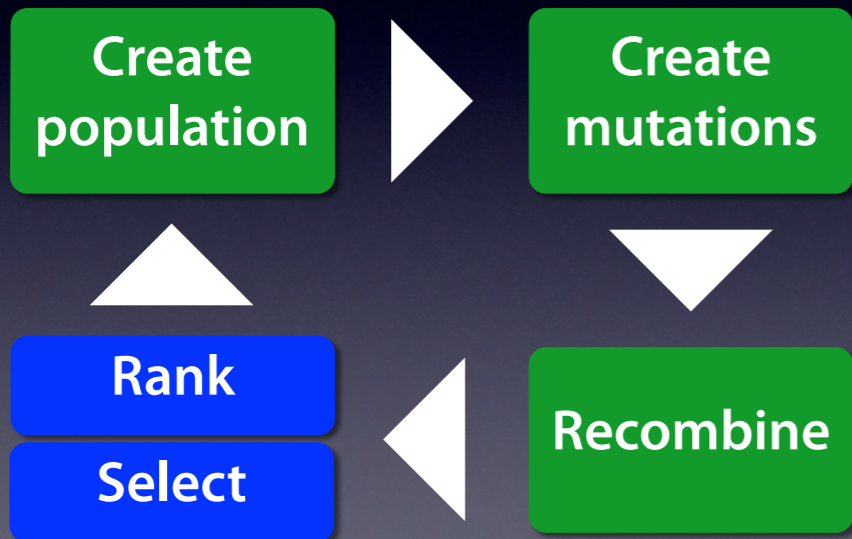
Software is manifold



static analysis
may detect errors here...



Evolutionary Algorithms



`int main() { ... }`
CC Tests

XML Tests

GUI Tests

Web Tests