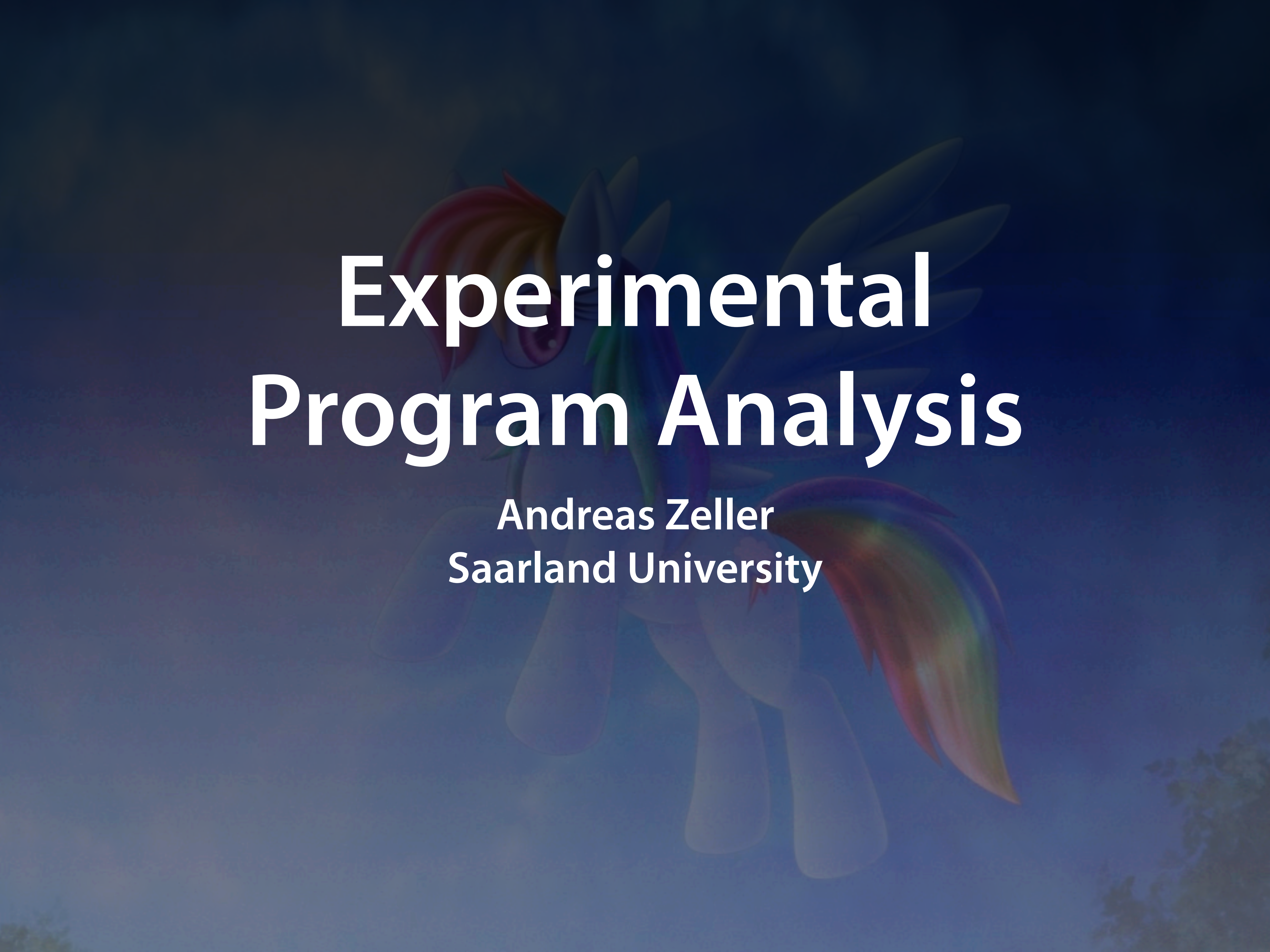
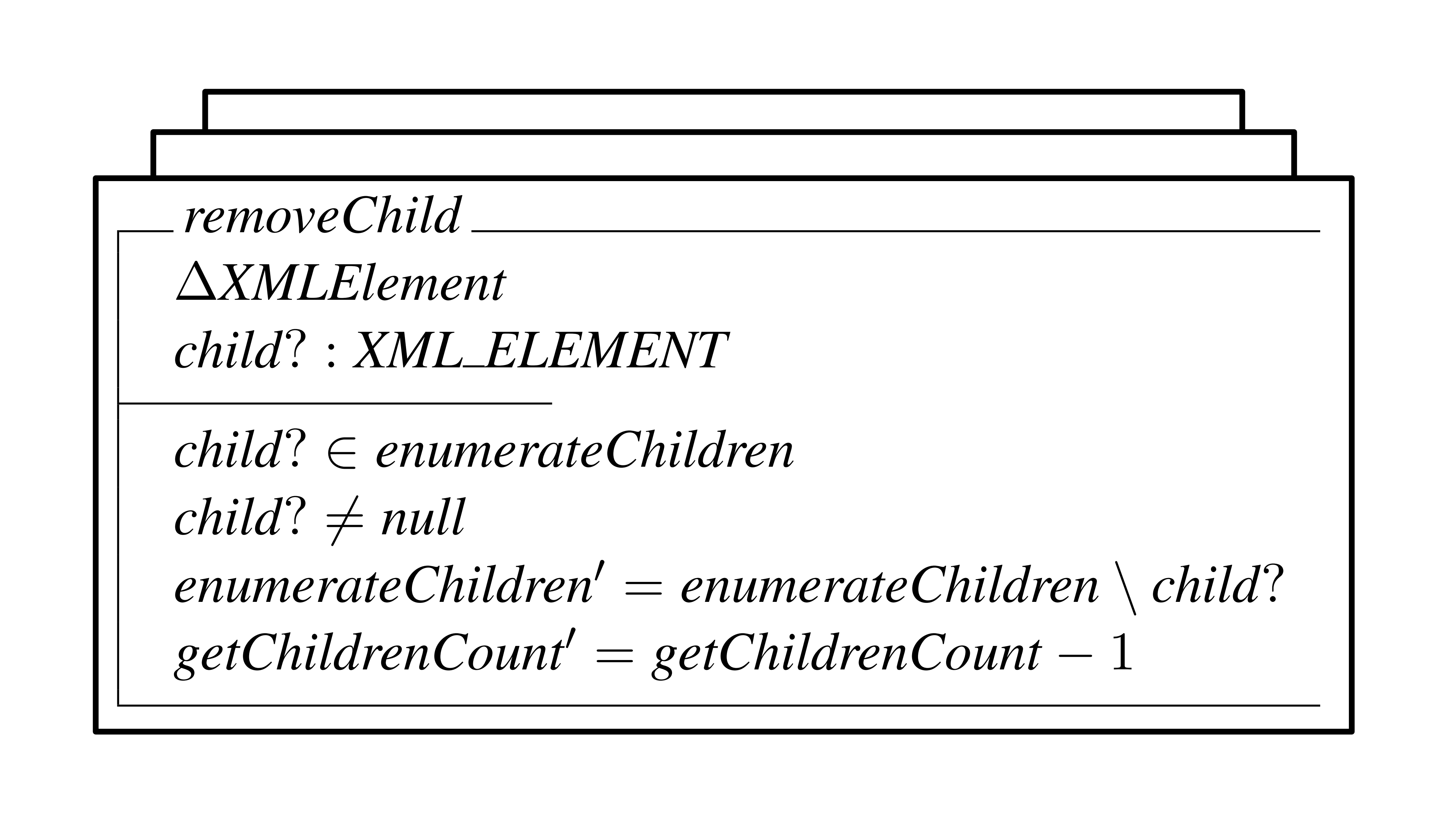


Experimental Program Analysis

Andreas Zeller
Saarland University

Specifications



removeChild

$\Delta \mathit{XMLElement}$

$\mathit{child}? : \mathit{XML\_ELEMENT}$

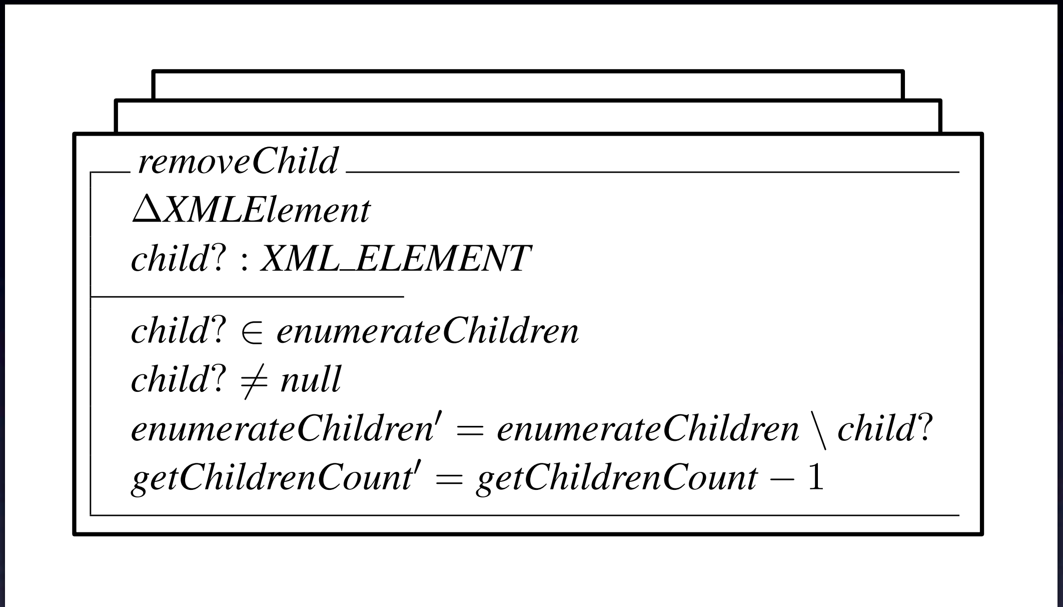
$\mathit{child}? \in \mathit{enumerateChildren}$

$\mathit{child}? \neq \mathit{null}$

$\mathit{enumerateChildren}' = \mathit{enumerateChildren} \setminus \mathit{child}?$

$\mathit{getChildrenCount}' = \mathit{getChildrenCount} - 1$

Specifications



removeChild

$\Delta XMLElement$

$child? : XML\_ELEMENT$

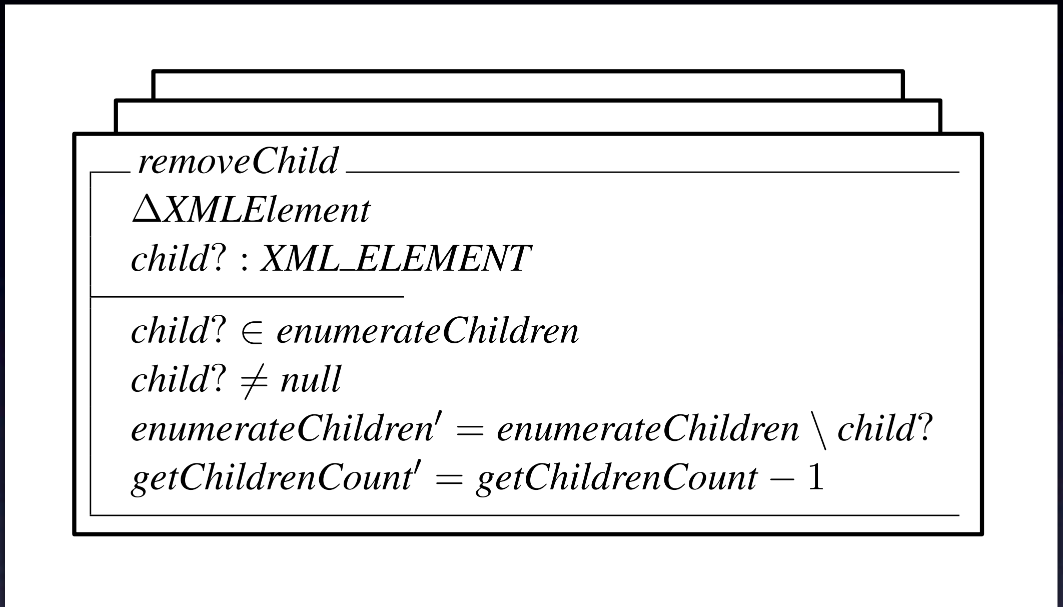
$child? \in enumerateChildren$

$child? \neq null$

$enumerateChildren' = enumerateChildren \setminus child?$

$getChildrenCount' = getChildrenCount - 1$

Specifications



$removeChild$

$\Delta XMLElement$

$child? : XML\_ELEMENT$

$child? \in enumerateChildren$

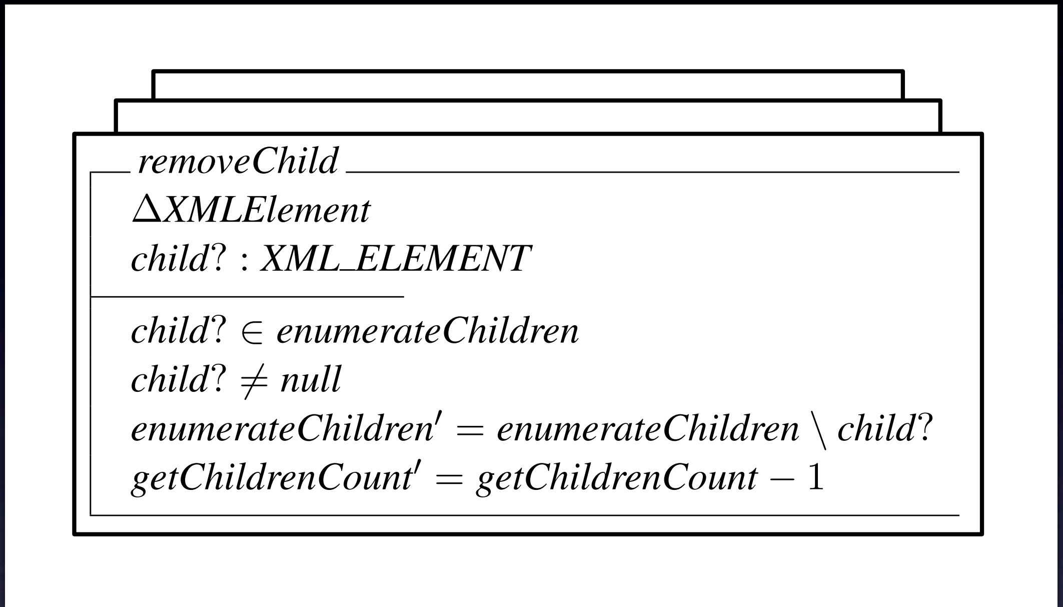
$child? \neq null$

$enumerateChildren' = enumerateChildren \setminus child?$

$getChildrenCount' = getChildrenCount - 1$

fully automated testing

Specifications



removeChild

$\Delta XMLElement$

$child? : XML\_ELEMENT$

$child? \in enumerateChildren$

$child? \neq null$

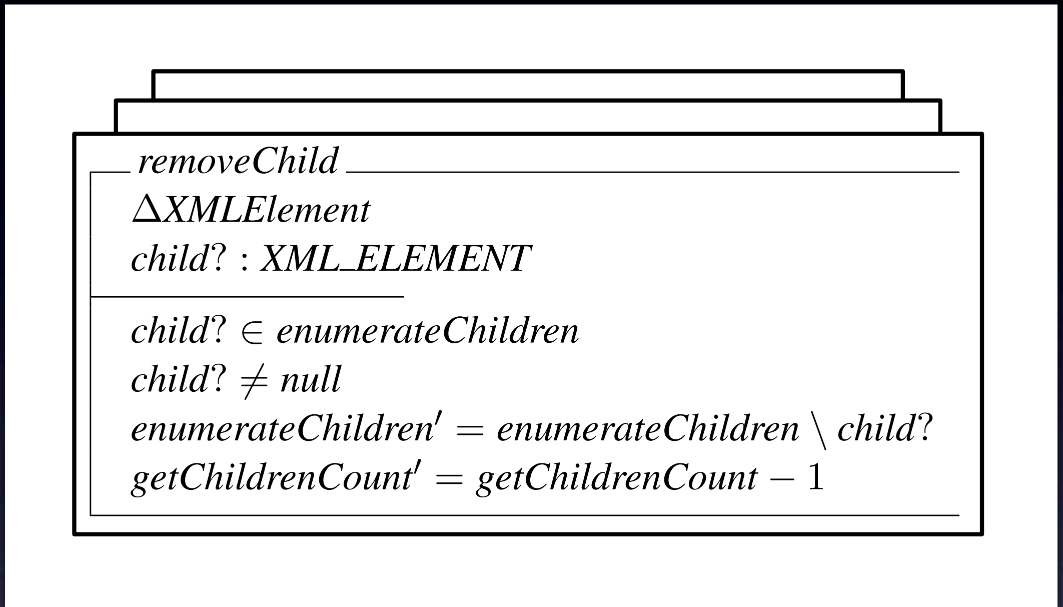
$enumerateChildren' = enumerateChildren \setminus child?$

$getChildrenCount' = getChildrenCount - 1$

fully automated testing

fully automated debugging

Specifications



removeChild

$\Delta XMLElement$

$child? : XML\_ELEMENT$

$child? \in enumerateChildren$

$child? \neq null$

$enumerateChildren' = enumerateChildren \setminus child?$

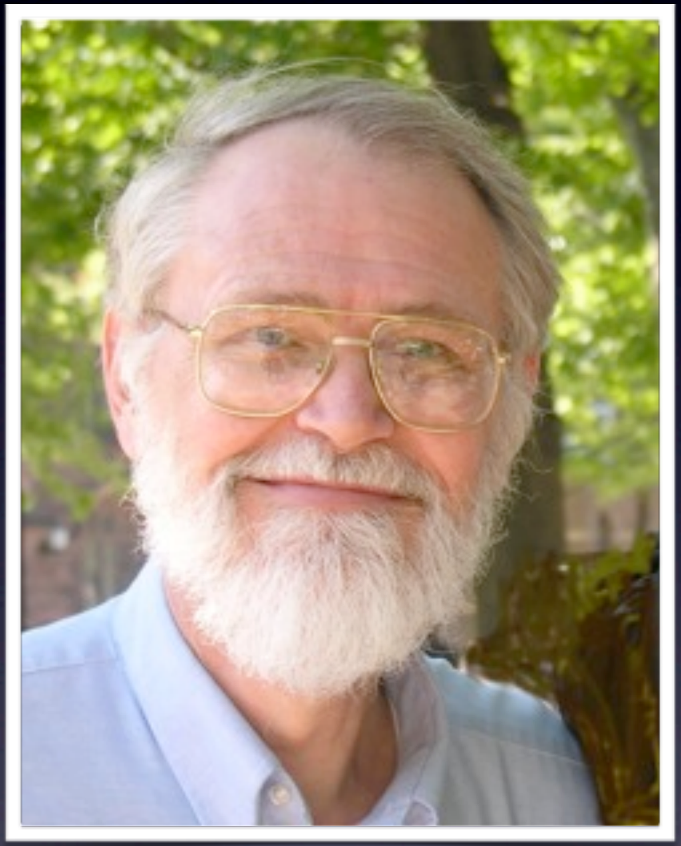
$getChildrenCount' = getChildrenCount - 1$

fully automated testing

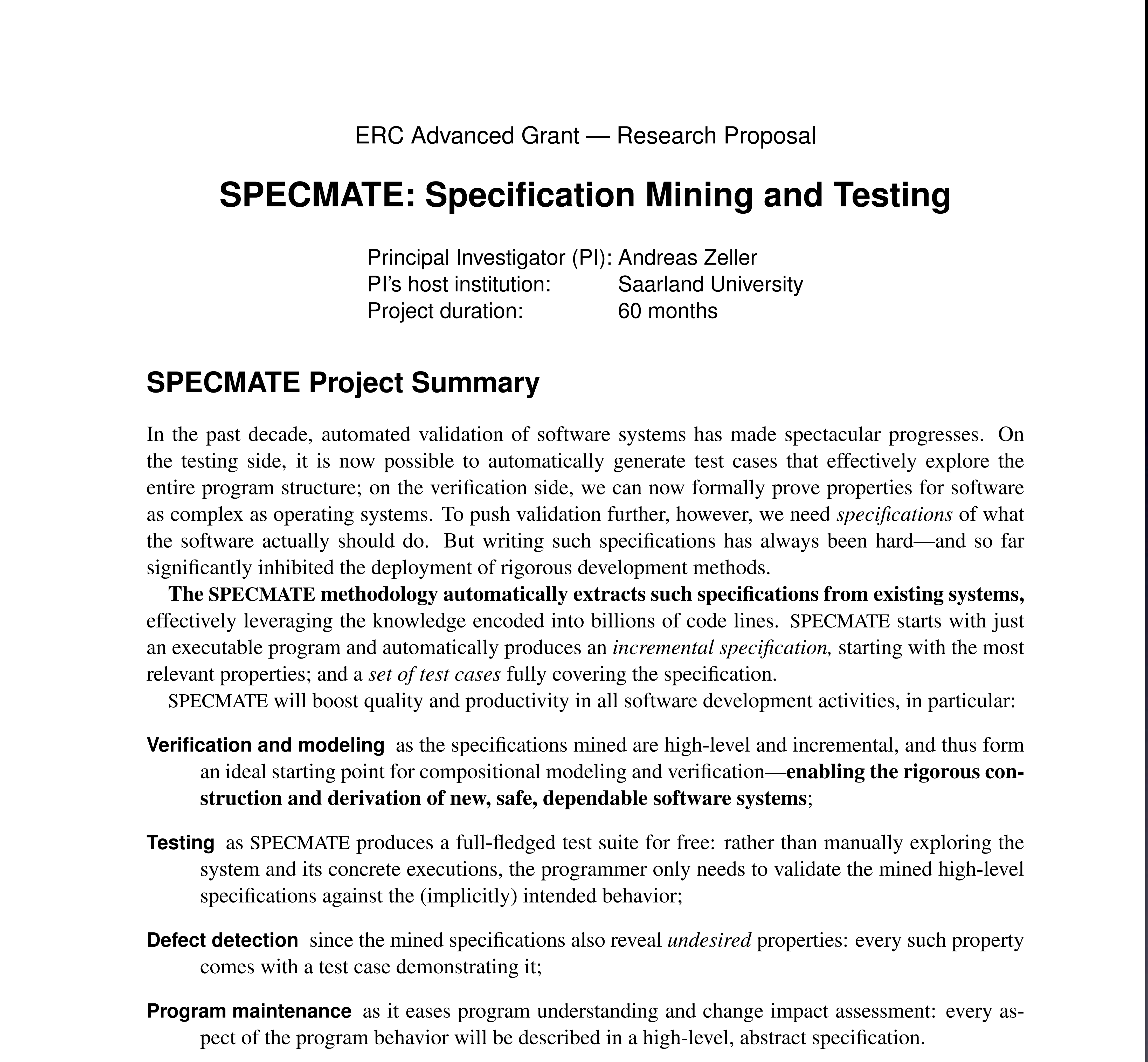
fully automated debugging

widely automated verification

" Without specification,
there are no bugs—
only surprises "



Brian Kernighan



ERC Advanced Grant — Research Proposal

SPECMATE: Specification Mining and Testing

Principal Investigator (PI): Andreas Zeller
PI's host institution: Saarland University
Project duration: 60 months

SPECMATE Project Summary

In the past decade, automated validation of software systems has made spectacular progresses. On the testing side, it is now possible to automatically generate test cases that effectively explore the entire program structure; on the verification side, we can now formally prove properties for software as complex as operating systems. To push validation further, however, we need *specifications* of what the software actually should do. But writing such specifications has always been hard—and so far significantly inhibited the deployment of rigorous development methods.

The SPECMATE methodology automatically extracts such specifications from existing systems, effectively leveraging the knowledge encoded into billions of code lines. SPECMATE starts with just an executable program and automatically produces an *incremental specification,* starting with the most relevant properties; and a *set of test cases* fully covering the specification.

SPECMATE will boost quality and productivity in all software development activities, in particular:

Verification and modeling as the specifications mined are high-level and incremental, and thus form an ideal starting point for compositional modeling and verification—**enabling the rigorous construction and derivation of new, safe, dependable software systems**;

Testing as SPECMATE produces a full-fledged test suite for free: rather than manually exploring the system and its concrete executions, the programmer only needs to validate the mined high-level specifications against the (implicitly) intended behavior;

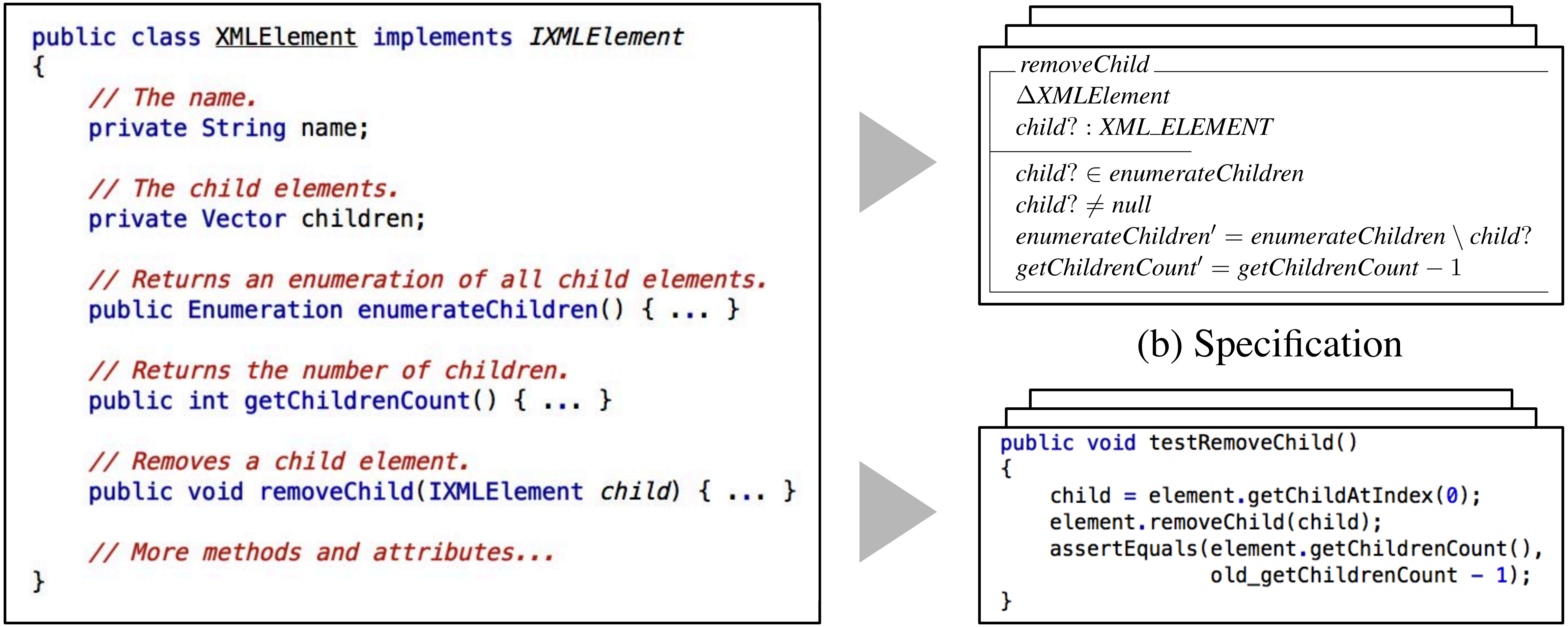
Defect detection since the mined specifications also reveal *undesired* properties: every such property comes with a test case demonstrating it;

Program maintenance as it eases program understanding and change impact assessment: every aspect of the program behavior will be described in a high-level, abstract specification.

ERC Advanced Grant — Research Proposal

SPECMATE: Specification Mining and Testing

Principal Investigator (PI): Andreas Zeller
PI's host institution: Saarland University



```
public class XMLElement implements IXMLElement
{
    // The name.
    private String name;

    // The child elements.
    private Vector children;

    // Returns an enumeration of all child elements.
    public Enumeration enumerateChildren() { ... }

    // Returns the number of children.
    public int getChildrenCount() { ... }

    // Removes a child element.
    public void removeChild(IXMLElement child) { ... }

    // More methods and attributes...
}
```

removeChild

$\Delta XMLElement$

$child? : XML\_ELEMENT$

$child? \in enumerateChildren$

$child? \neq null$

$enumerateChildren' = enumerateChildren \setminus child?$

$getChildrenCount' = getChildrenCount - 1$

(b) Specification

```
public void testRemoveChild()
{
    child = element.getChildAtIndex(0);
    element.removeChild(child);
    assertEquals(element.getChildrenCount(),
                 old_getChildrenCount - 1);
}
```

(a) Executable Program

(c) Test

Defect detection since the mined specifications also reveal *undesired* properties: every such property comes with a test case demonstrating it;

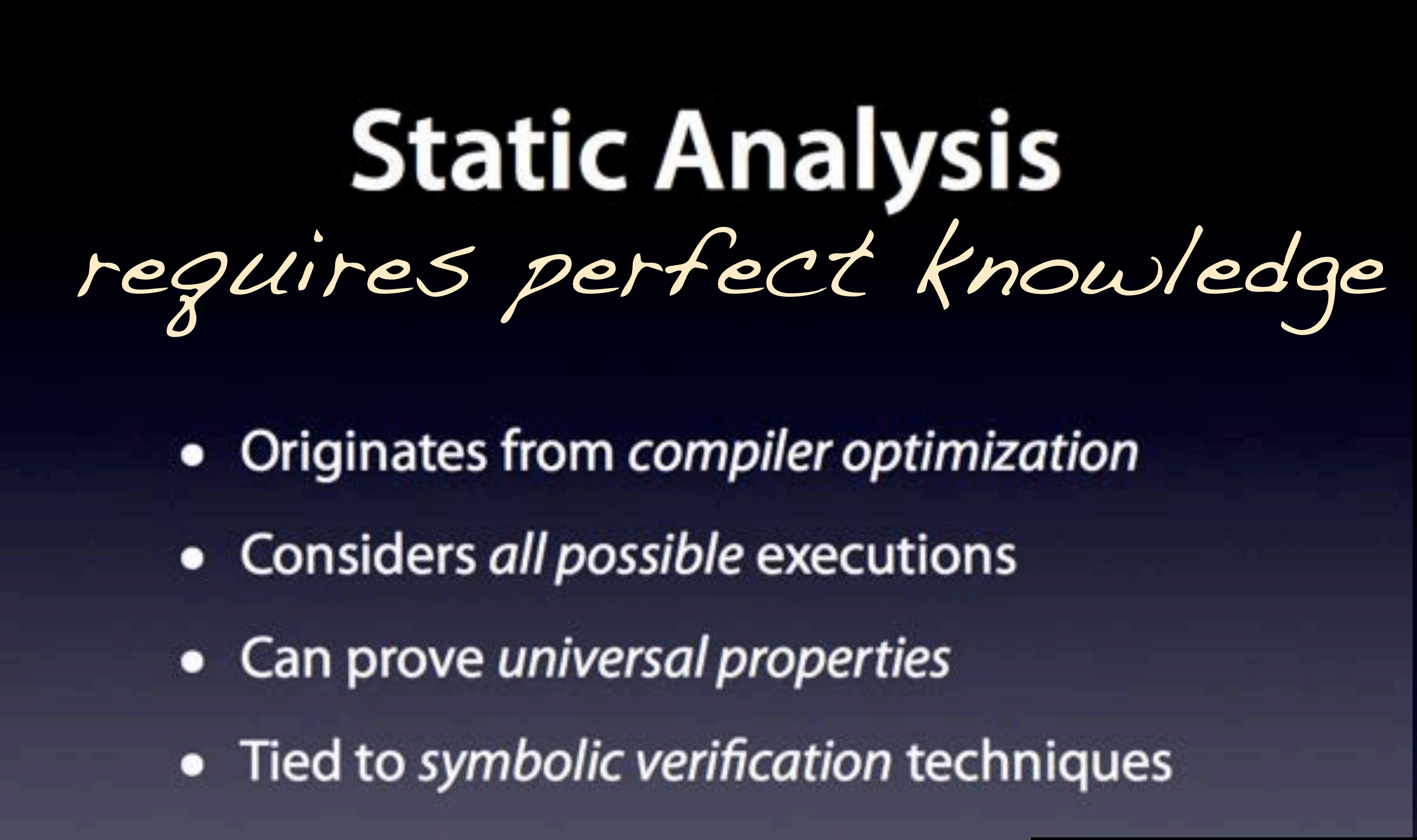
Program maintenance as it eases program understanding and change impact assessment: every aspect of the program behavior will be described in a high-level, abstract specification.

Static Analysis

- Originates from *compiler optimization*
- Considers *all possible* executions
- Can prove *universal properties*
- Tied to *symbolic verification* techniques

Dynamic Analysis

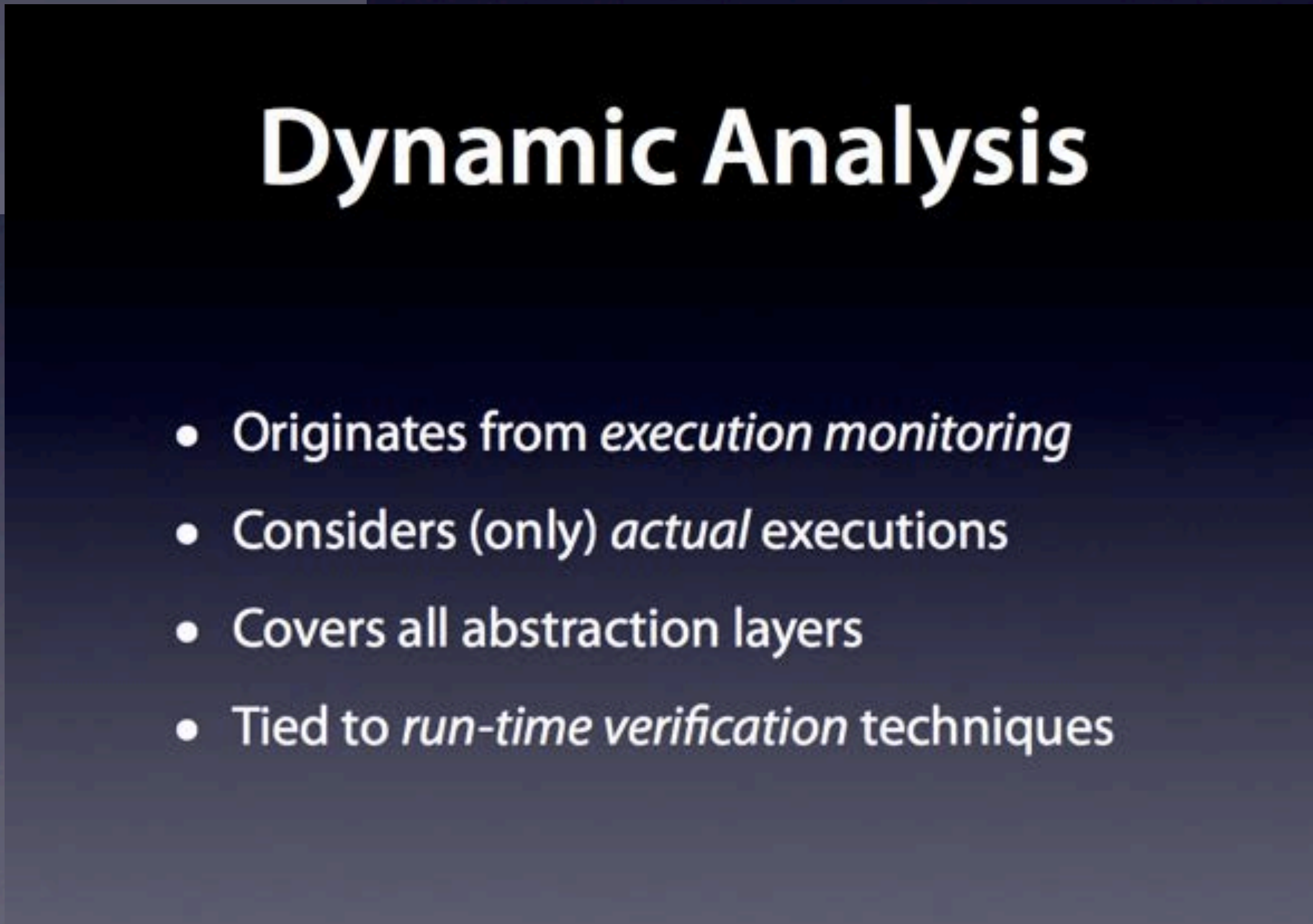
- Originates from *execution monitoring*
- Considers (only) *actual* executions
- Covers all abstraction layers
- Tied to *run-time verification* techniques



Static Analysis

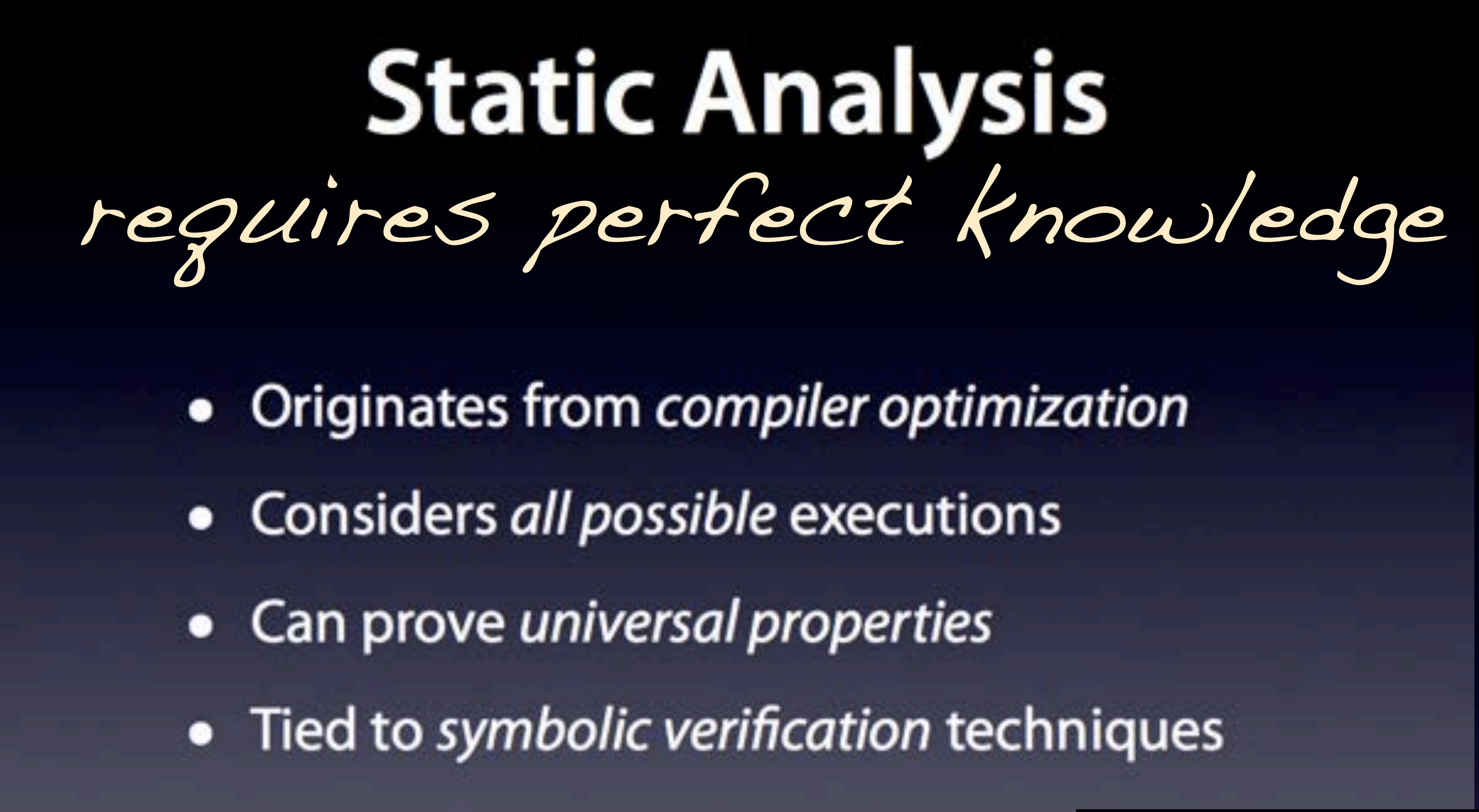
requires perfect knowledge

- Originates from *compiler optimization*
- Considers *all possible* executions
- Can prove *universal properties*
- Tied to *symbolic verification* techniques



Dynamic Analysis

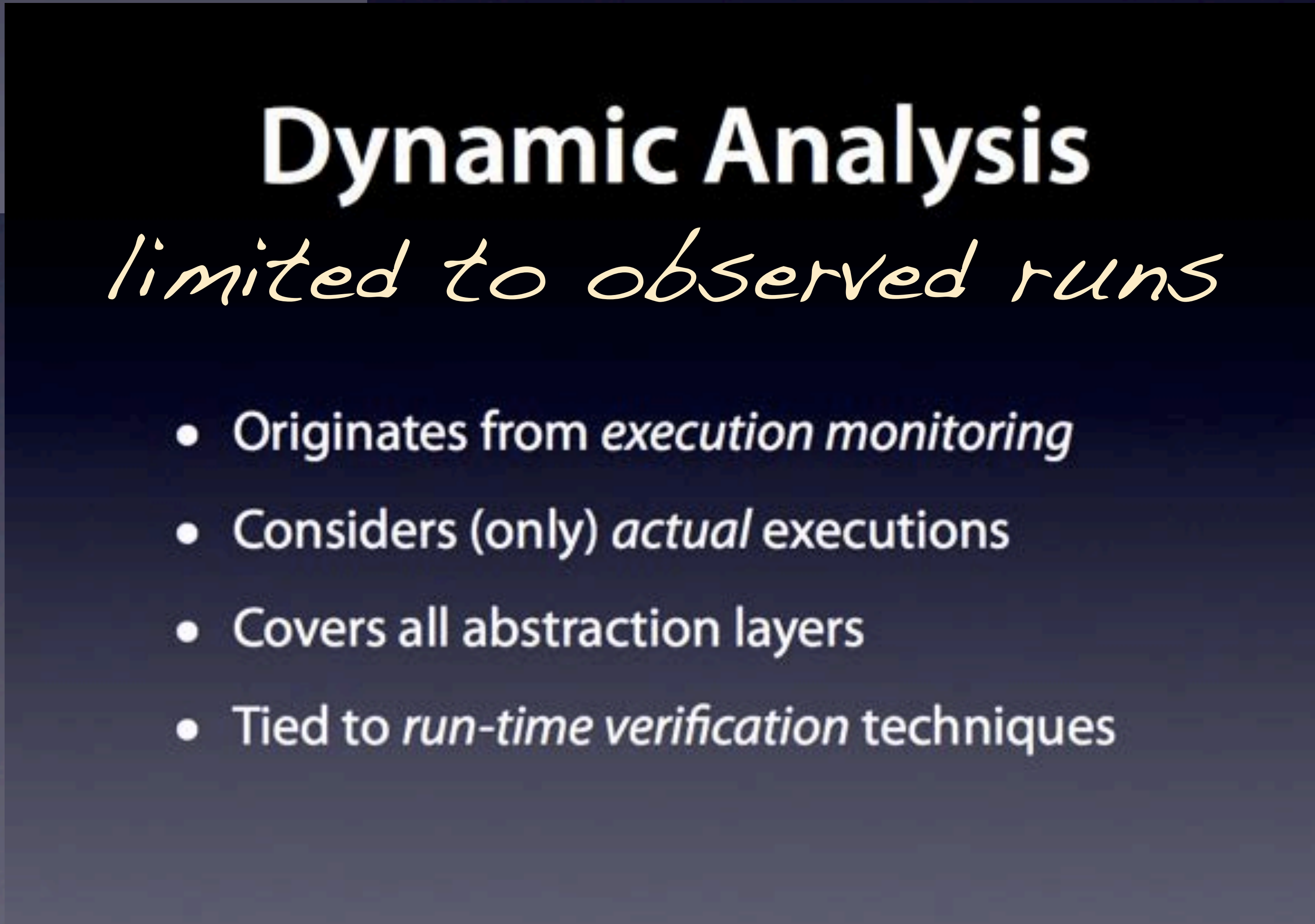
- Originates from *execution monitoring*
- Considers (only) *actual* executions
- Covers all abstraction layers
- Tied to *run-time verification* techniques



Static Analysis

requires perfect knowledge

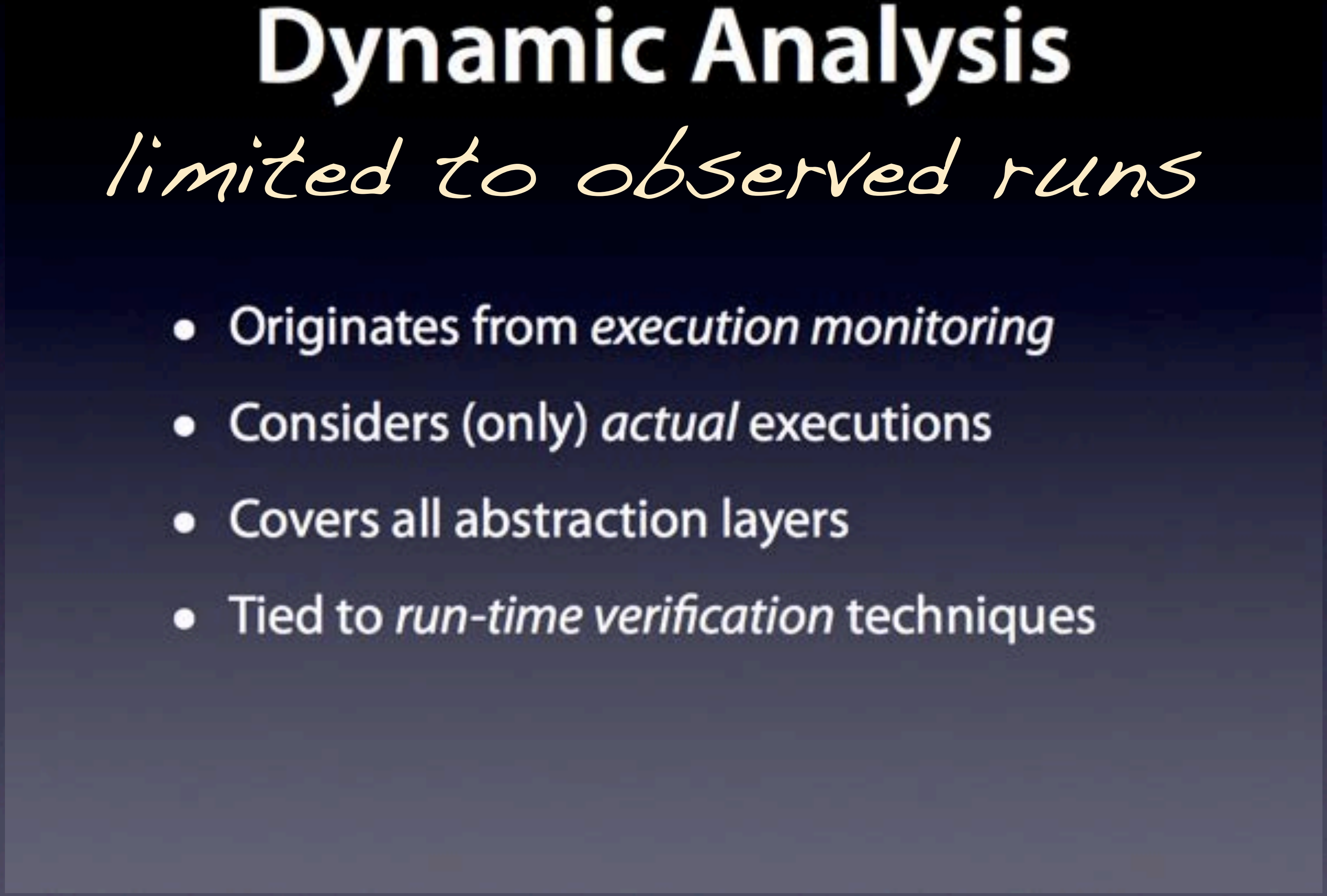
- Originates from *compiler optimization*
- Considers *all possible* executions
- Can prove *universal properties*
- Tied to *symbolic verification* techniques



Dynamic Analysis

limited to observed runs

- Originates from *execution monitoring*
- Considers (only) *actual* executions
- Covers all abstraction layers
- Tied to *run-time verification* techniques



Dynamic Analysis

limited to observed runs

- Originates from *execution monitoring*
- Considers (only) *actual* executions
- Covers all abstraction layers
- Tied to *run-time verification* techniques

limited to observed runs

need more runs

Generate test cases
to systematically
explore behavior

executions

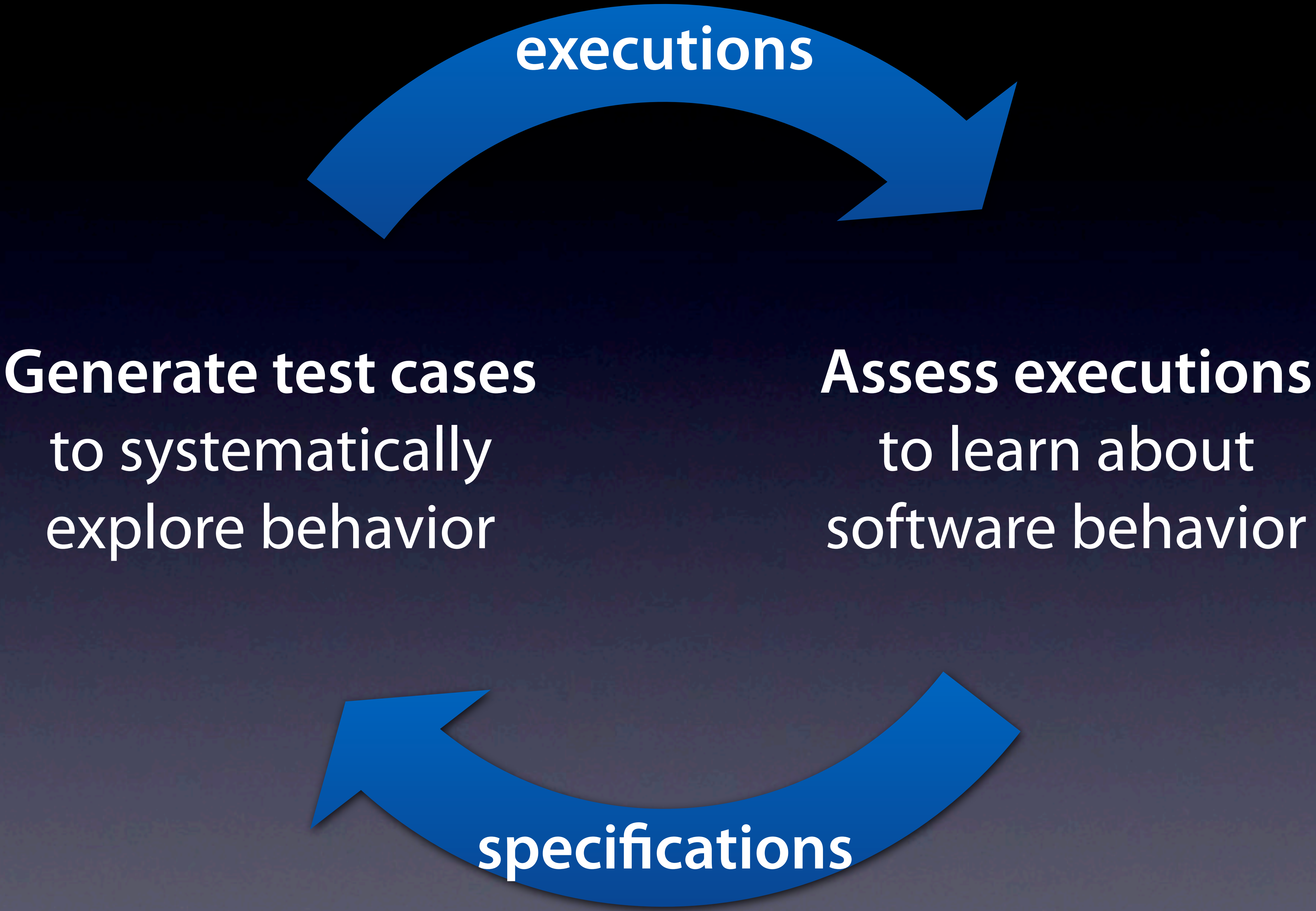
Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior

executions

Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior



executions

Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior

specifications

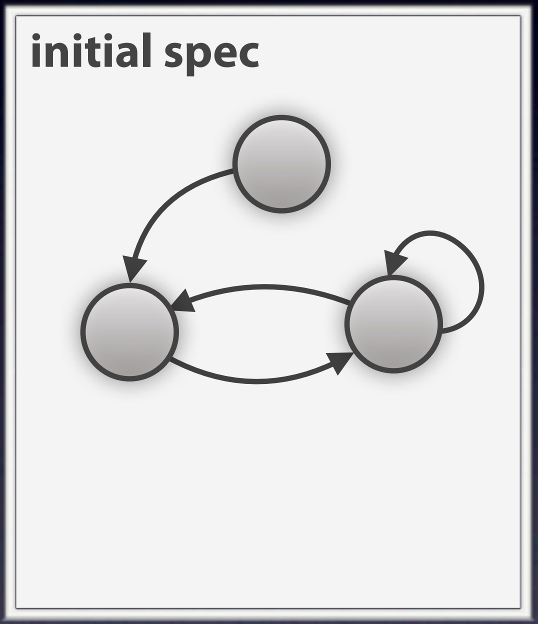
Enriching specifications

Enriching specifications

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

Enriching specifications

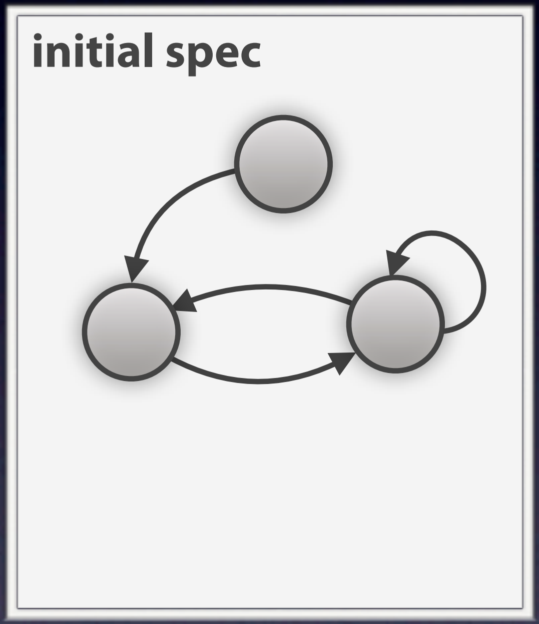
```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

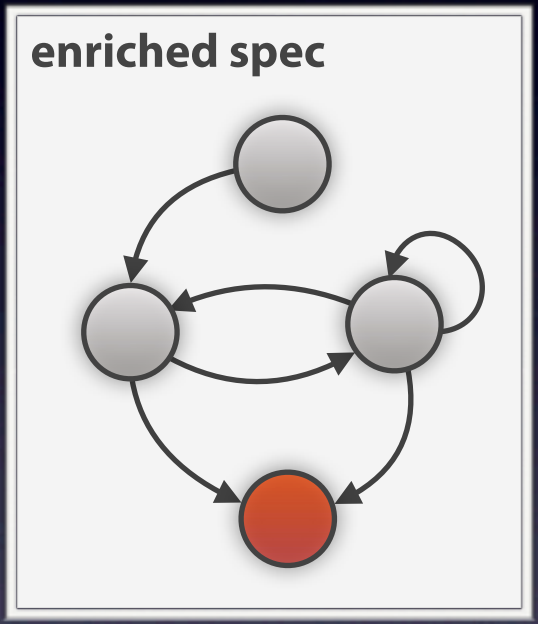


Execute and extract initial spec

Enriching specifications

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```





Execute and extract initial spec

Generate test mutants and enrich specs

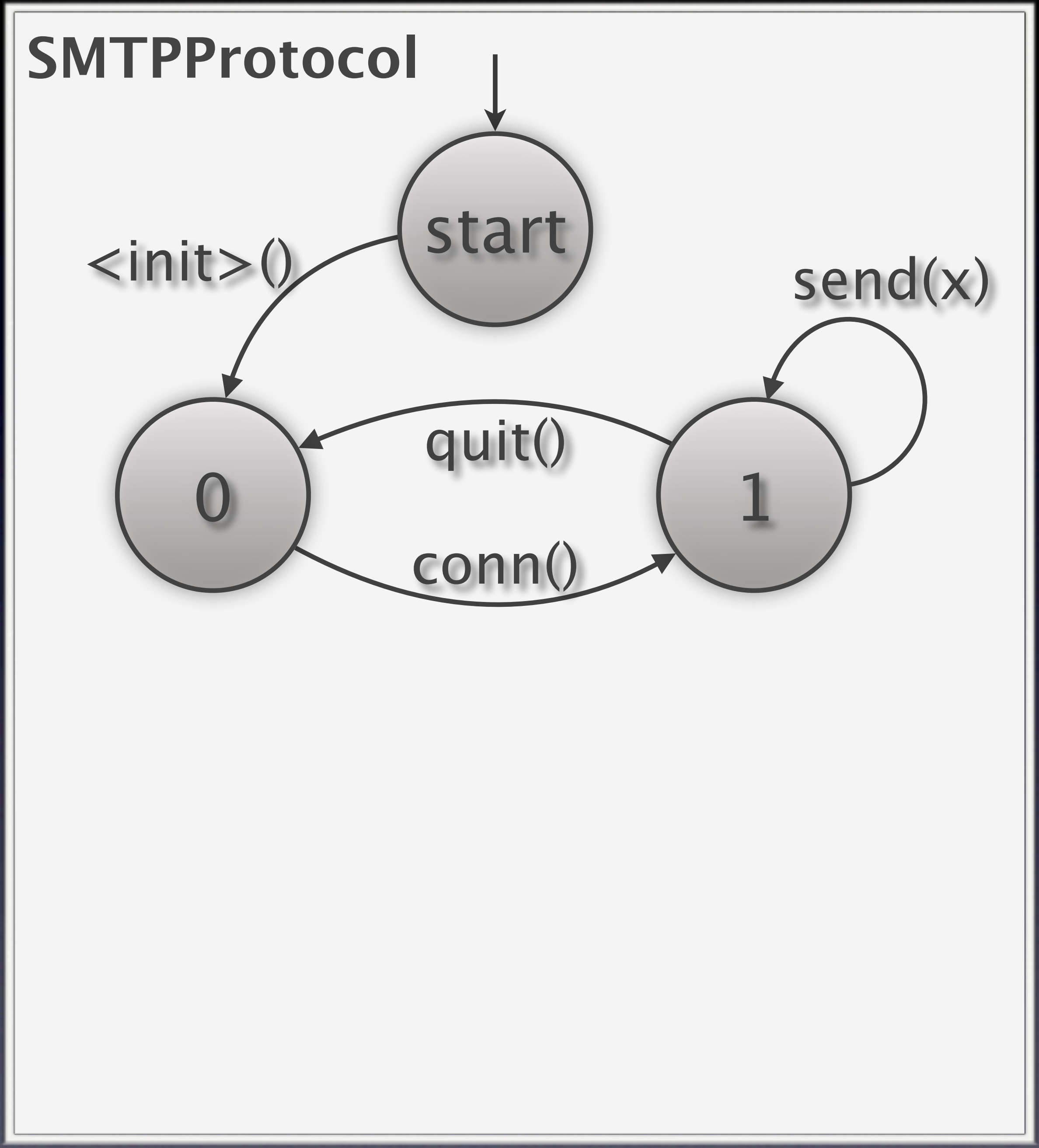
Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

Dallmeier et al: “Generating Test Cases for Specification Mining”, ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

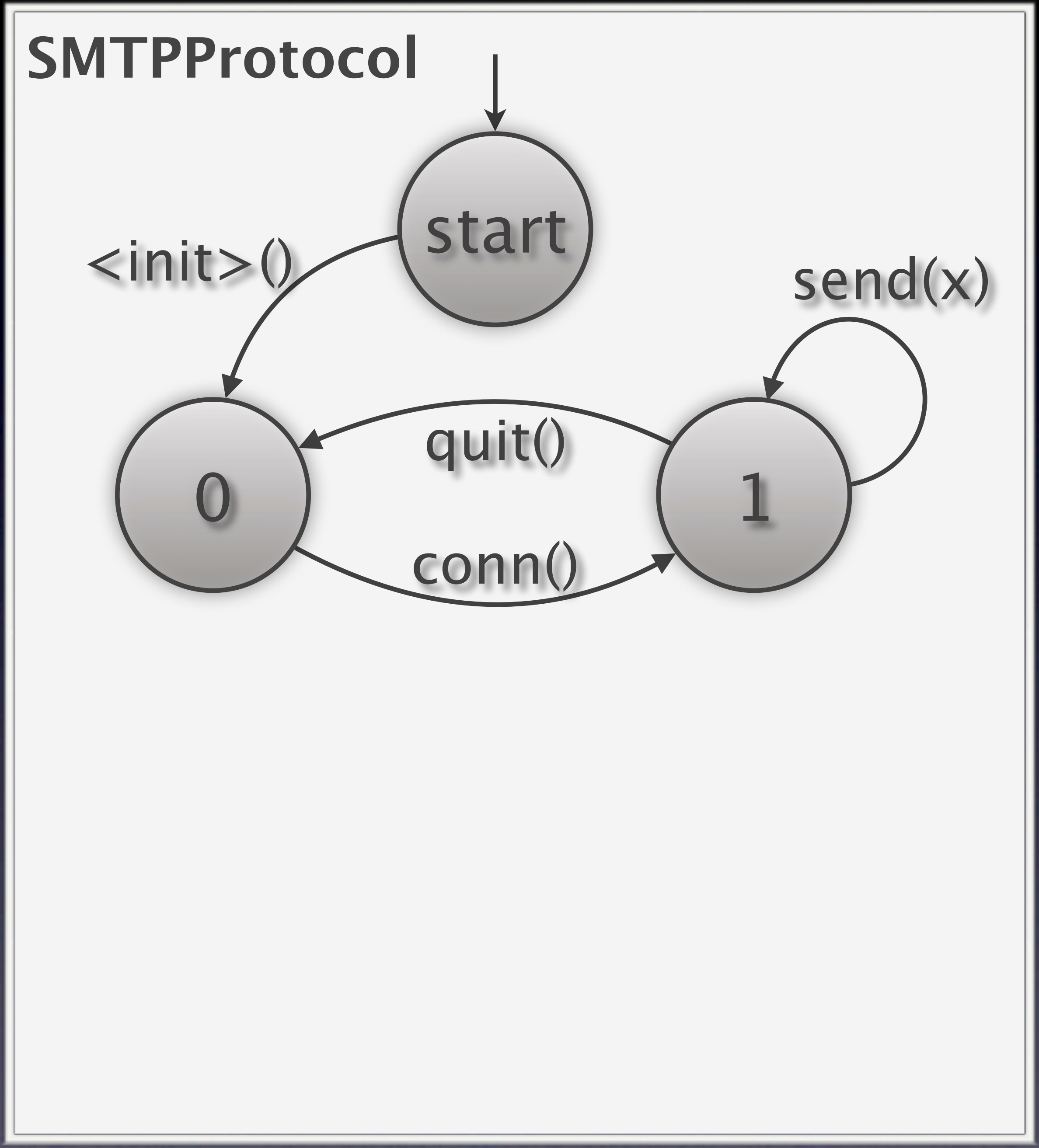
Dallmeier et al: “Generating Test Cases for Specification Mining”, ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```



Dallmeier et al: “Generating Test Cases for Specification Mining”, ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

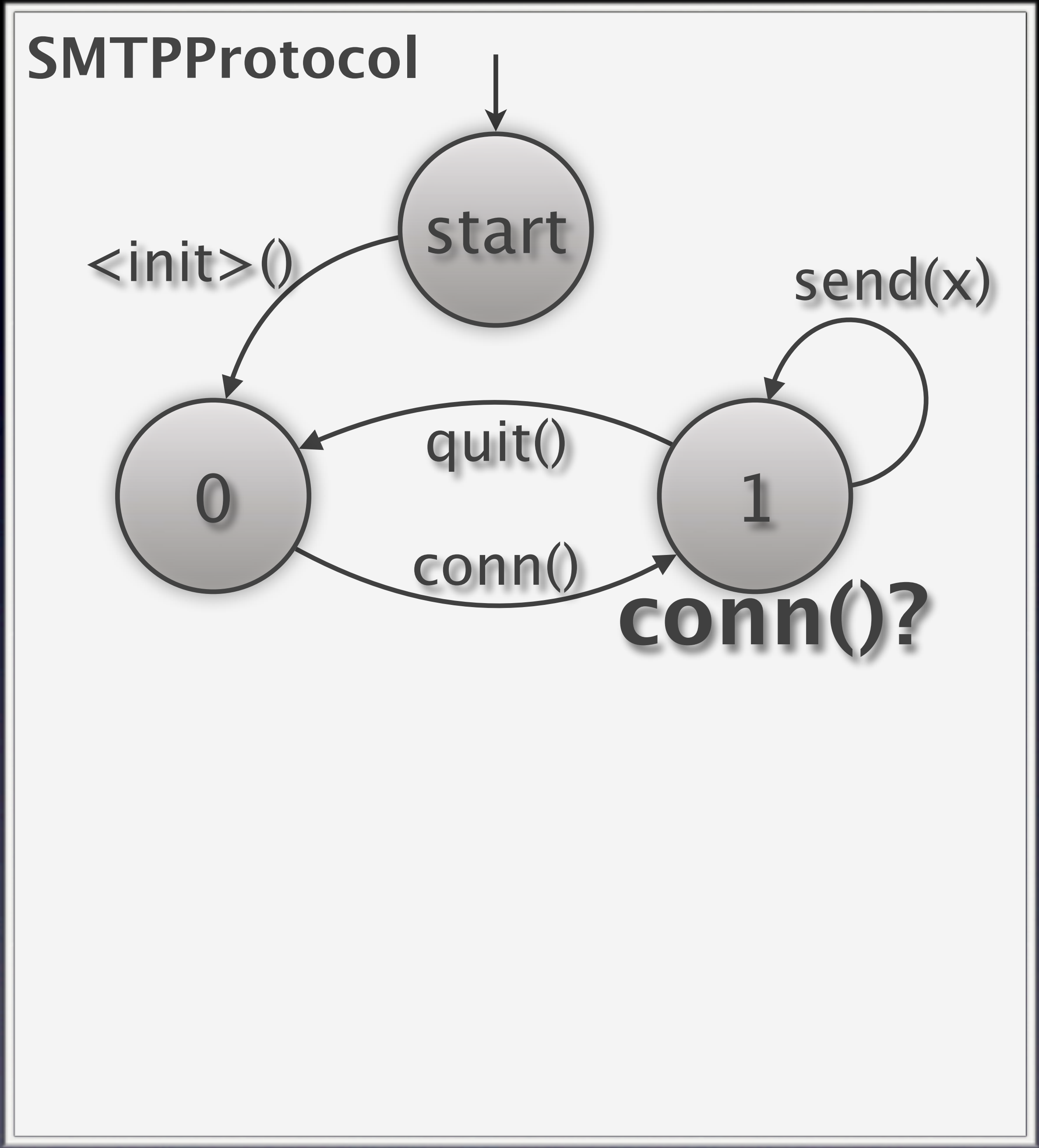


Uncovered

0: send(x)
   quit()
1: conn()

Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```



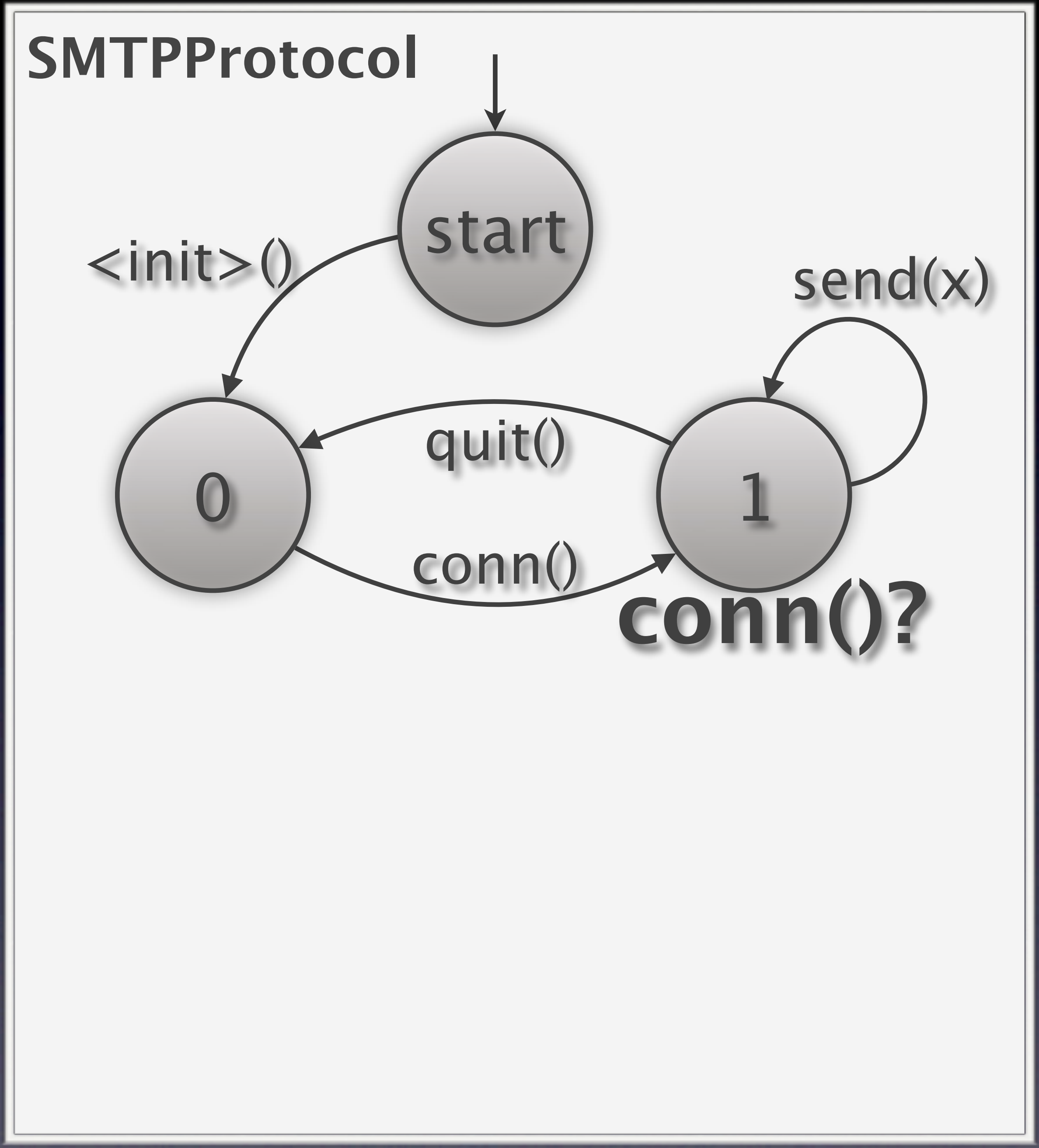
Uncovered

0: send(x)
quit()
1: conn()

Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

```
void TestMutant1() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.conn();
    p.quit();
}
```

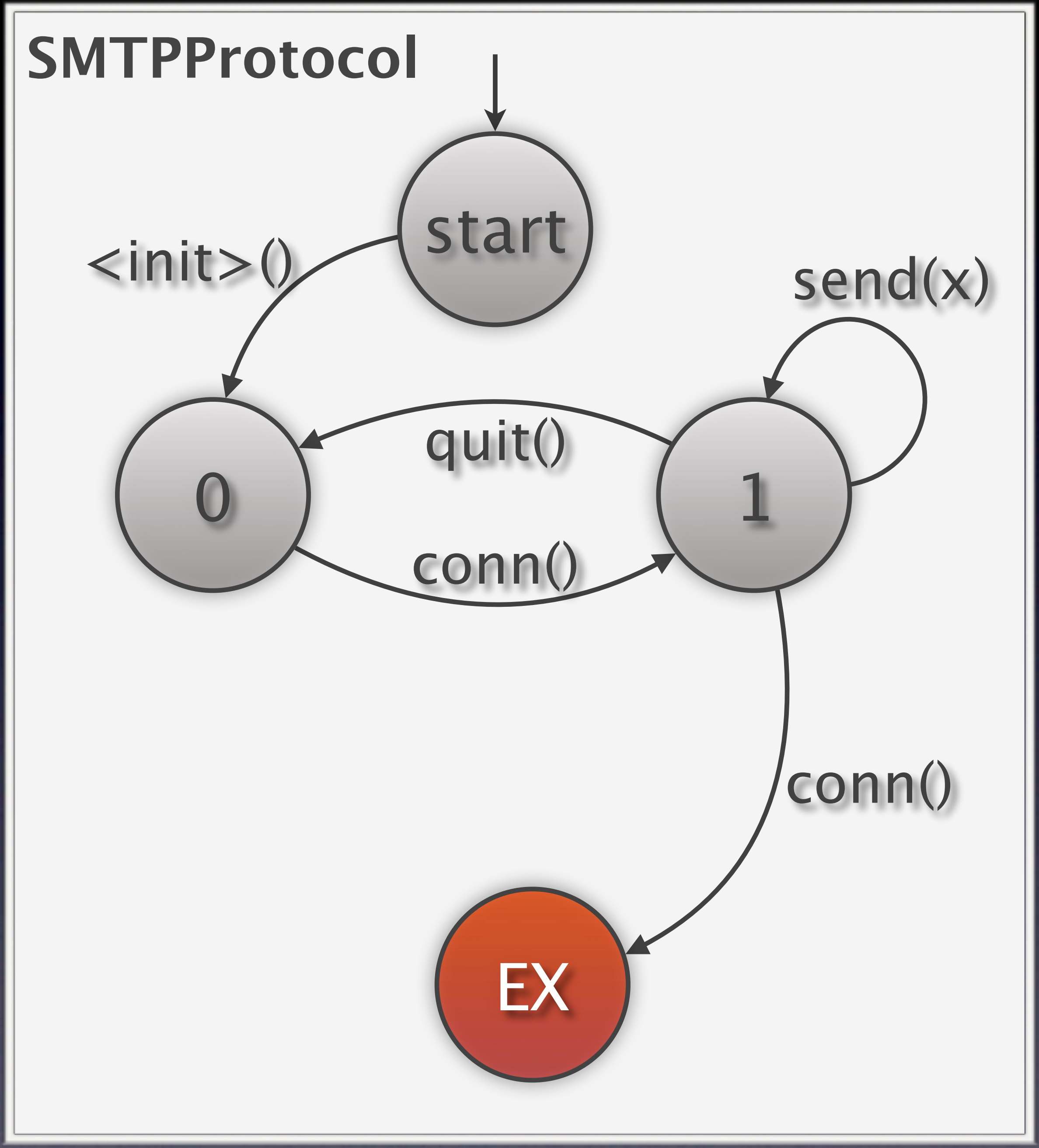


conn()?

Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

```
void TestMutant1() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.conn();
    p.quit();
}
```



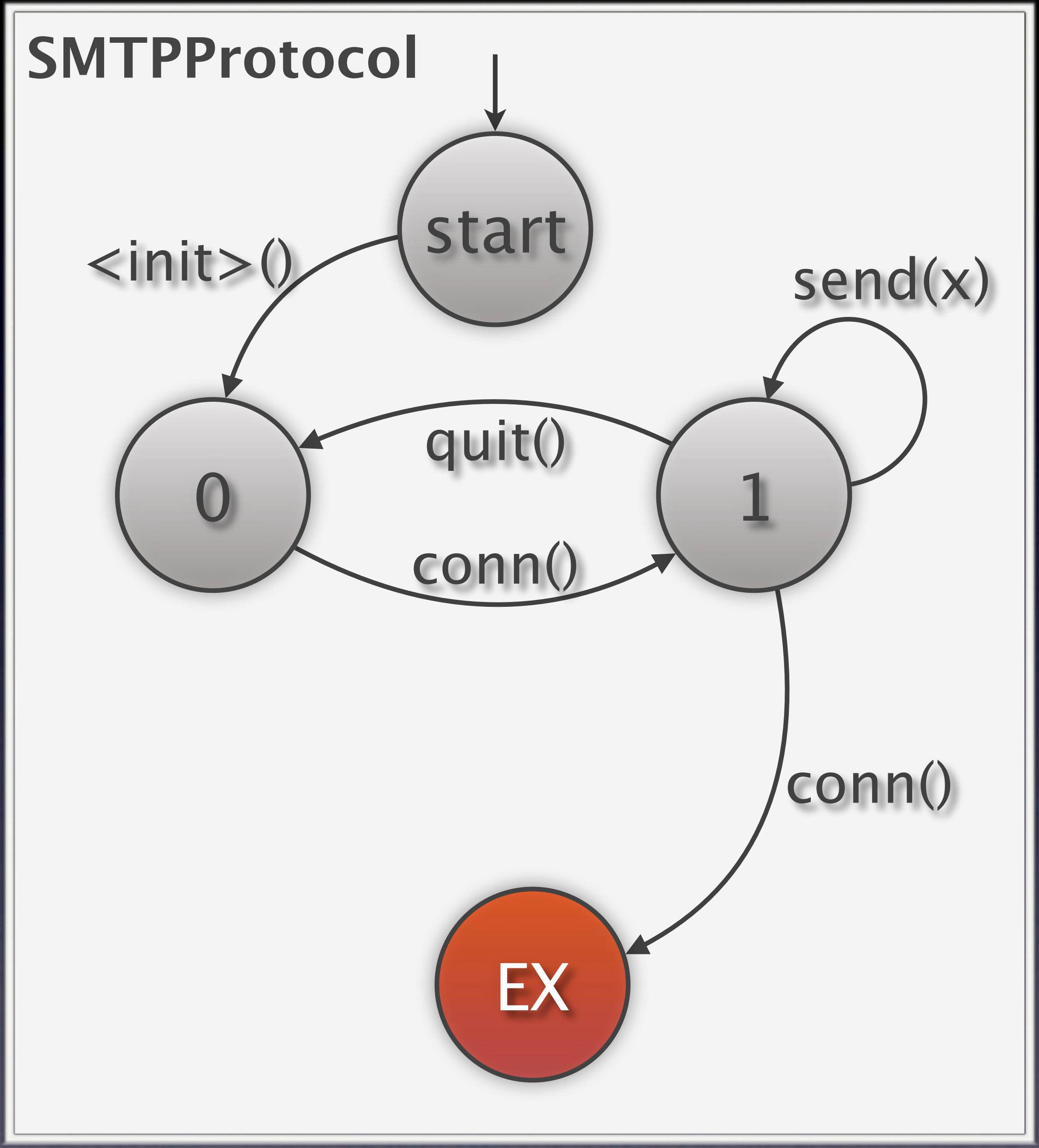
Dallmeier et al: “Generating Test Cases for Specification Mining”, ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

Uncovered

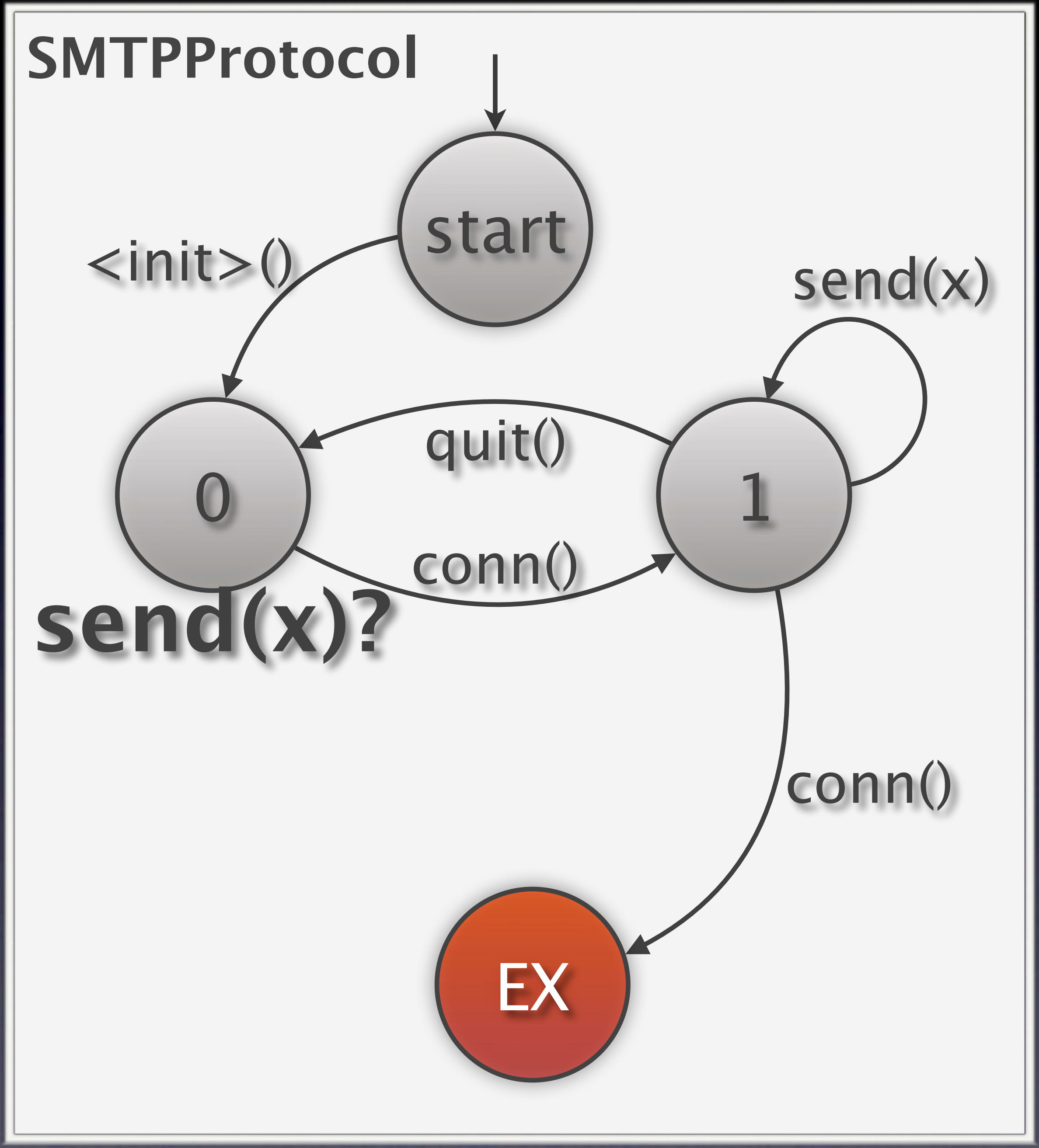
0: send(x)
quit()

~~1: conn()~~



Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```



send(x)?

Uncovered

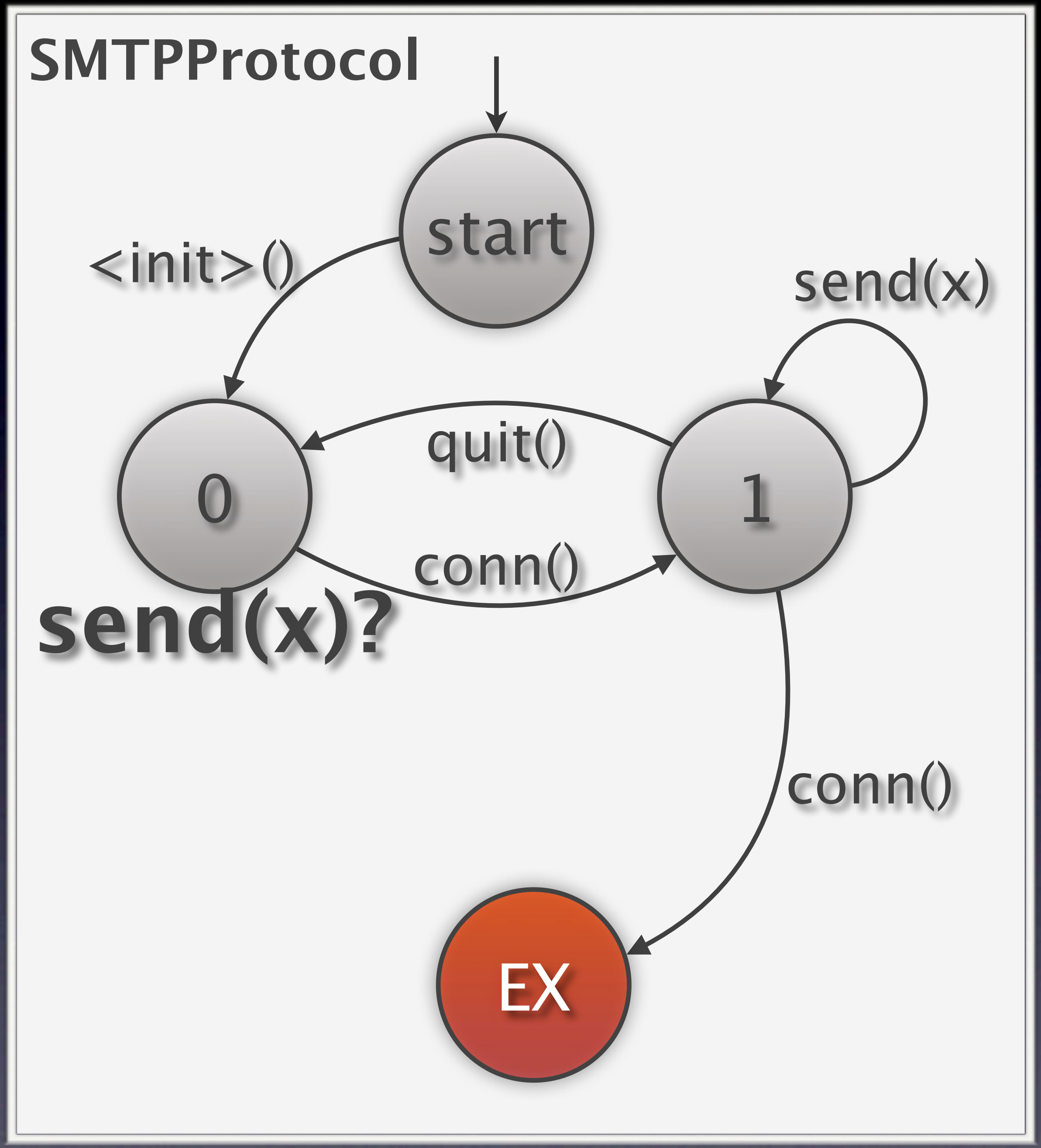
0: send(x)
quit()

~~1: conn()~~

Dallmeier et al: “Generating Test Cases for Specification Mining”, ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

```
void TestMutant2() {
    Protocol p = new ...
    //p.conn();
    p.send(x);
    p.quit();
}
```

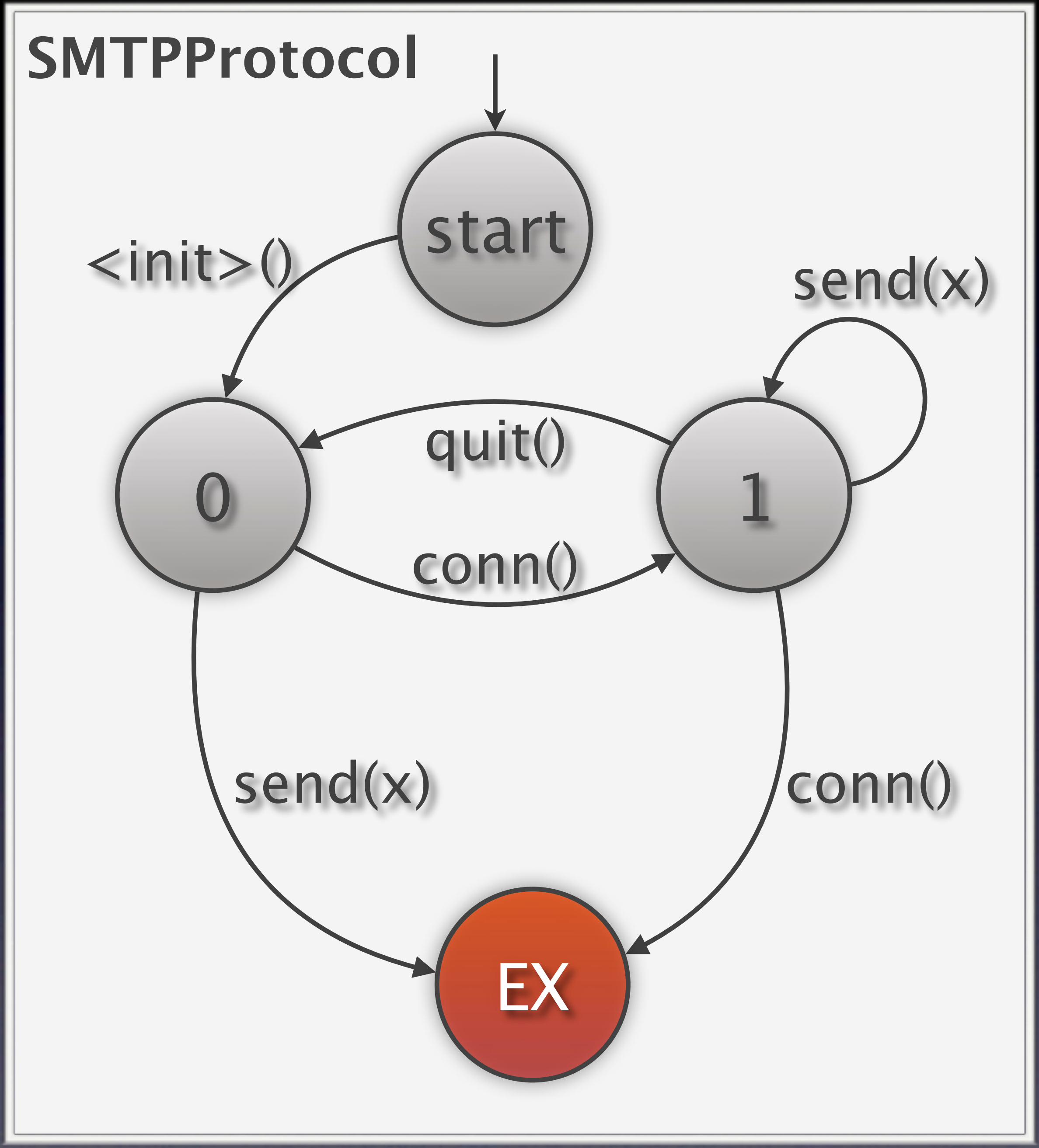


send(x)?

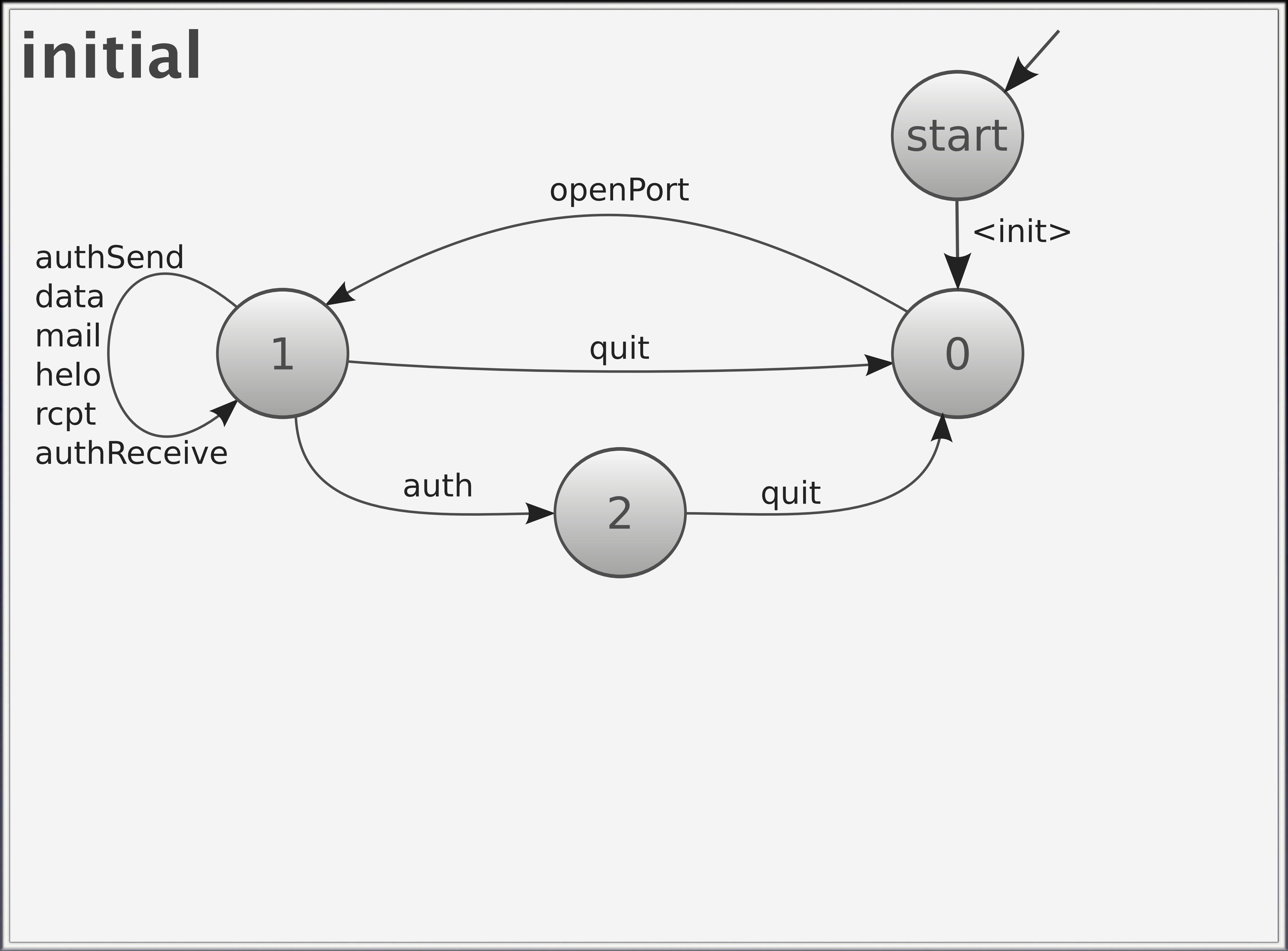
Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

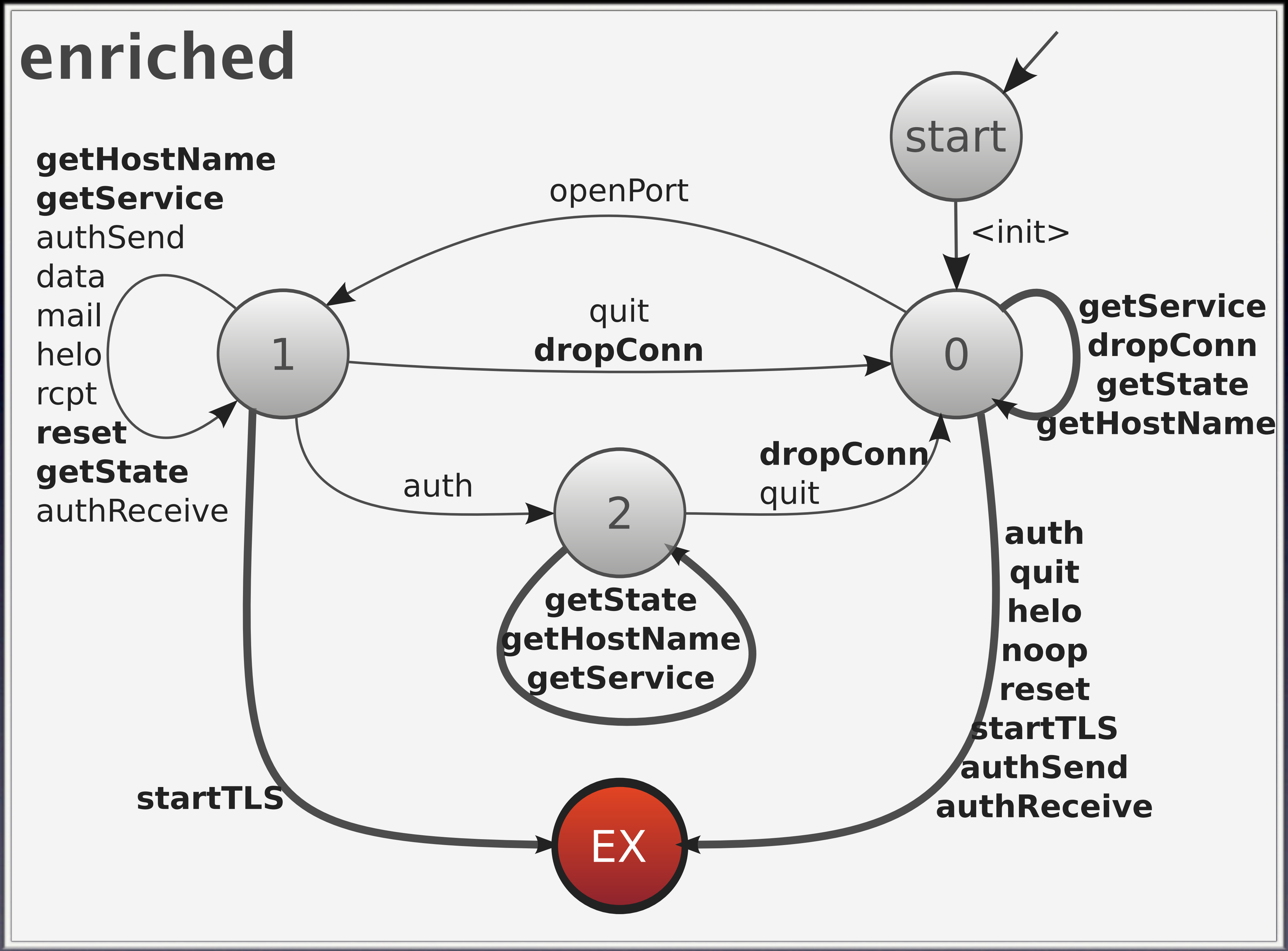
```
void TestMutant2() {
    Protocol p = new ...
    //p.conn();
    p.send(x);
    p.quit();
}
```



Dallmeier et al: “Generating Test Cases for Specification Mining”, ISSTA 2010



Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010



Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

A new kind of Analysis

A new kind of Analysis

- Static analysis

A new kind of Analysis

- Static analysis - 0 runs

A new kind of Analysis

- Static analysis - 0 runs
- Dynamic analysis

A new kind of Analysis

- Static analysis – 0 runs
- Dynamic analysis – n given runs

A new kind of Analysis

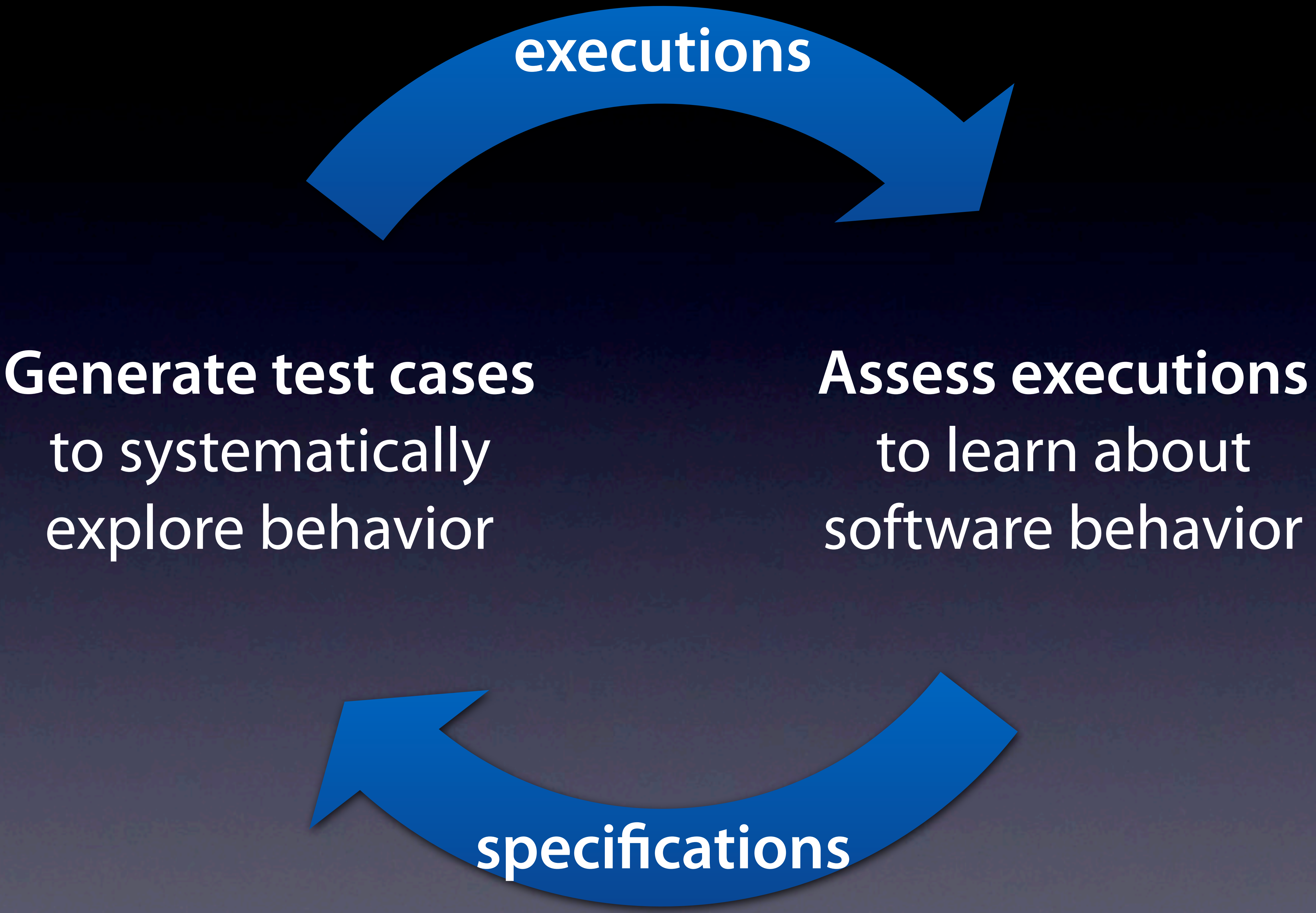
- Static analysis - 0 runs
- Dynamic analysis - n <u>given</u> runs
- Experimental analysis

A new kind of Analysis

- Static analysis – 0 runs
- Dynamic analysis – n given runs
- Experimental analysis – n generated runs

A new kind of Analysis

- Static analysis – 0 runs
- Dynamic analysis – n given runs
- **Experimental analysis** – n generated runs

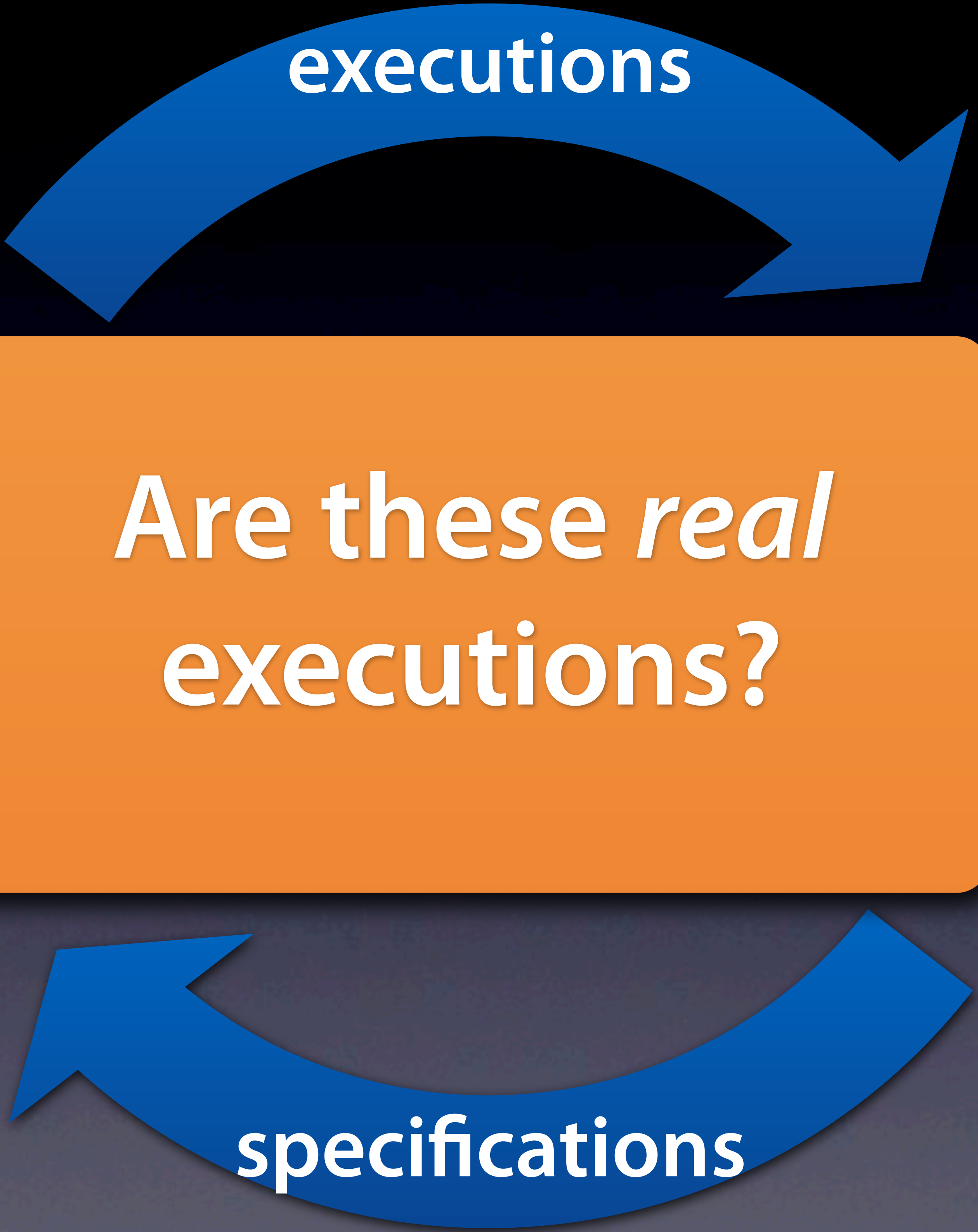


executions

Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior

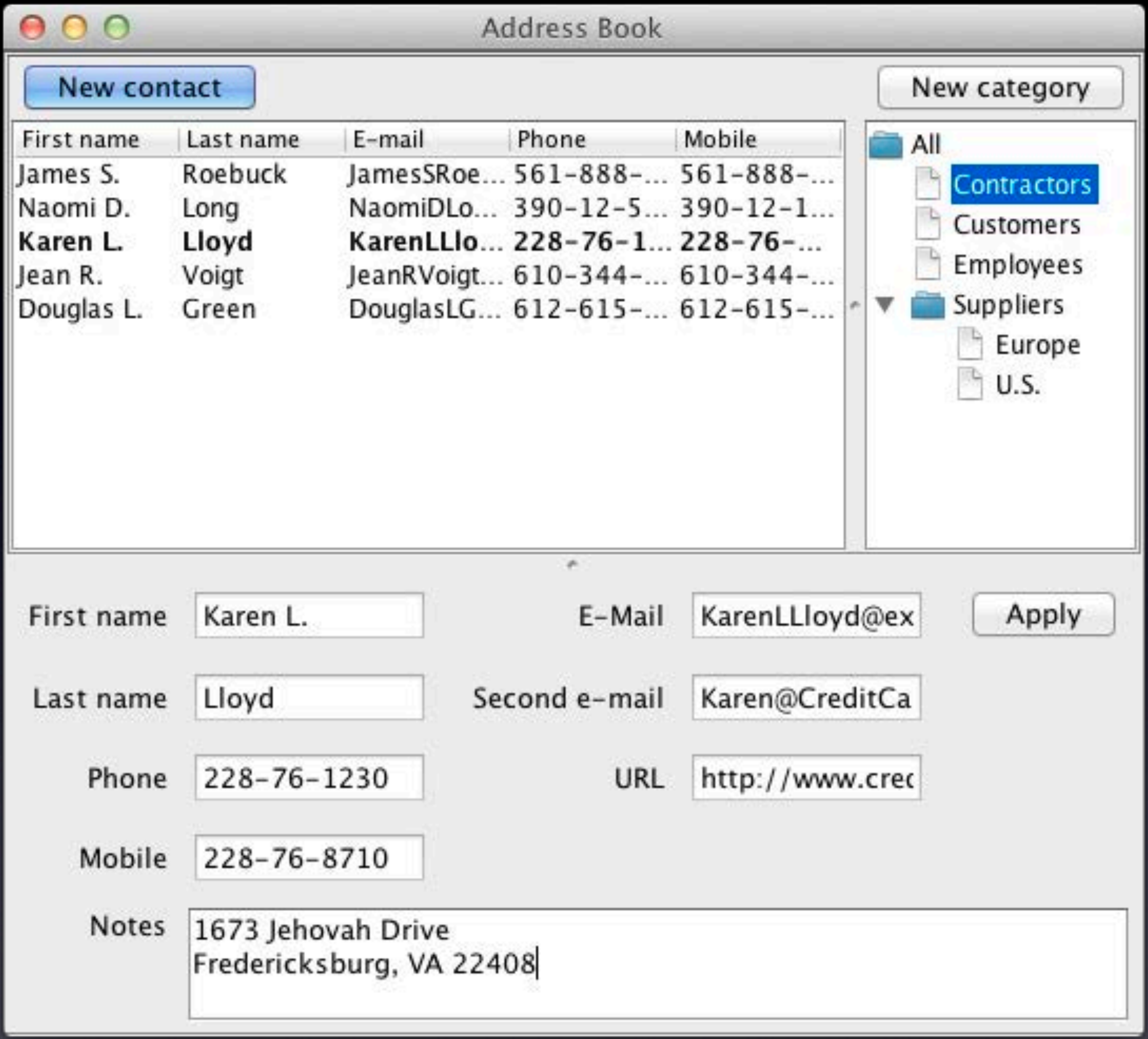
specifications



Generate
to syste
explore

Are these *real* executions?

executions
rn about
e behavior



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

First name: Karen L.

Last name: Lloyd

Phone: 228-76-1230

Mobile: 228-76-8710

E-Mail: KarenLLloyd@ex

Second e-mail: Karen@CreditCa

URL: http://www.crec

Notes:
1673 Jehovah Drive
Fredericksburg, VA 22408

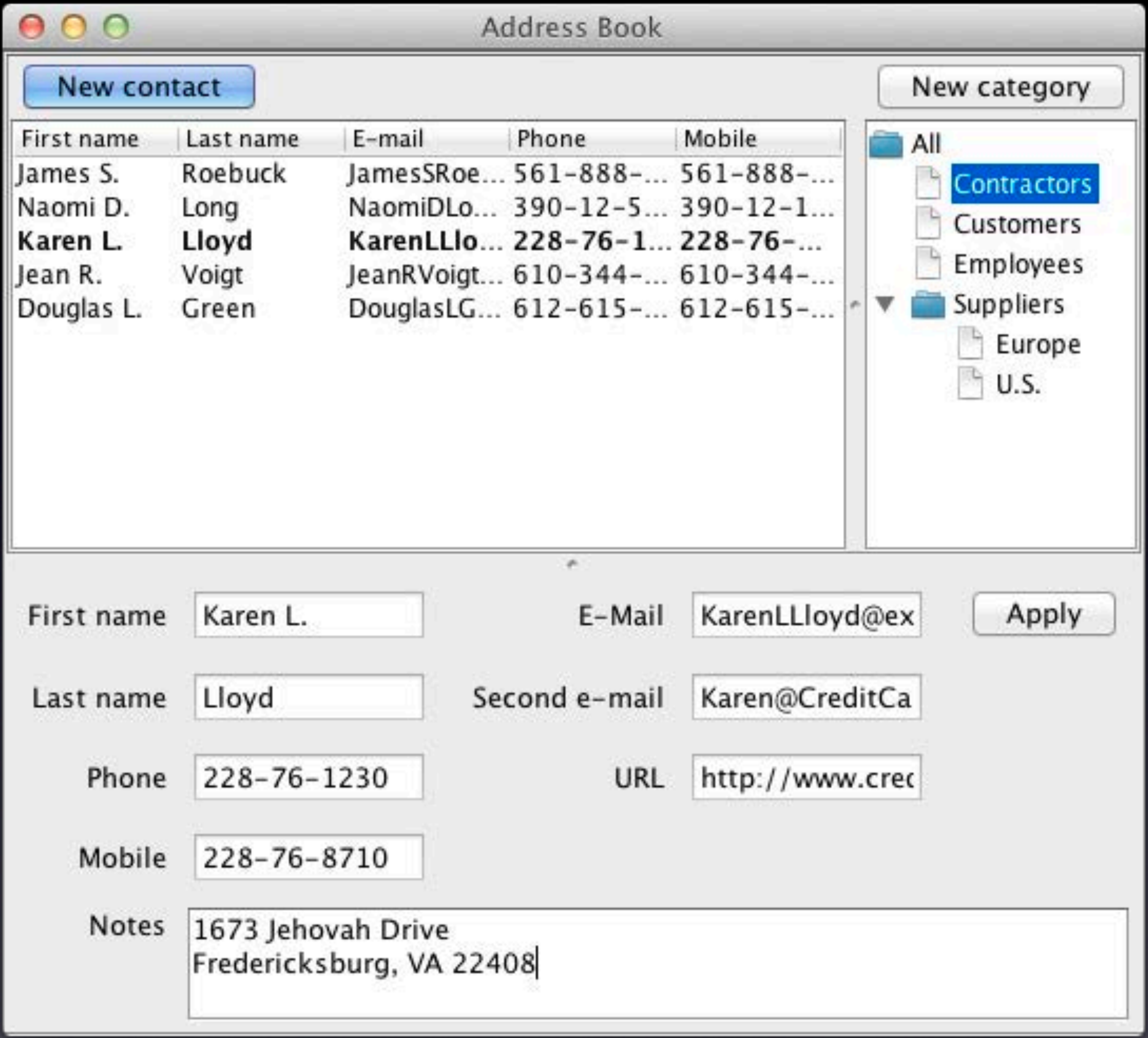
Random Testing

```
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 var64 = 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");
}
```

```
  MainWindow var64 = 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

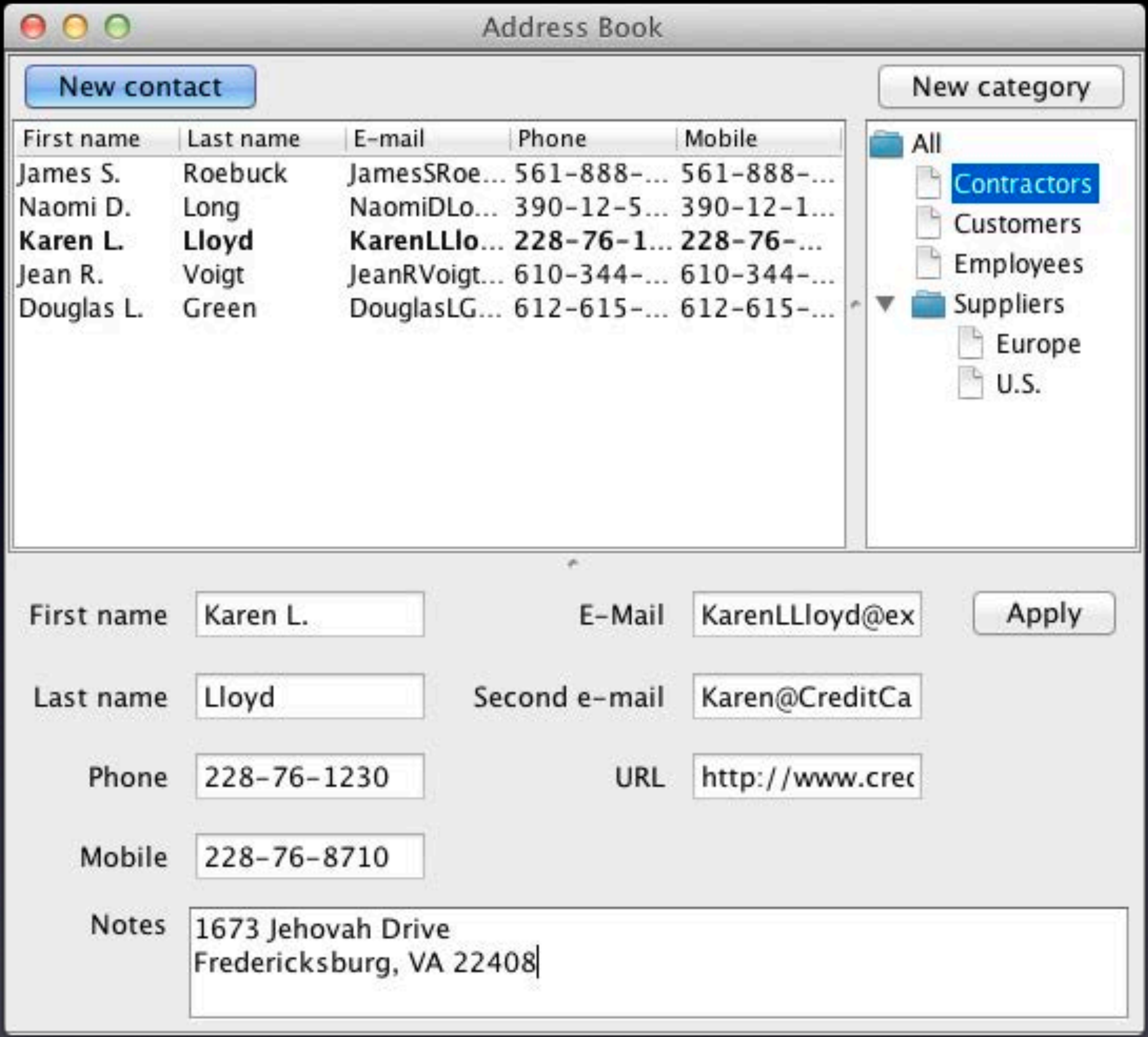
```
public class RandoopTest0 extends TestCase {
  public void test8() throws Throwable {
    if (debug) System.out.printf("%nRandoopTest0.test8");

    AddressBook a1 = new AddressBook();
    AddressBook a2 = new AddressBook();
    Category a1c = a1.createCategory(a1.getRootCategory(), "a1c");
    Category a2c = a2.createCategory(a1c, "a2c");
  }
}
```



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

First name	Karen L.	E-Mail	KarenLLloyd@ex
Last name	Lloyd	Second e-mail	Karen@CreditCa
Phone	228-76-1230	URL	http://www.crec
Mobile	228-76-8710		

Notes
1673 Jehovah Drive
Fredericksburg, VA 22408



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

First name: Karen L.

Last name: Lloyd

Phone: 228-76-1230

Mobile: 228-76-8710

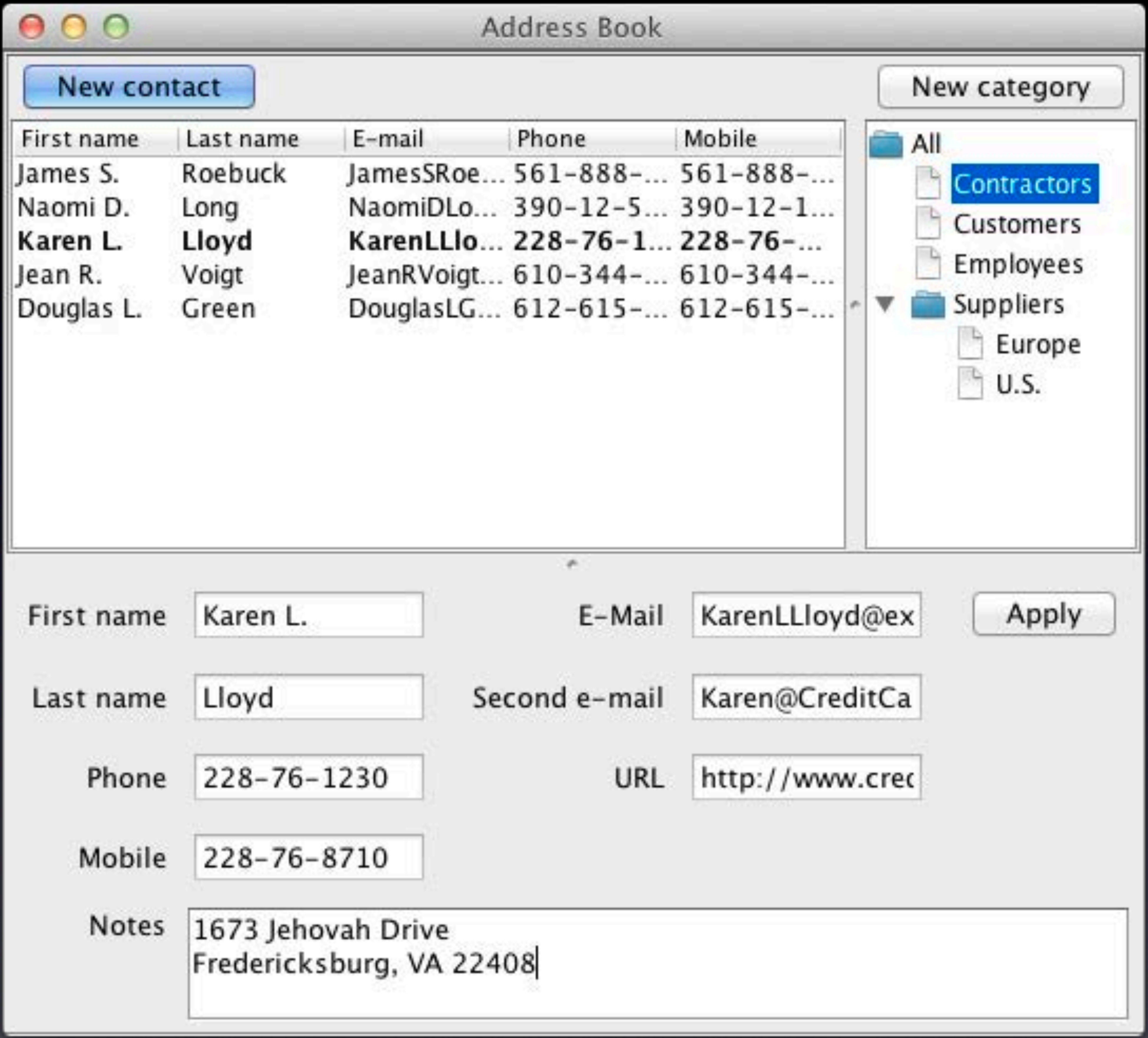
E-Mail: KarenLLloyd@ex

Second e-mail: Karen@CreditCa

URL: http://www.crec

Notes:
1673 Jehovah Drive
Fredericksburg, VA 22408

how many addressbooks?



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

First name: Karen L.

Last name: Lloyd

Phone: 228-76-1230

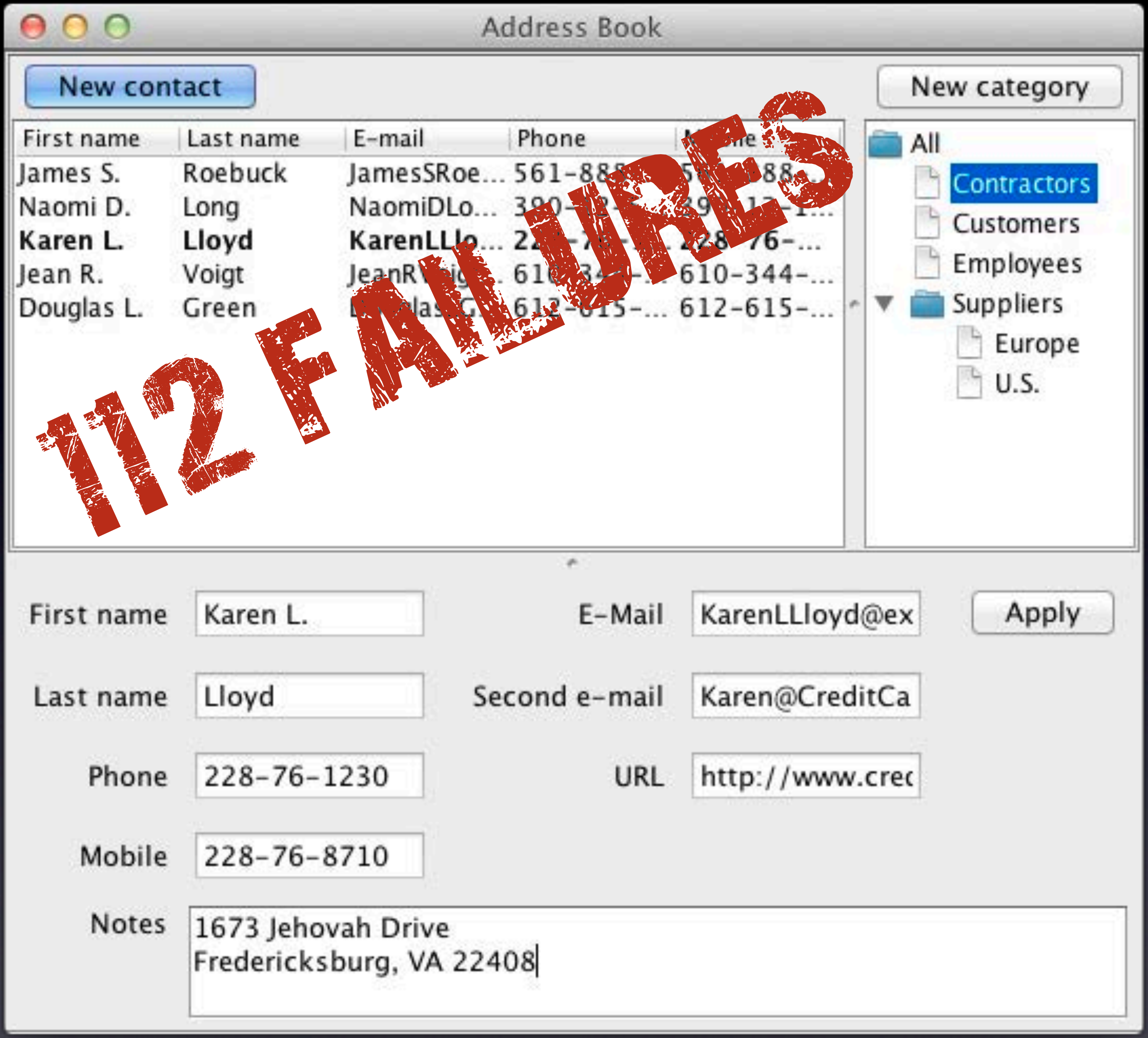
Mobile: 228-76-8710

E-Mail: KarenLLloyd@ex

Second e-mail: Karen@CreditCa

URL: http://www.crec

Notes:
1673 Jehovah Drive
Fredericksburg, VA 22408

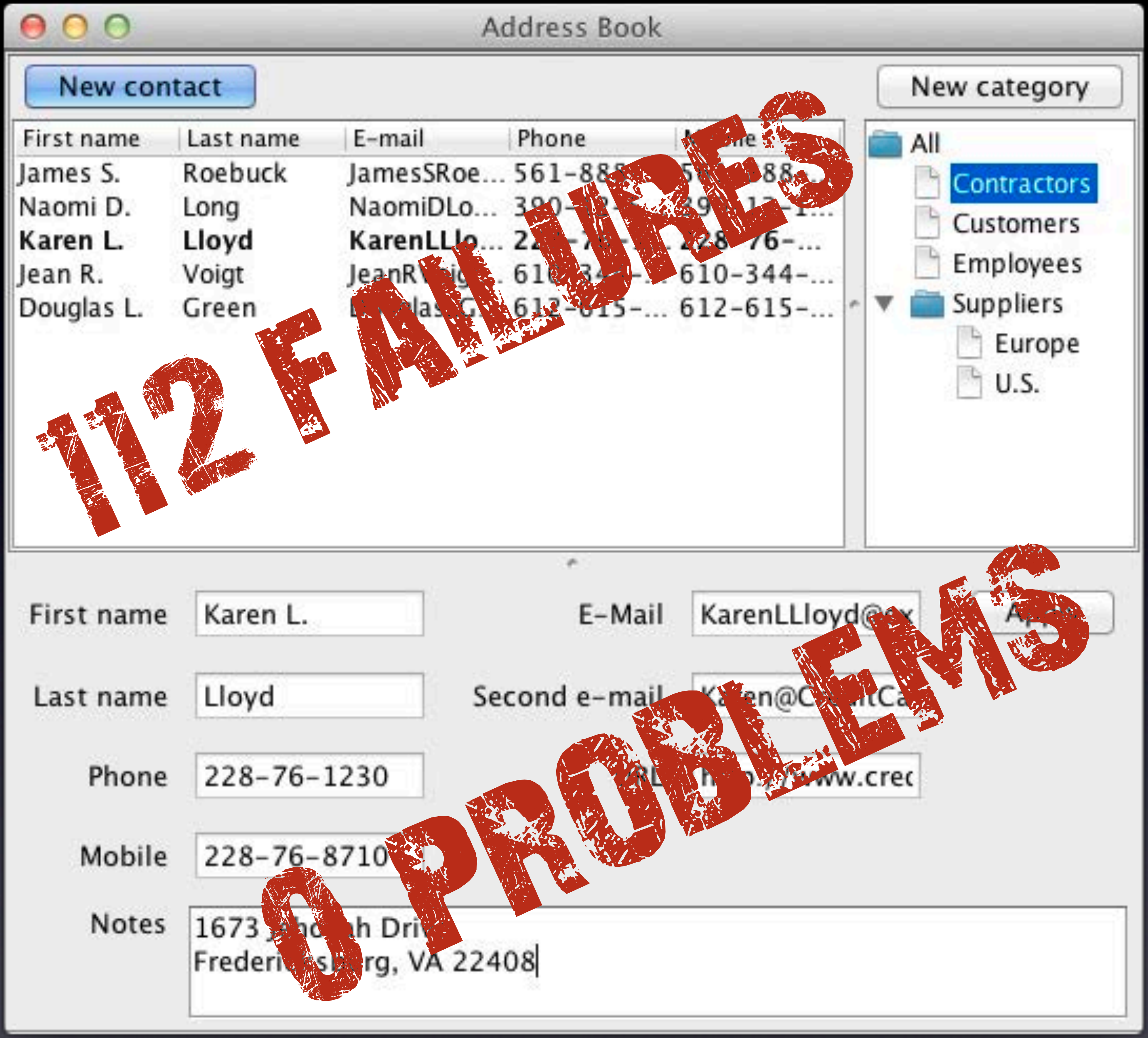


First name	Last name	E-mail	Phone	Mobile
James S.	Roebuck	JamesSRoe...	561-88...	561-88...
Naomi D.	Long	NaomiDLo...	390-...	390-...
Karen L.	**Lloyd**	**KarenLLlo...**	**228-76-...**	**228-76-...**
Jean R.	Voigt	JeanRVoig...	610-344-...	610-344-...
Douglas L.	Green	DouglasG...	612-615-...	612-615-...

First name	Karen L.	E-Mail	KarenLLloyd@ex
Last name	Lloyd	Second e-mail	Karen@CreditCa
Phone	228-76-1230	URL	http://www.crec
Mobile	228-76-8710		

Notes
1673 Jehovah Drive
Fredericksburg, VA 22408

112 FAILURES



First name	Last name	E-mail	Phone	Mobile
James S.	Roebuck	JamesSRoe...	561-88...	561-88...
Naomi D.	Long	NaomiDLo...	390-...	390-1...
Karen L.	**Lloyd**	**KarenLLlo...**	**228-76-...**	**228-76-...**
Jean R.	Voigt	JeanRVoig...	610-344-...	610-344-...
Douglas L.	Green	DouglasG...	612-615-...	612-615-...

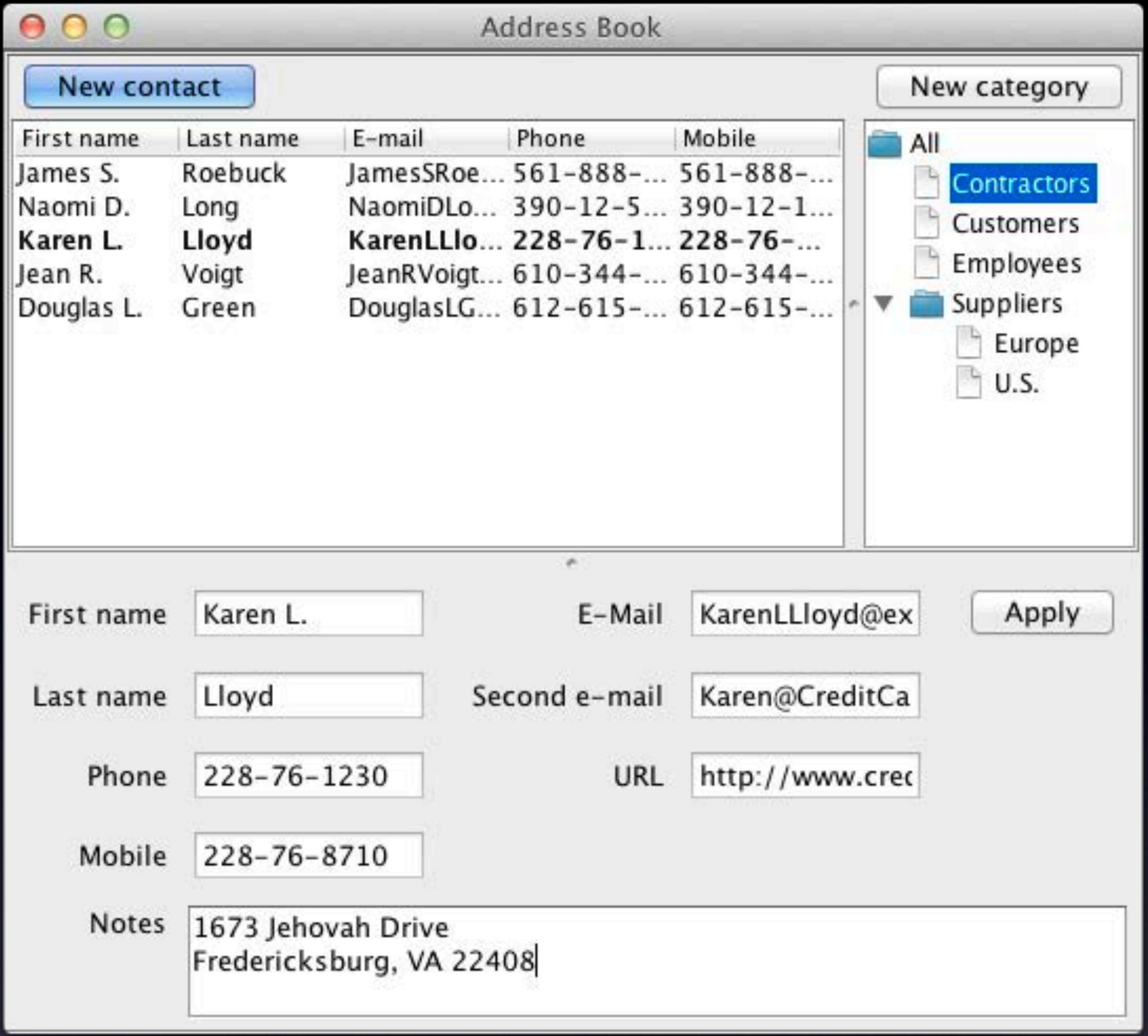
First name	Karen L.	E-Mail	KarenLLloyd@ex
Last name	Lloyd	Second e-mail	Karen@C[illegible]tCa
Phone	228-76-1230	URL	http://www.crec
Mobile	228-76-8710		

Notes

1673 [illegible] Drive
Fredericksburg, VA 22408

112 FAILURES

0 PROBLEMS



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

First name: Karen L.

E-Mail: KarenLLloyd@ex

Last name: Lloyd

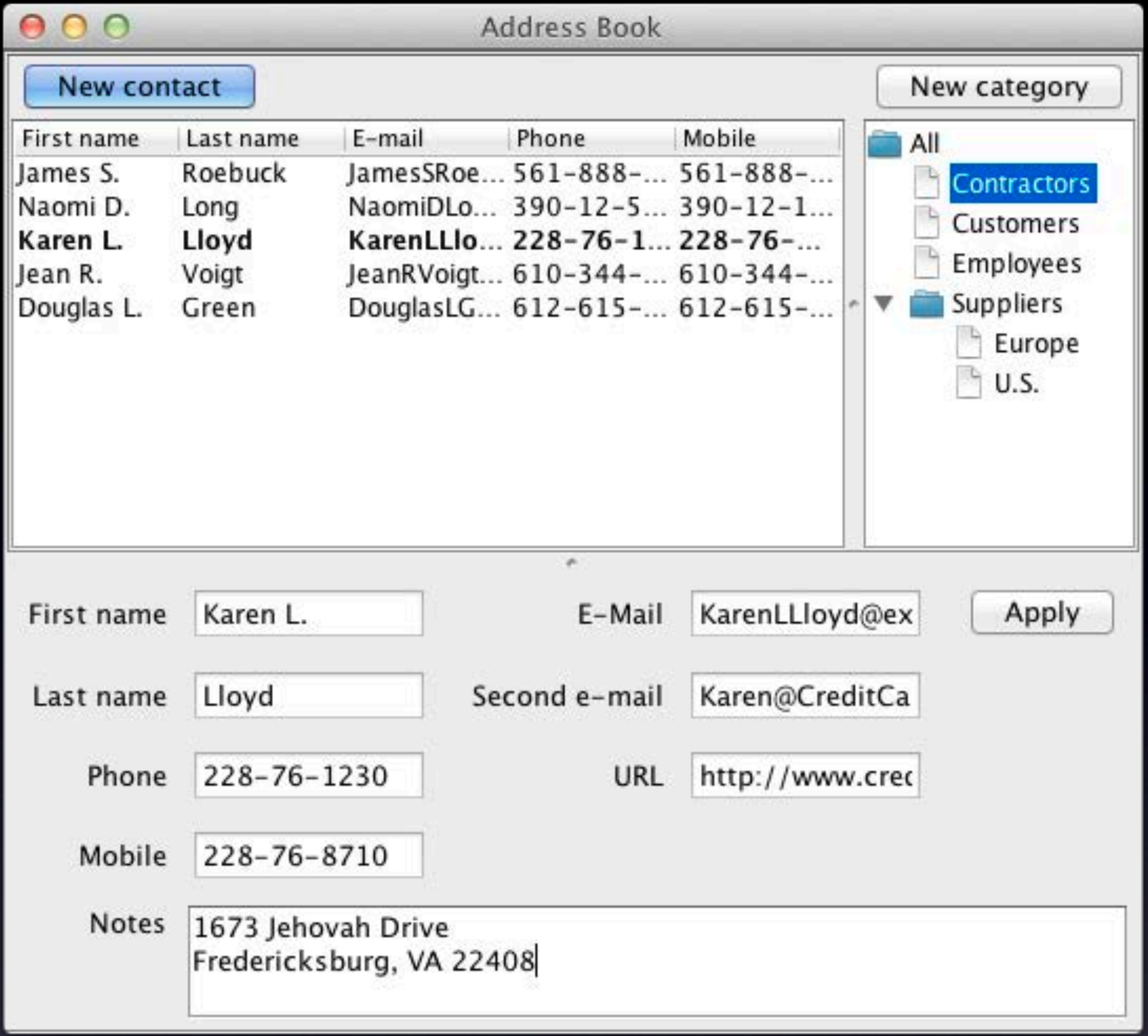
Second e-mail: Karen@CreditCa

Phone: 228-76-1230

URL: http://www.crec

Mobile: 228-76-8710

Notes: 1673 Jehovah Drive
Fredericksburg, VA 22408



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**	**KarenLLlo…**	**228-76-1…**	**228-76-…**
Jean R.	Voigt	JeanRVoigt…	610-344-…	610-344-…
Douglas L.	Green	DouglasLG…	612-615-…	612-615-…

First name: Karen L.

E-Mail: KarenLLloyd@ex

Last name: Lloyd

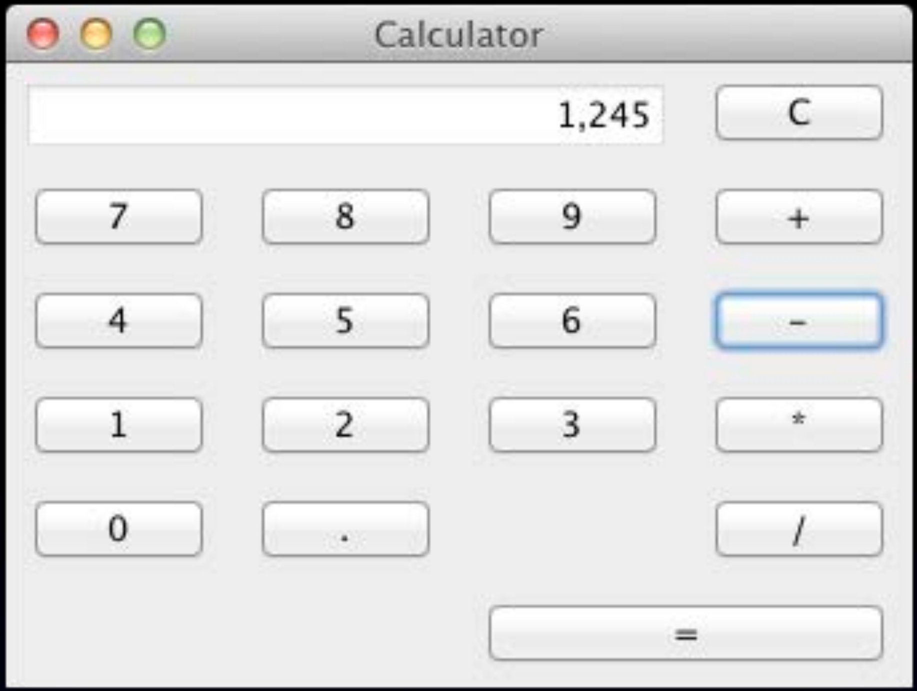
Second e-mail: Karen@CreditCa

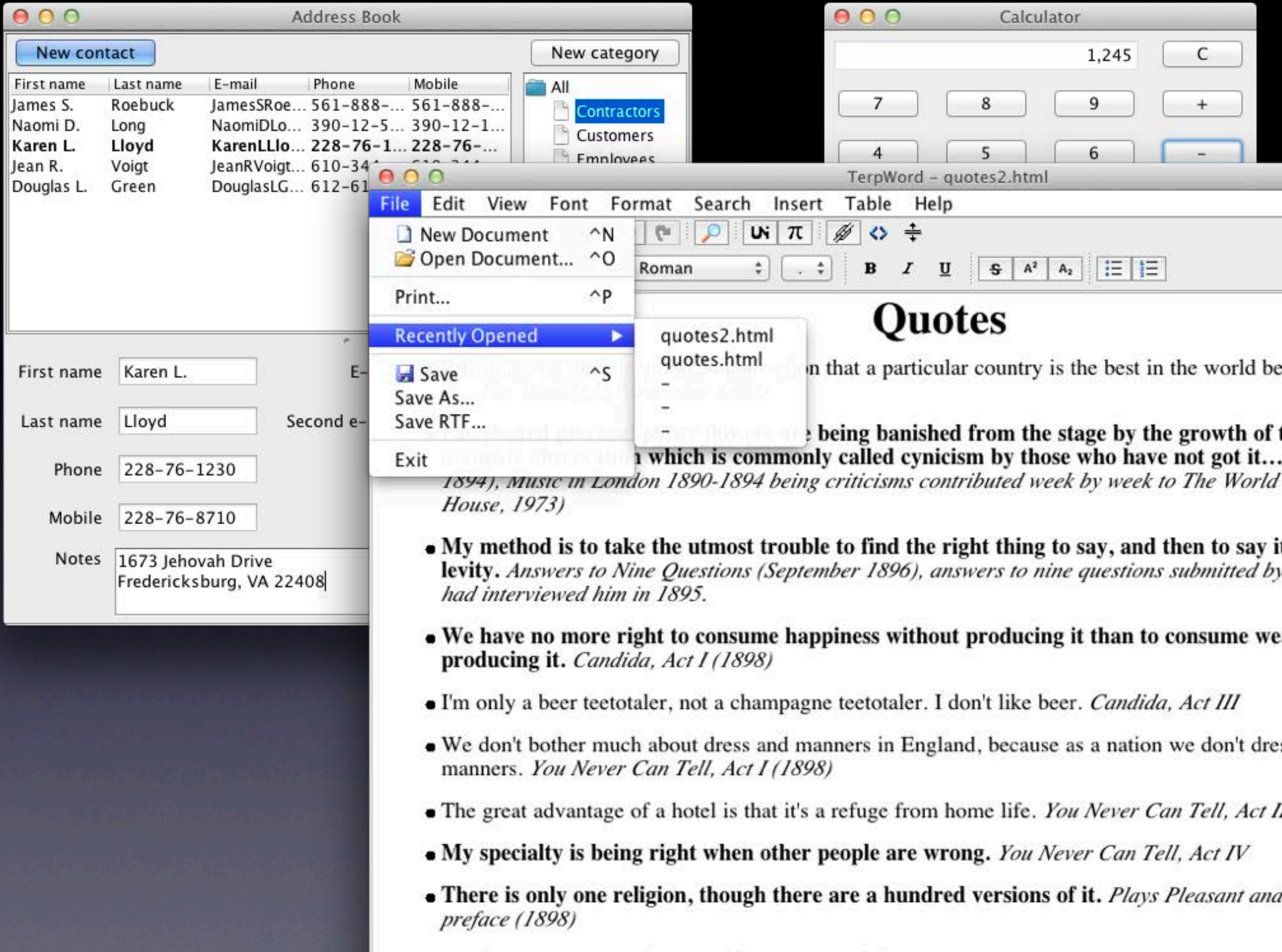
Phone: 228-76-1230

URL: http://www.crec

Mobile: 228-76-8710

Notes: 1673 Jehovah Drive
Fredericksburg, VA 22408





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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-34	
Douglas L.	Green	DouglasLG...	612-61	

First name: Karen L.

Last name: Lloyd

Phone: 228-76-1230

Mobile: 228-76-8710

Notes: 1673 Jehovah Drive
Fredericksburg, VA 22408

E-

Second e-

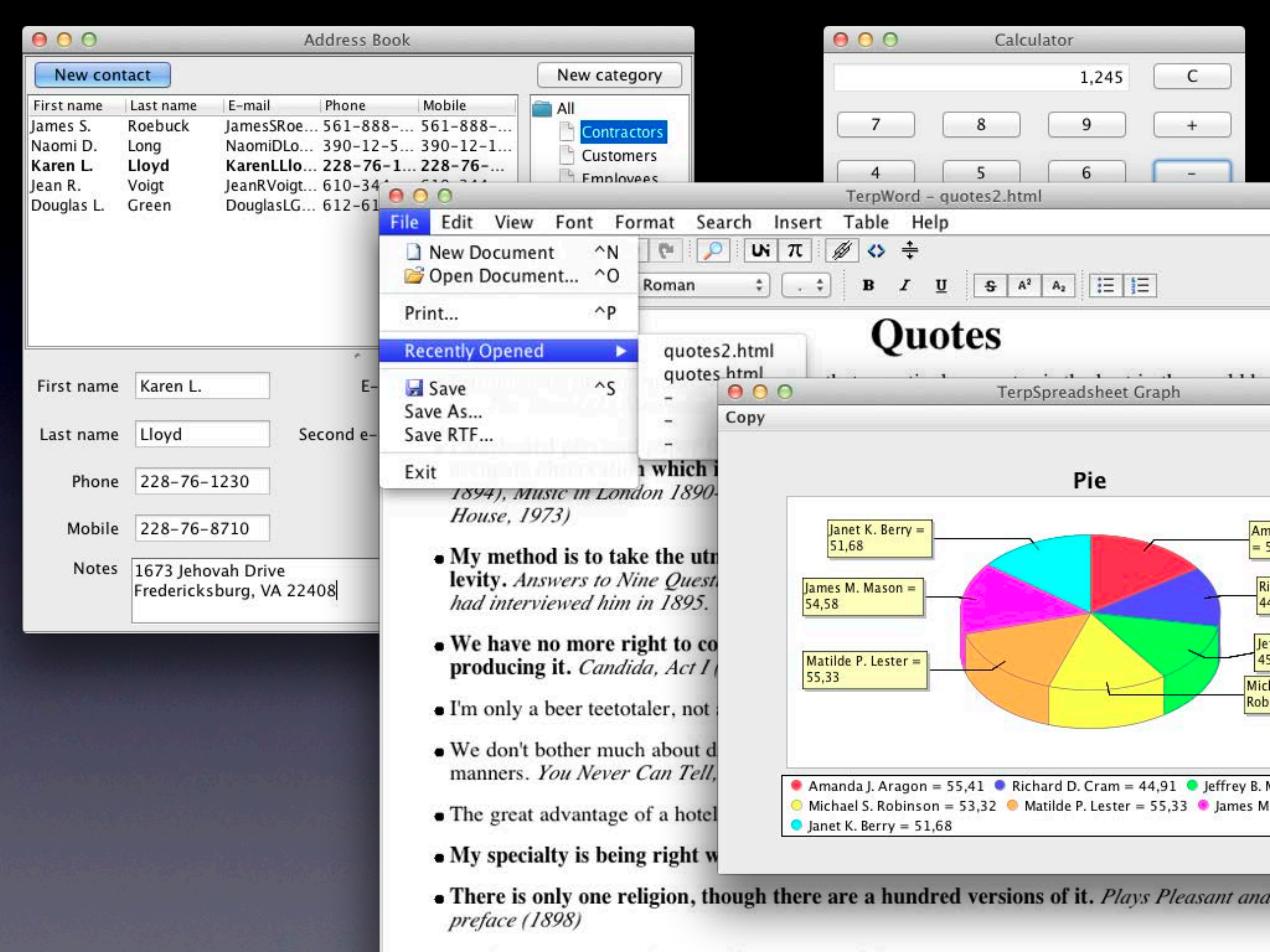
1,245

Quotes

n that a particular country is the best in the world be

e being banished from the stage by the growth of t
n which is commonly called cynicism by those who have not got it… *1894), Music in London 1890-1894 being criticisms contributed week by week to The World House, 1973)*

- **My method is to take the utmost trouble to find the right thing to say, and then to say it levity.** *Answers to Nine Questions (September 1896), answers to nine questions submitted by had interviewed him in 1895.*
- **We have no more right to consume happiness without producing it than to consume we producing it.** *Candida, Act I (1898)*
- I'm only a beer teetotaler, not a champagne teetotaler. I don't like beer. *Candida, Act III*
- We don't bother much about dress and manners in England, because as a nation we don't dre manners. *You Never Can Tell, Act I (1898)*
- The great advantage of a hotel is that it's a refuge from home life. *You Never Can Tell, Act I*
- **My specialty is being right when other people are wrong.** *You Never Can Tell, Act IV*
- **There is only one religion, though there are a hundred versions of it.** *Plays Pleasant and preface (1898)*



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-34	
Douglas L.	Green	DouglasLG...	612-61	

First name: Karen L.

Last name: Lloyd

Second e-

Phone: 228-76-1230

Mobile: 228-76-8710

Notes: 1673 Jehovah Drive
Fredericksburg, VA 22408

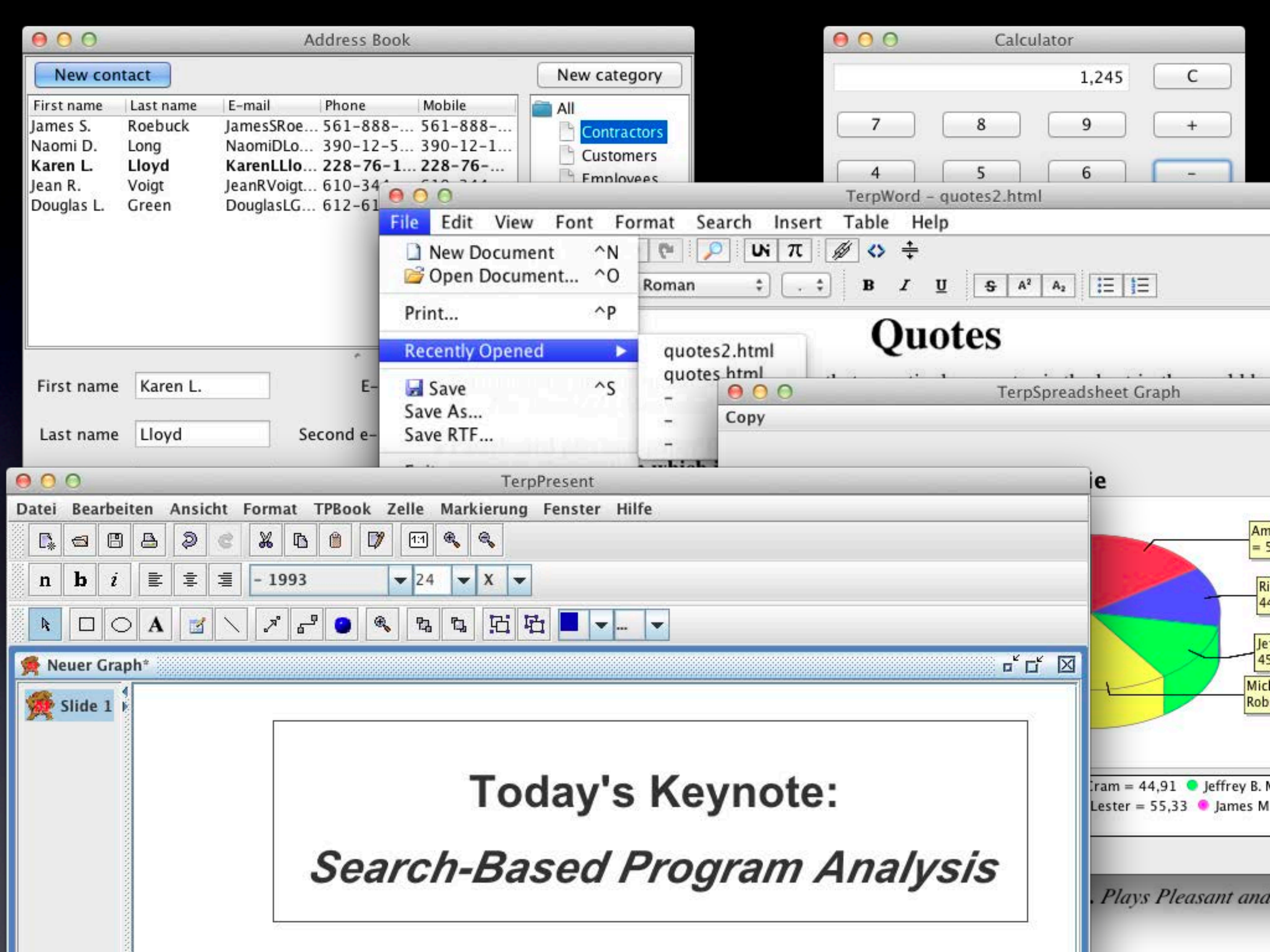
1,245

Quotes

1894), Music in London 1890-
House, 1973)

- **My method is to take the utr levity.** *Answers to Nine Questi had interviewed him in 1895.*
- **We have no more right to co producing it.** *Candida, Act I (*
- I'm only a beer teetotaler, not
- We don't bother much about d manners. *You Never Can Tell,*
- The great advantage of a hotel
- **My specialty is being right w**
- **There is only one religion, though there are a hundred versions of it.** *Plays Pleasant and preface (1898)*

● Amanda J. Aragon = 55,41 ● Richard D. Cram = 44,91 ● Jeffrey B. M
● Michael S. Robinson = 53,32 ● Matilde P. Lester = 55,33 ● James M
● Janet K. Berry = 51,68



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-34	
Douglas L.	Green	DouglasLG...	612-61	

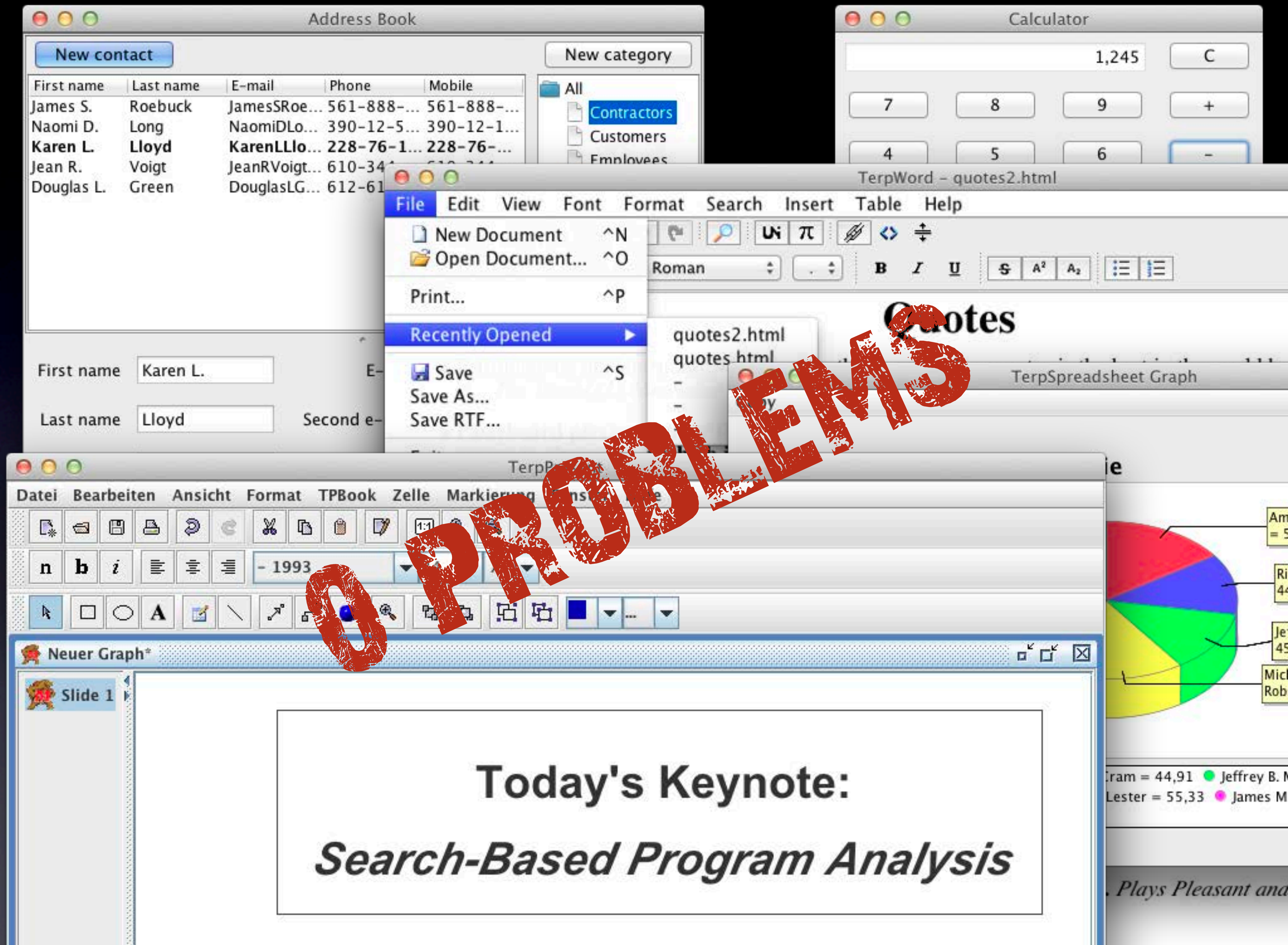
First name: Karen L.

Last name: Lloyd

Second e-

Today's Keynote:

Search-Based Program Analysis



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-34	
Douglas L.	Green	DouglasLG...	612-61	

First name: Karen L.

Last name: Lloyd

Today's Keynote:

Search-Based Program Analysis

0 PROBLEMS

Search-based System Testing

Joint work with Florian Gross and Gordon Fraser

Search-based System Testing

- Generate tests at the user interface level

Joint work with Florian Gross and Gordon Fraser

Search-based System Testing

- Generate tests at the user interface level
- Aim for *code coverage* and *GUI coverage*

Joint work with Florian Gross and Gordon Fraser

Search-based System Testing

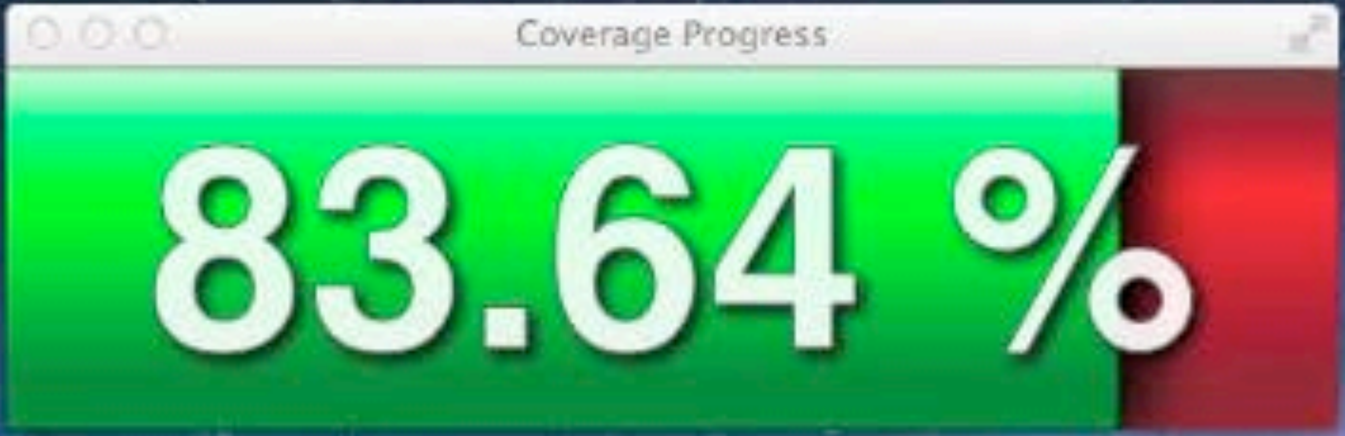
- Generate tests at the user interface level
- Aim for *code coverage* and *GUI coverage*
- Synthesize artificial input events

Joint work with Florian Gross and Gordon Fraser

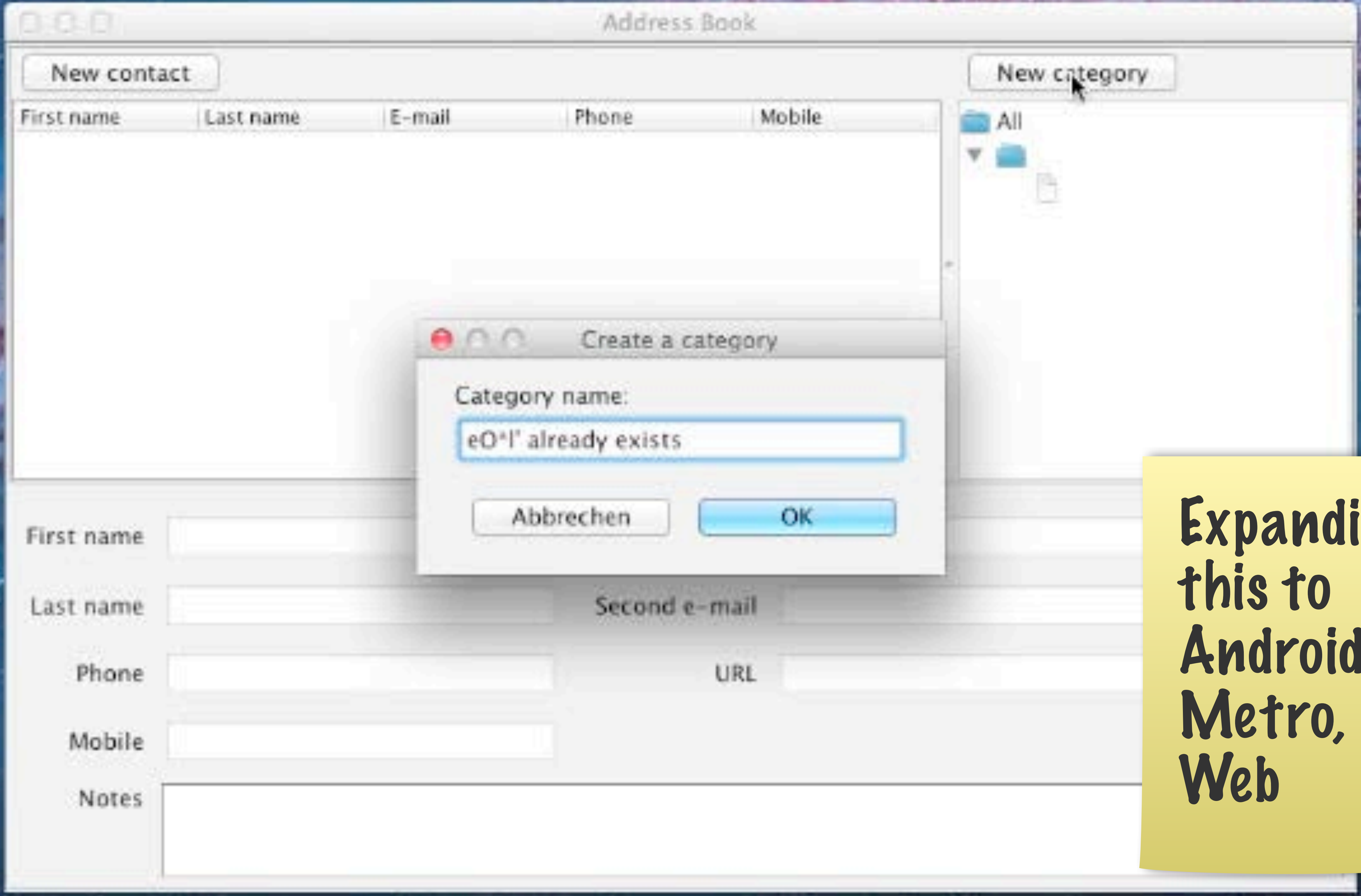
Search-based System Testing

- Generate tests at the user interface level
- Aim for *code coverage* and *GUI coverage*
- Synthesize artificial input events
- Any test generated is a valid input

Joint work with Florian Gross and Gordon Fraser



83.64 %



Category name: eO*I' already exists

Expanding this to Android, Metro, Web

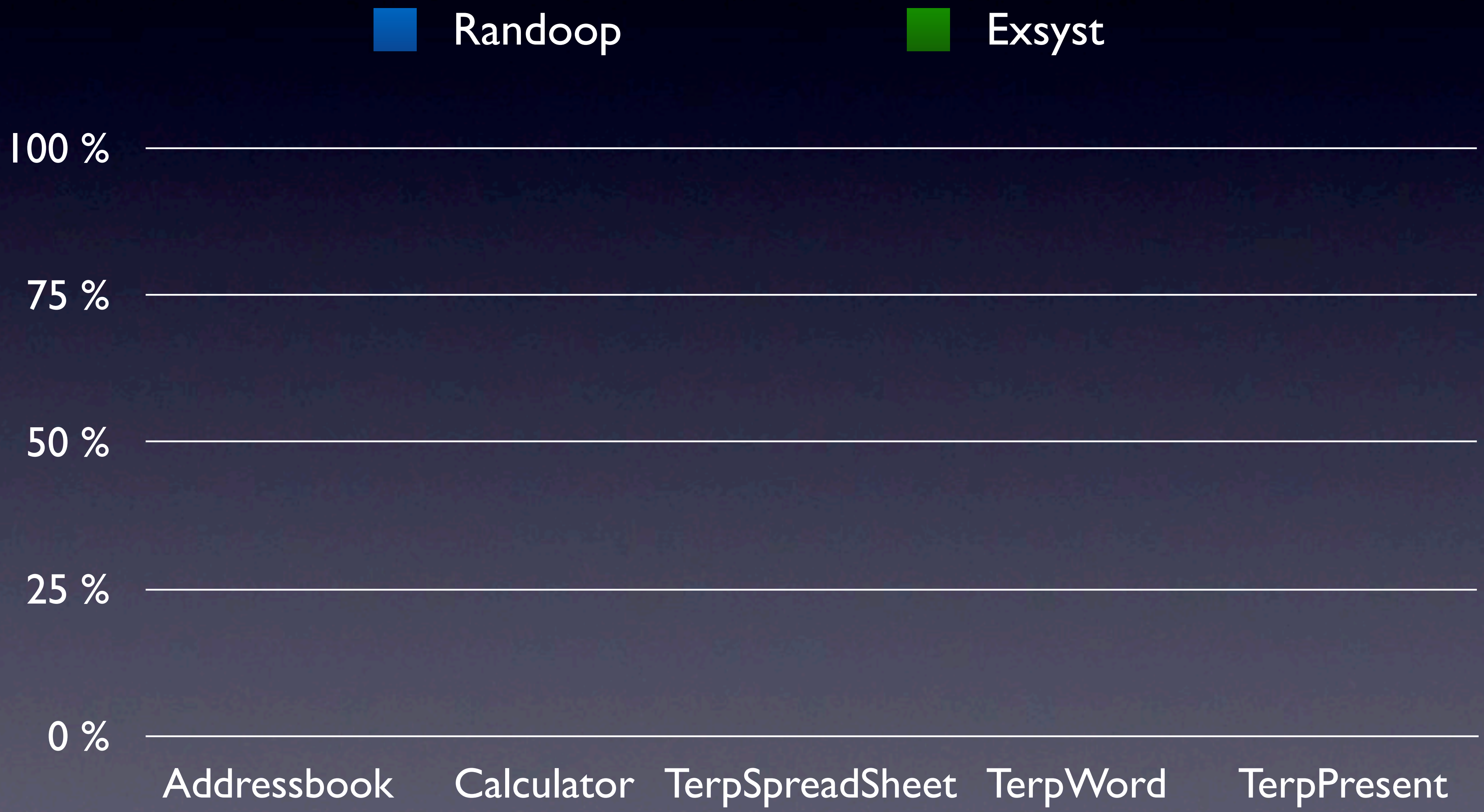
See formal demo at ICSE 2012

Coverage achieved

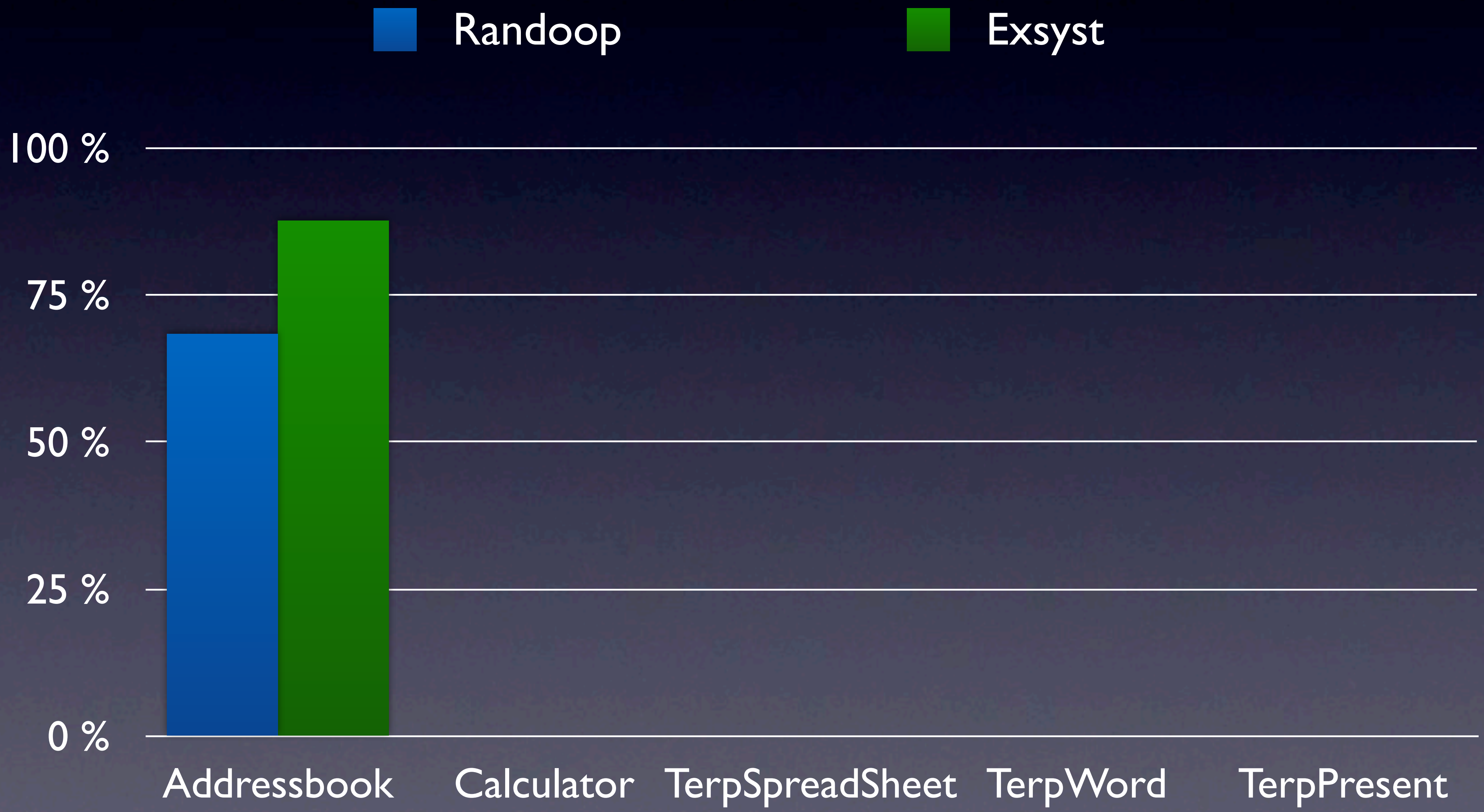
🟦 Randoop

🟩 Exsyst

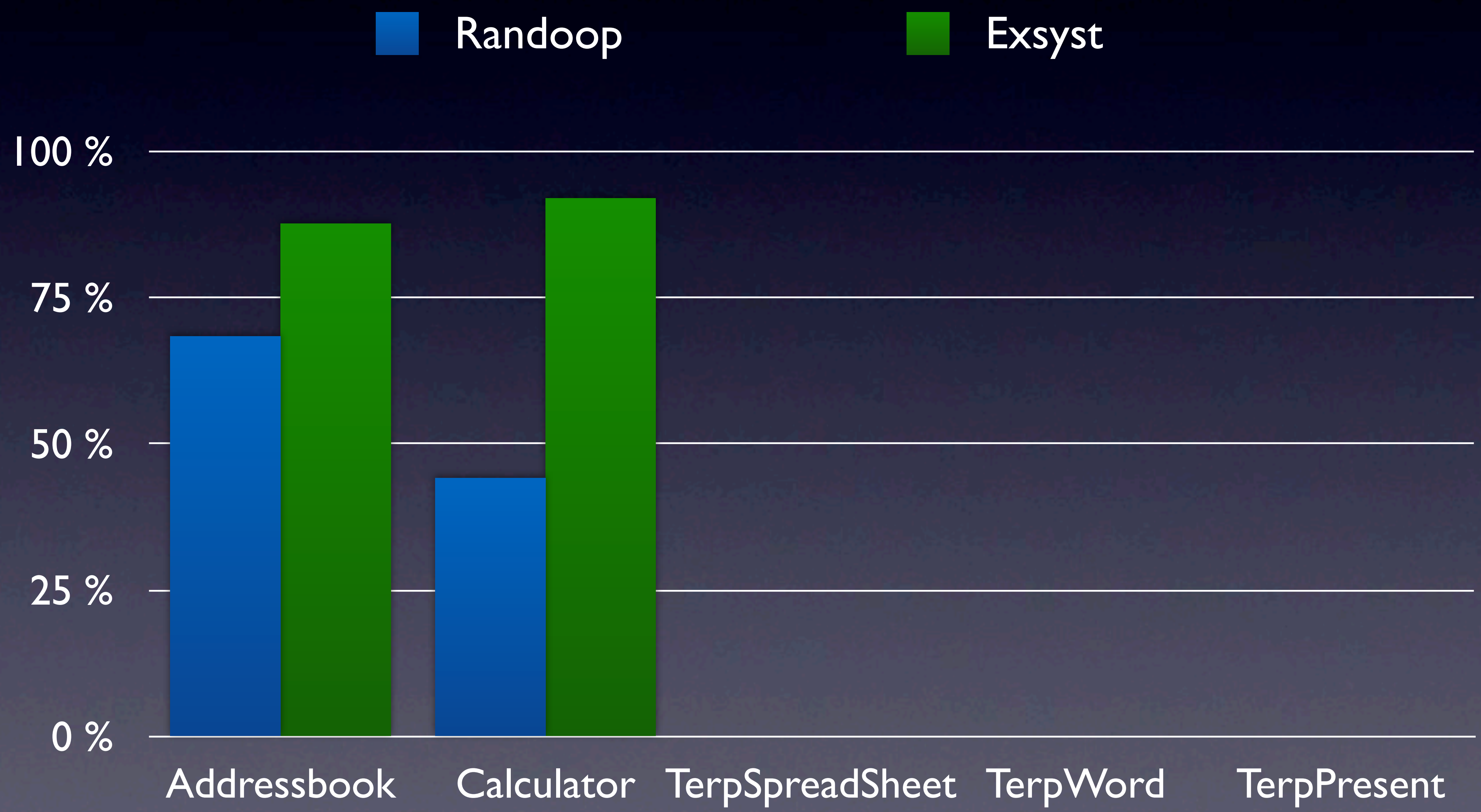
Coverage achieved



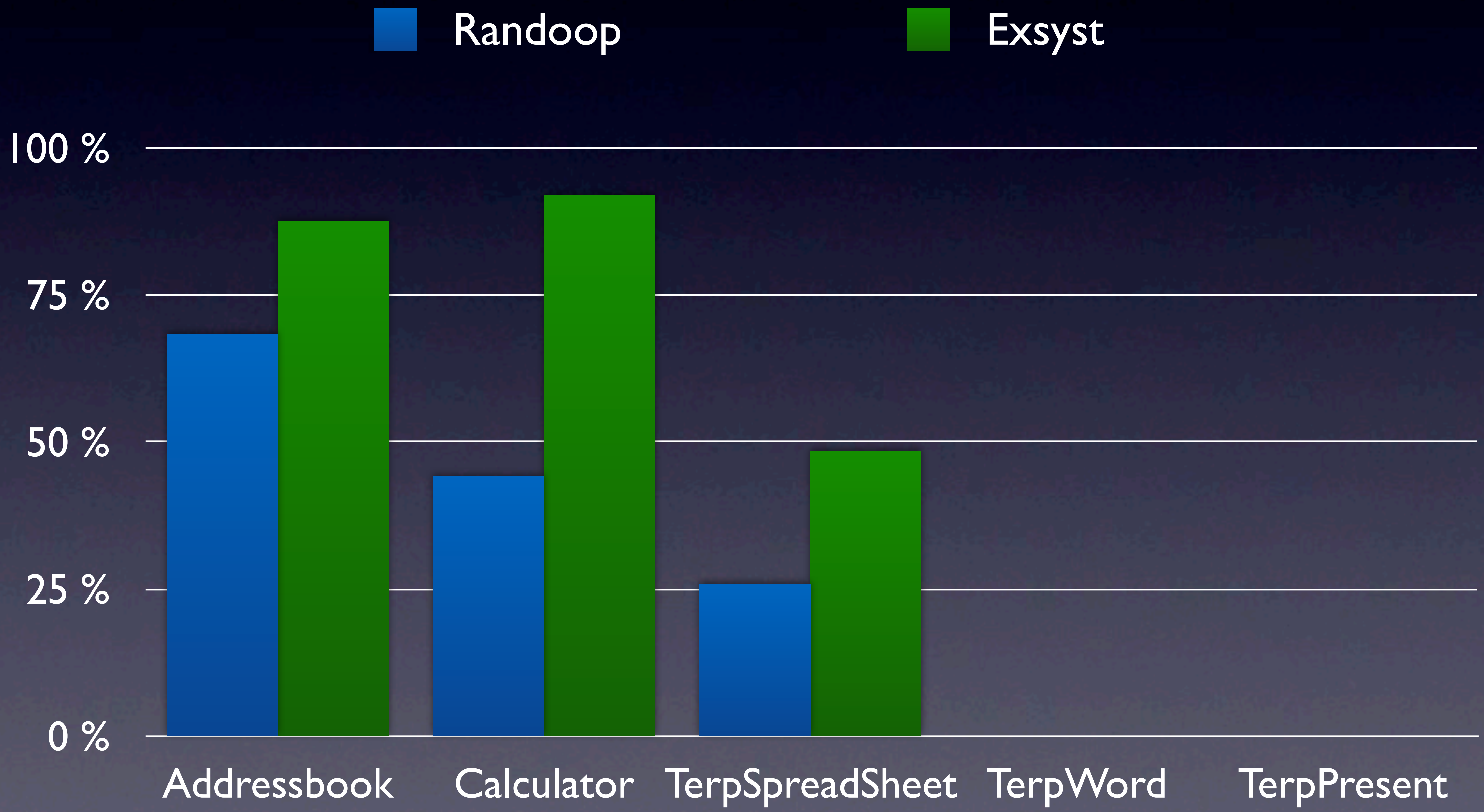
Coverage achieved



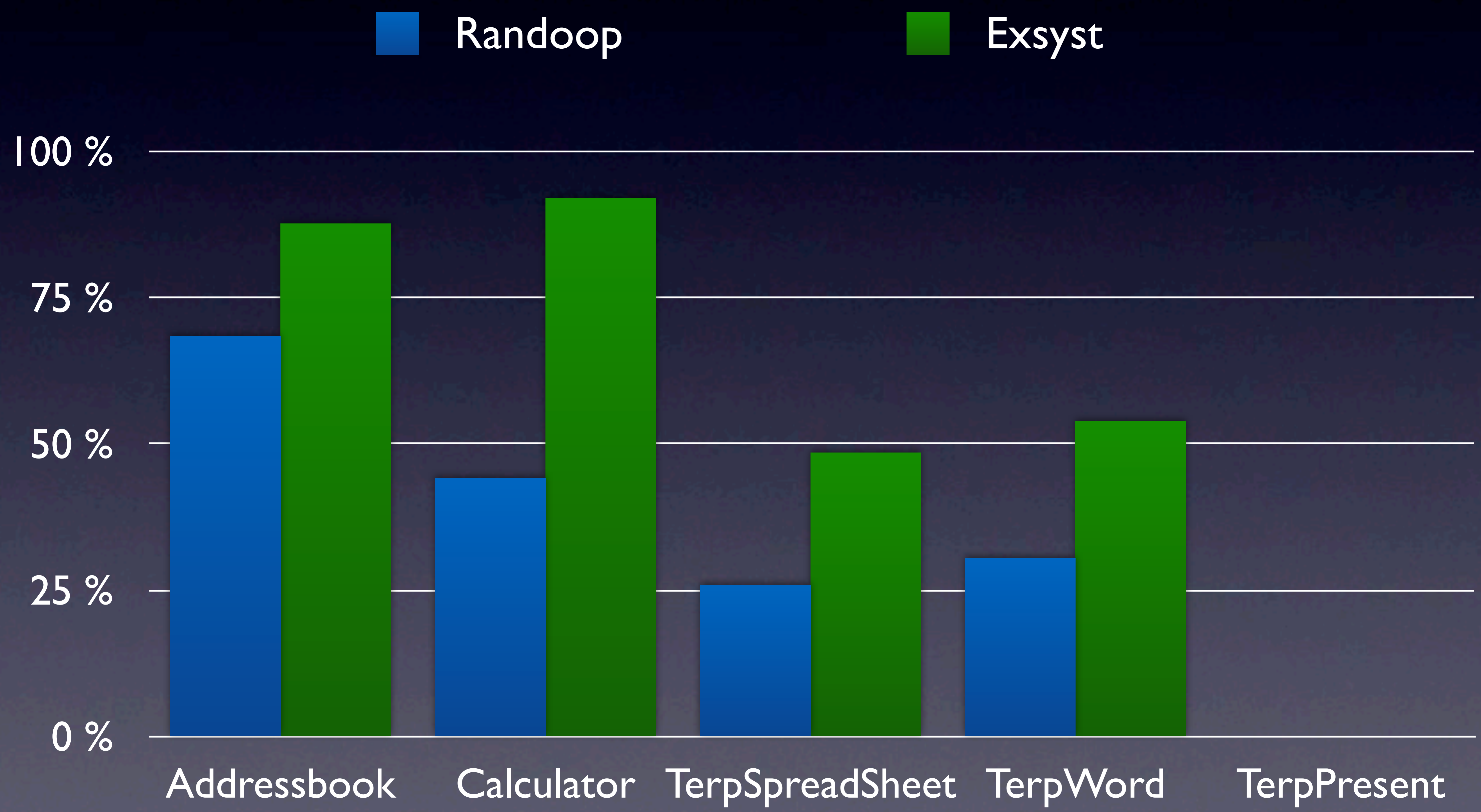
Coverage achieved



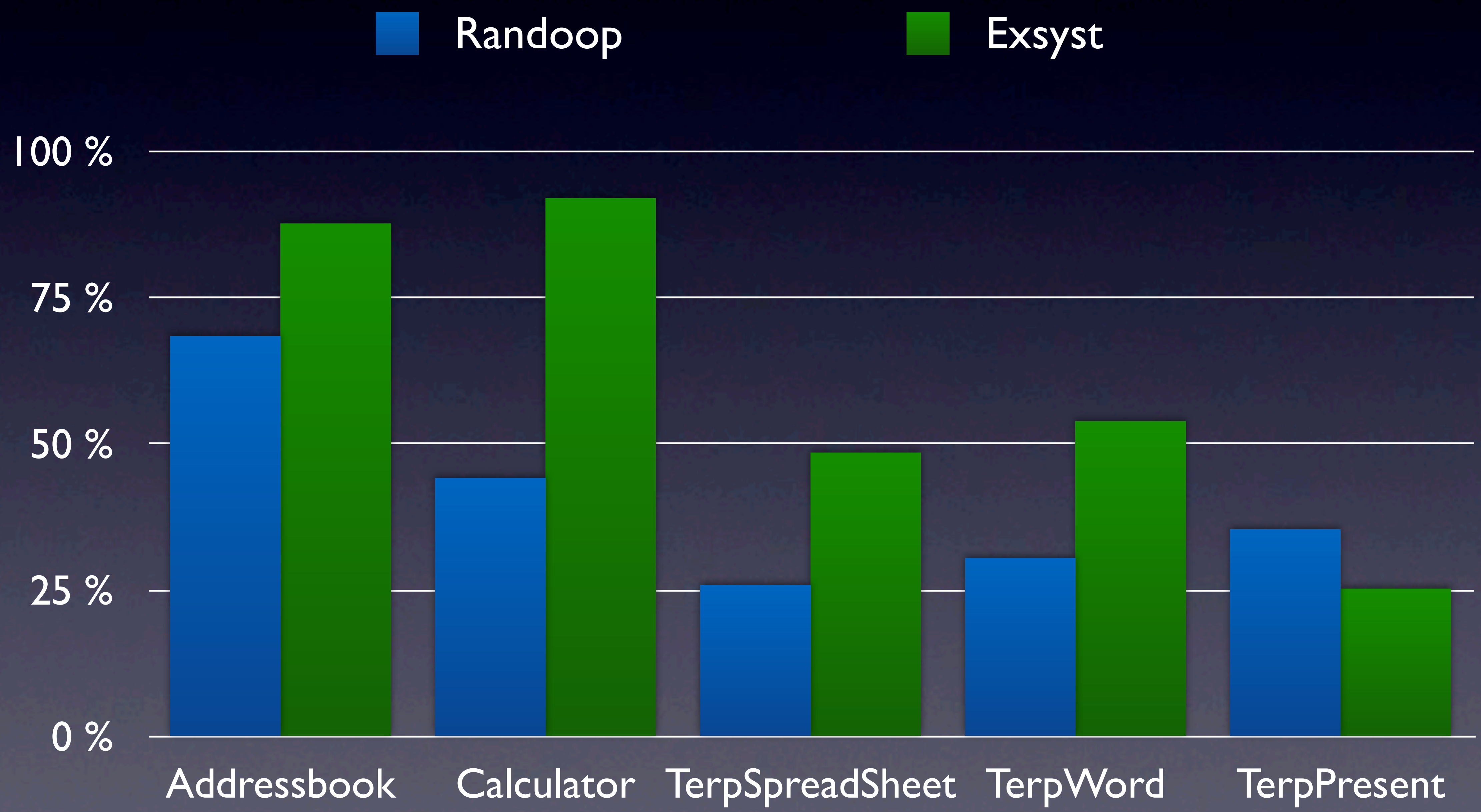
Coverage achieved

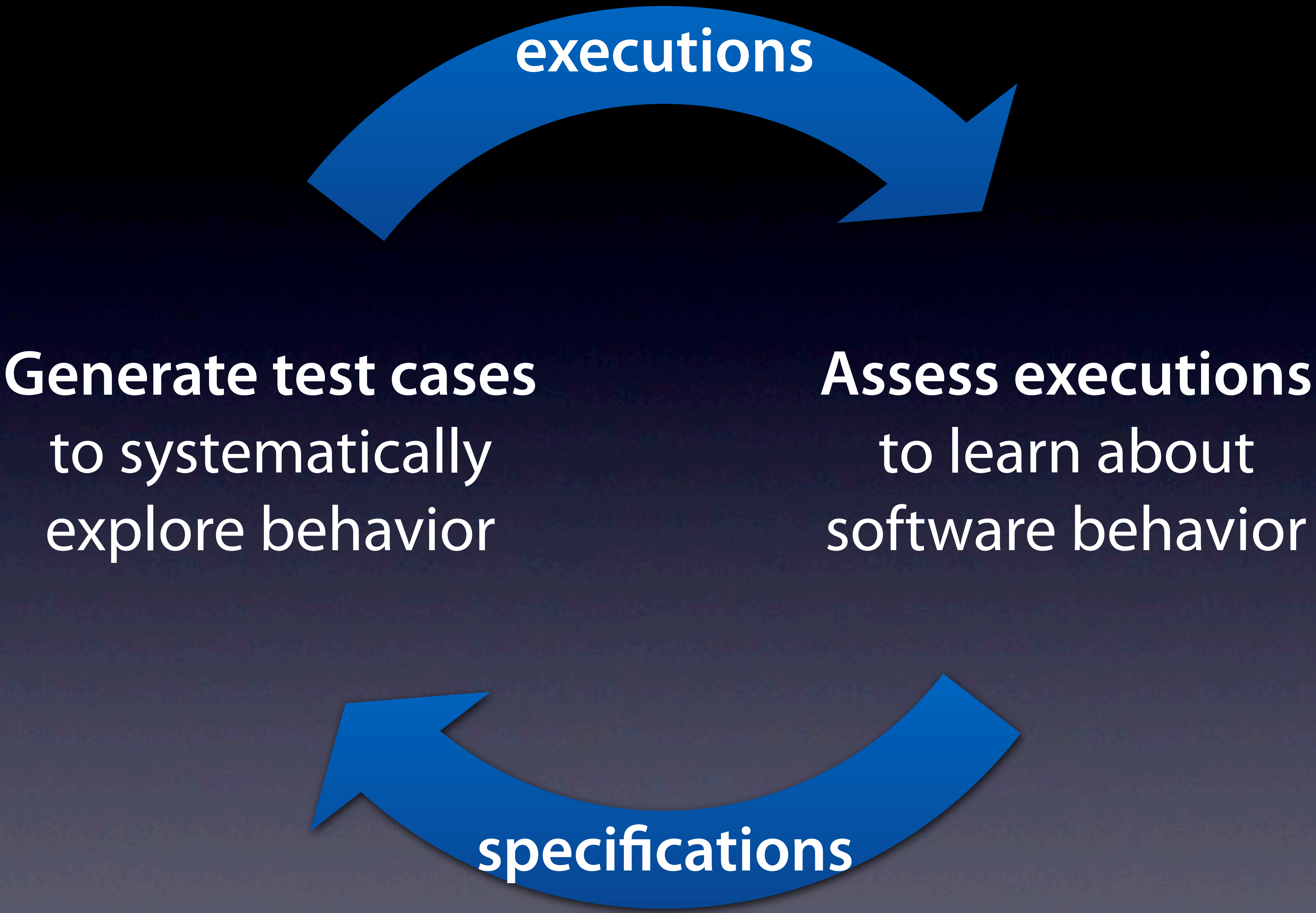


Coverage achieved



Coverage achieved



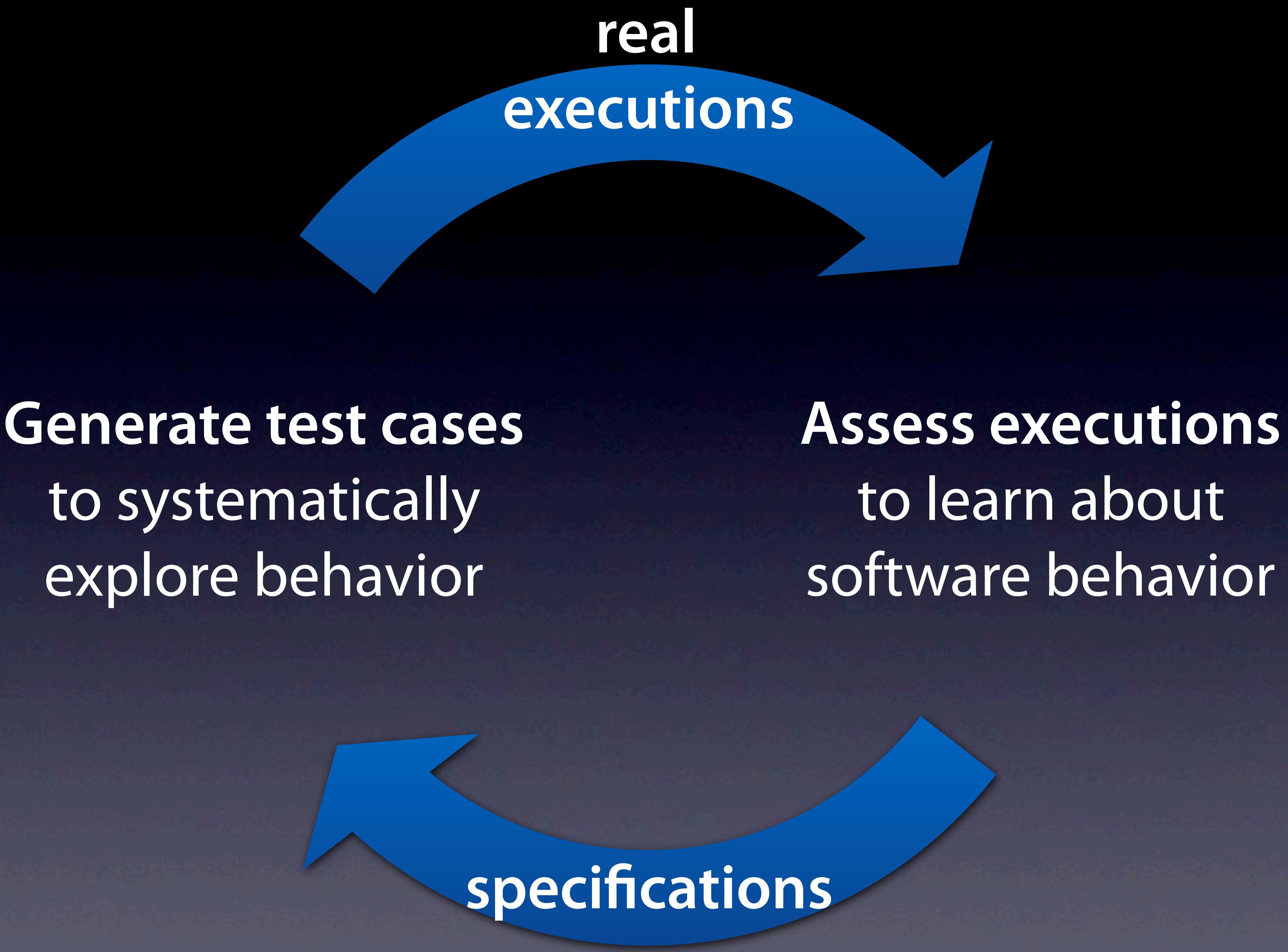


executions

Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior

specifications

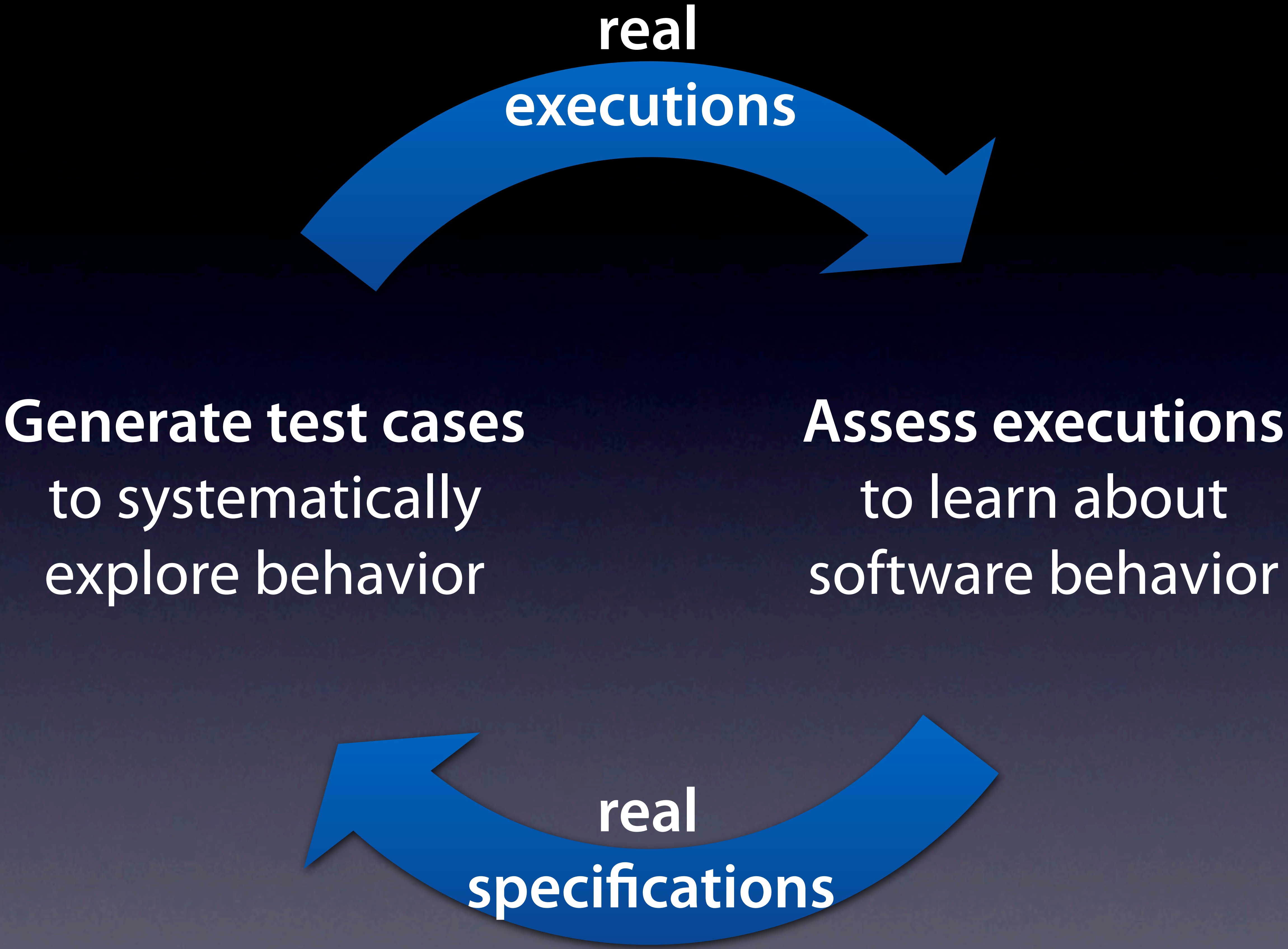


real executions

Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior

specifications



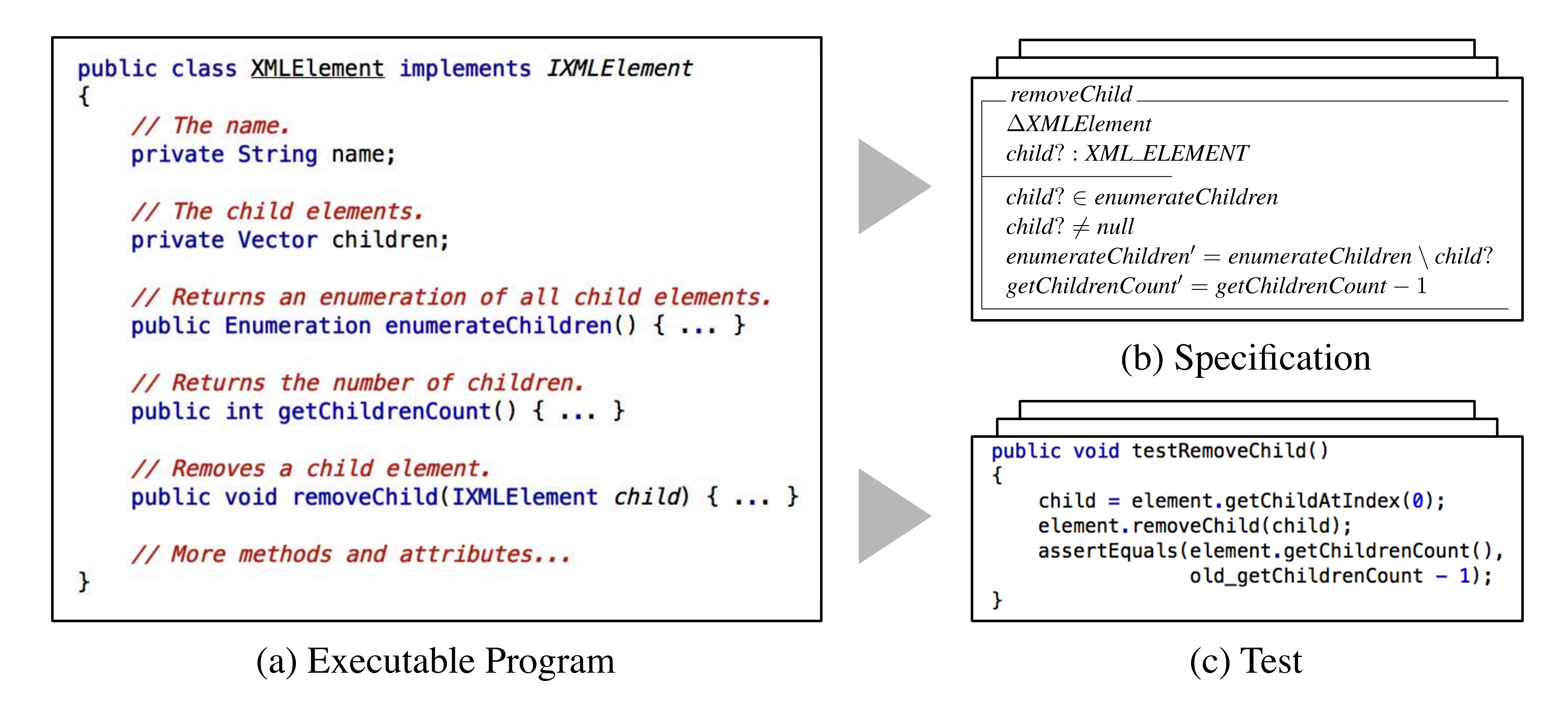
real executions

Generate test cases
to systematically
explore behavior

Assess executions
to learn about
software behavior

real specifications

Carving Invariants



```
public class XMLElement implements IXMLElement
{
    // The name.
    private String name;

    // The child elements.
    private Vector children;

    // Returns an enumeration of all child elements.
    public Enumeration enumerateChildren() { ... }

    // Returns the number of children.
    public int getChildrenCount() { ... }

    // Removes a child element.
    public void removeChild(IXMLElement child) { ... }

    // More methods and attributes...
}
```

(a) Executable Program

removeChild

$\Delta XMLElement$

$child? : XML\_ELEMENT$

$child? \in enumerateChildren$

$child? \neq null$

$enumerateChildren' = enumerateChildren \setminus child?$

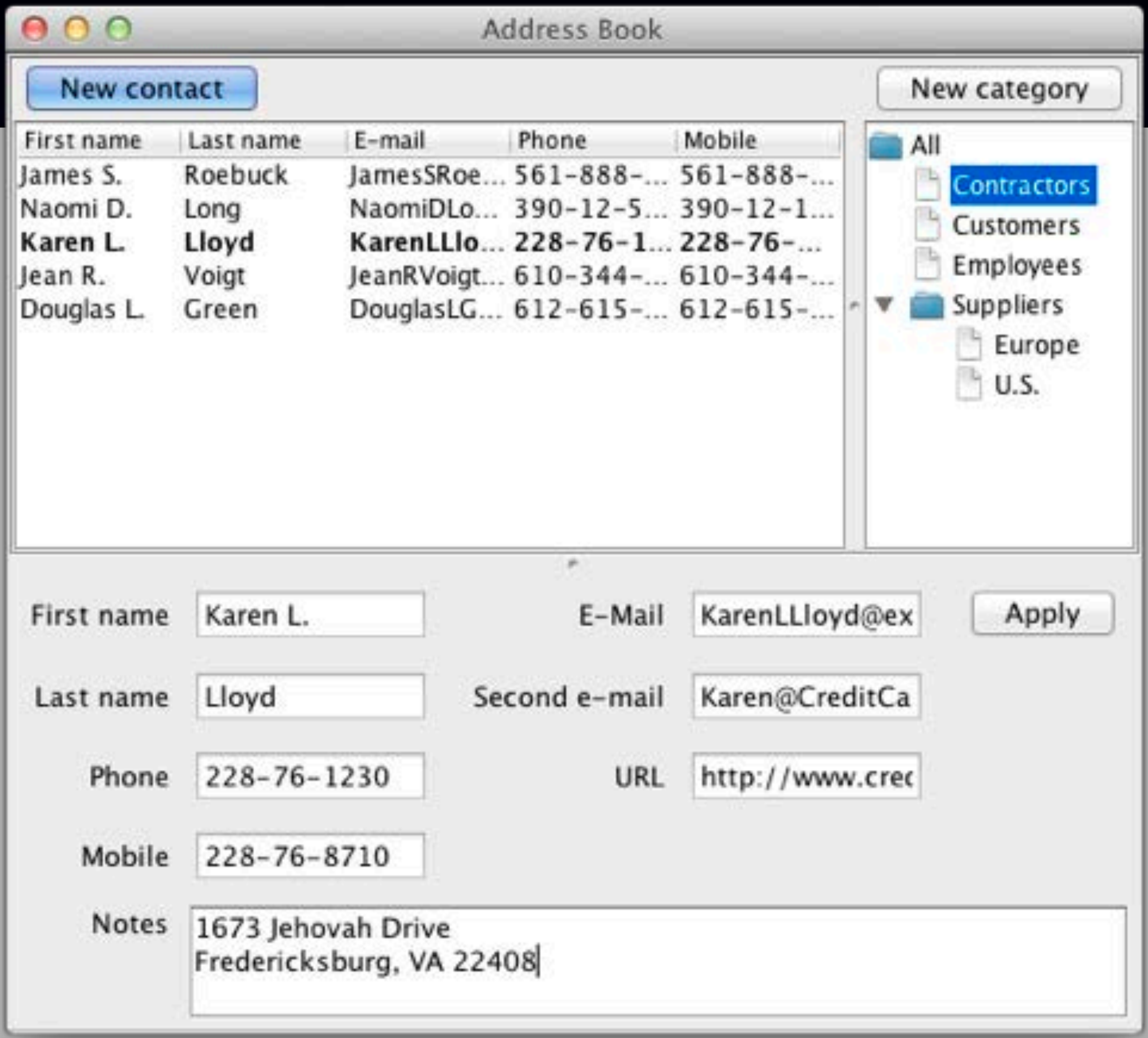
$getChildrenCount' = getChildrenCount - 1$

(b) Specification

```
public void testRemoveChild()
{
    child = element.getChildAtIndex(0);
    element.removeChild(child);
    assertEquals(element.getChildrenCount(),
                 old_getChildrenCount - 1);
}
```

(c) Test

Carving Invariants



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**	**KarenLLlo...**	**228-76-1...**	**228-76-...**
Jean R.	Voigt	JeanRVoigt...	610-344-...	610-344-...
Douglas L.	Green	DouglasLG...	612-615-...	612-615-...

First name: Karen L.

E-Mail: KarenLLloyd@ex

Last name: Lloyd

Second e-mail: Karen@CreditCa

Phone: 228-76-1230

URL: http://www.crec

Mobile: 228-76-8710

Notes: 1673 Jehovah Drive
Fredericksburg, VA 22408

(a) Executable Program

removeChild

$\Delta XMLElement$

$child? : XML\_ELEMENT$

$child? \in enumerateChildren$

$child? \neq null$

$enumerateChildren' = enumerateChildren \setminus child?$

$getChildrenCount' = getChildrenCount - 1$

(b) Specification

```
public void testRemoveChild()
{
    child = element.getChildAtIndex(0);
    element.removeChild(child);
    assertEquals(element.getChildrenCount(),
                 old_getChildrenCount - 1);
}
```

(c) Test

Challenges

Challenges

- **Finding relevant specifications**
  Ranking wrt usage, bug-finding capabilities

Challenges

- **Finding relevant specifications**
  Ranking wrt usage, bug-finding capabilities
- **Expressing specifications**
  Choosing a generic, domain-specific vocabulary

Challenges

- **Finding relevant specifications**
  Ranking wrt usage, bug-finding capabilities
- **Expressing specifications**
  Choosing a generic, domain-specific vocabulary
- **Continuous specification**
  Abstract feedback while you program

Challenges

- **Finding relevant specifications**

  Ranking wrt usage, bug-finding capabilities

- **Expressing specifications**

  Choosing a generic, domain-specific vocabulary

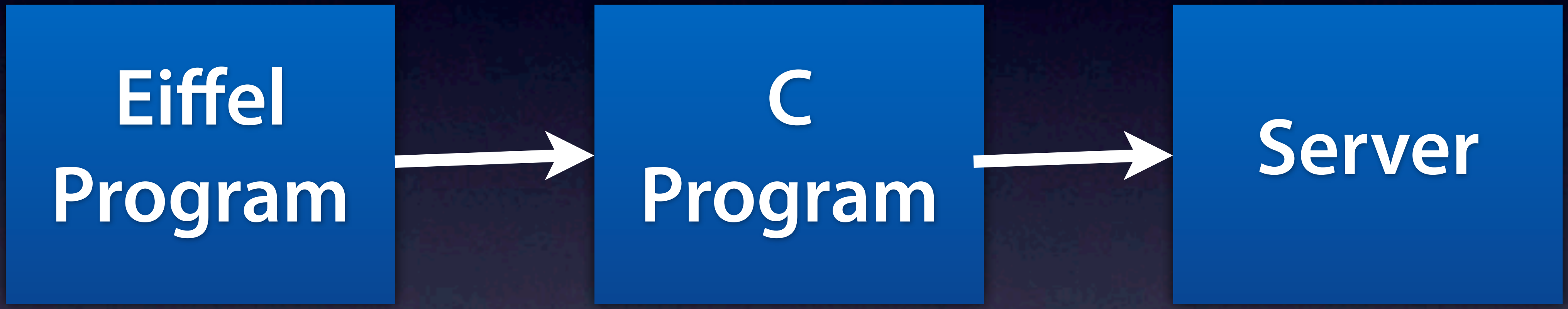
- **Continuous specification**

  Abstract feedback while you program

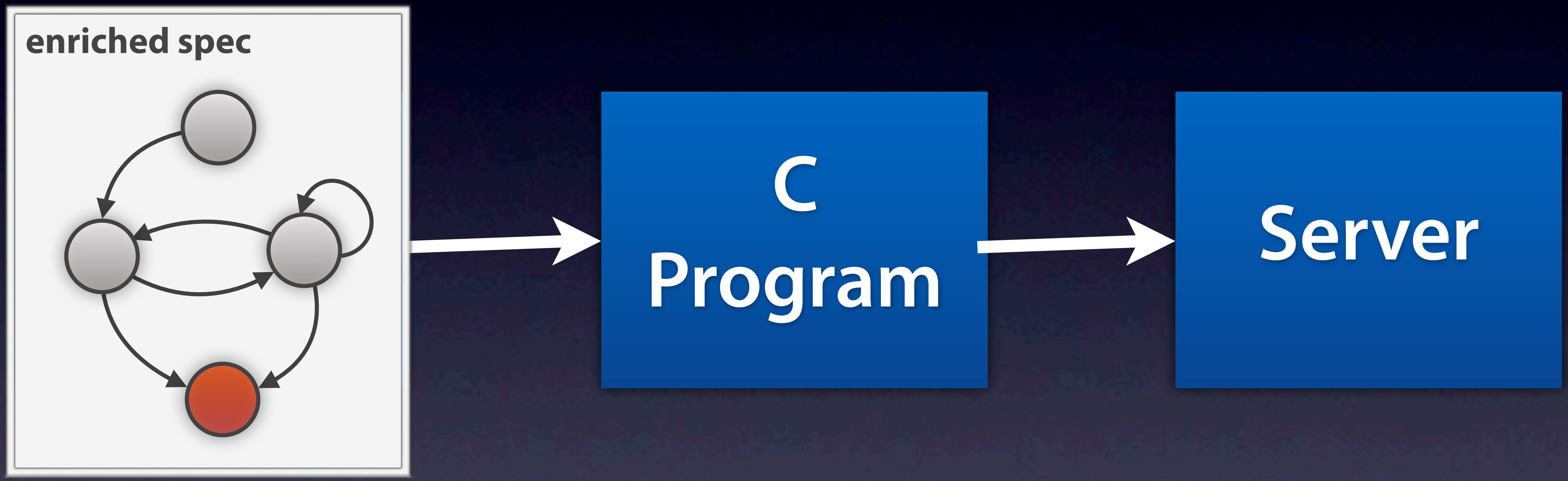
- **Verified specifications**

  Integration with symbolic verification

Compositional Verification



Compositional Verification

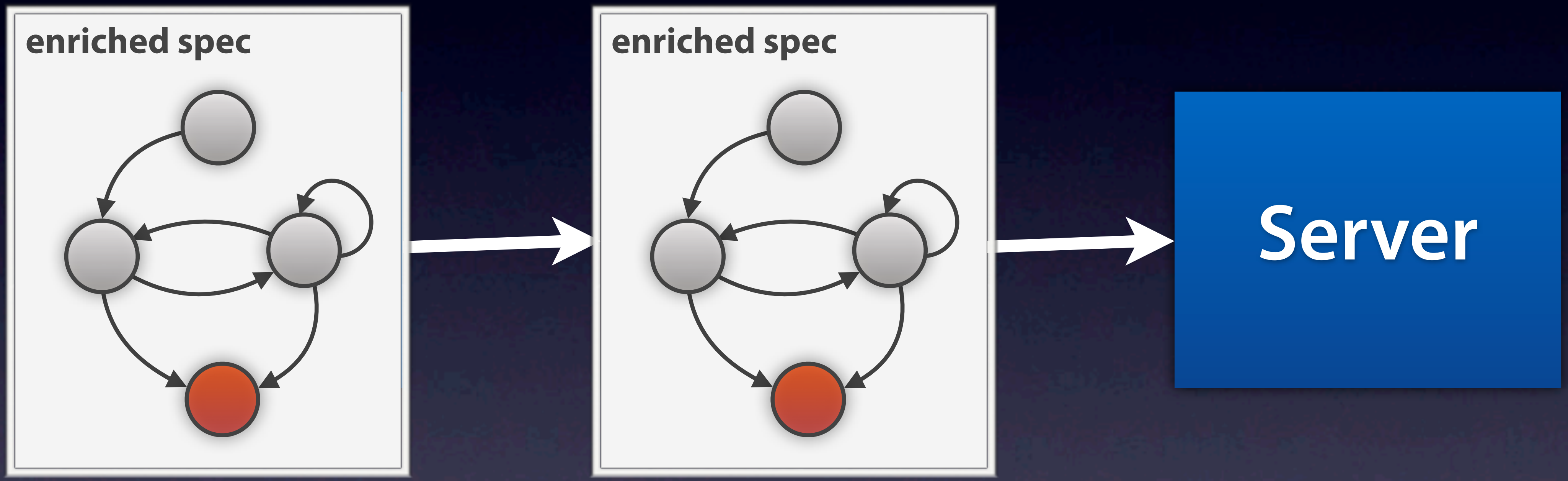


enriched spec

C Program

Server

Compositional Verification

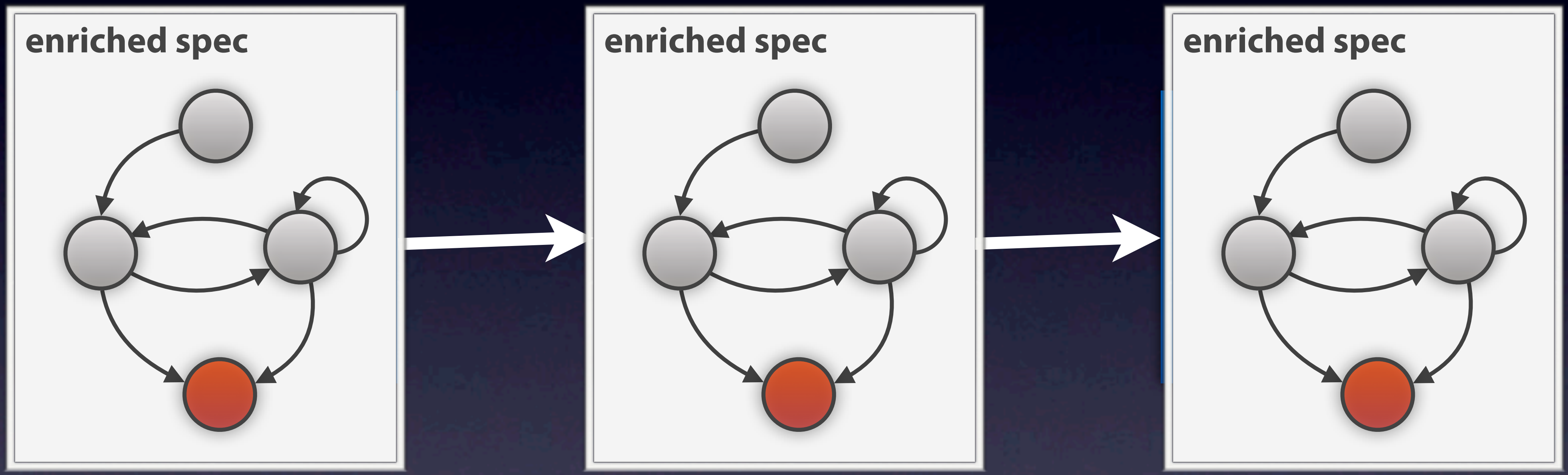


enriched spec

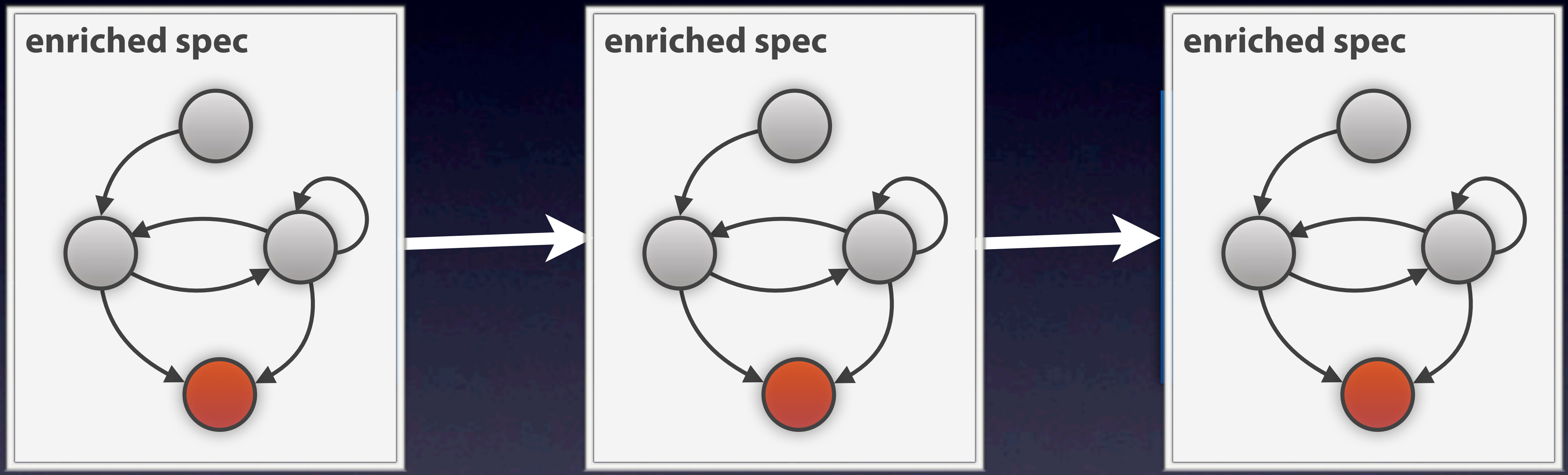
enriched spec

Server

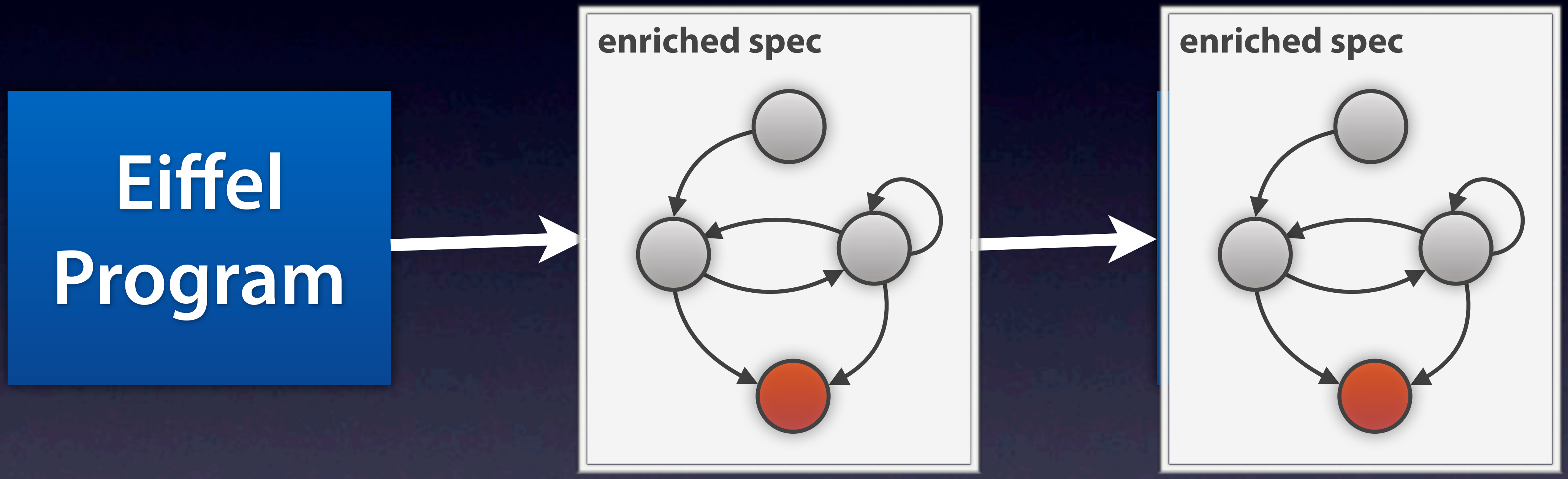
Compositional Verification



Compositional Verification



Compositional Verification

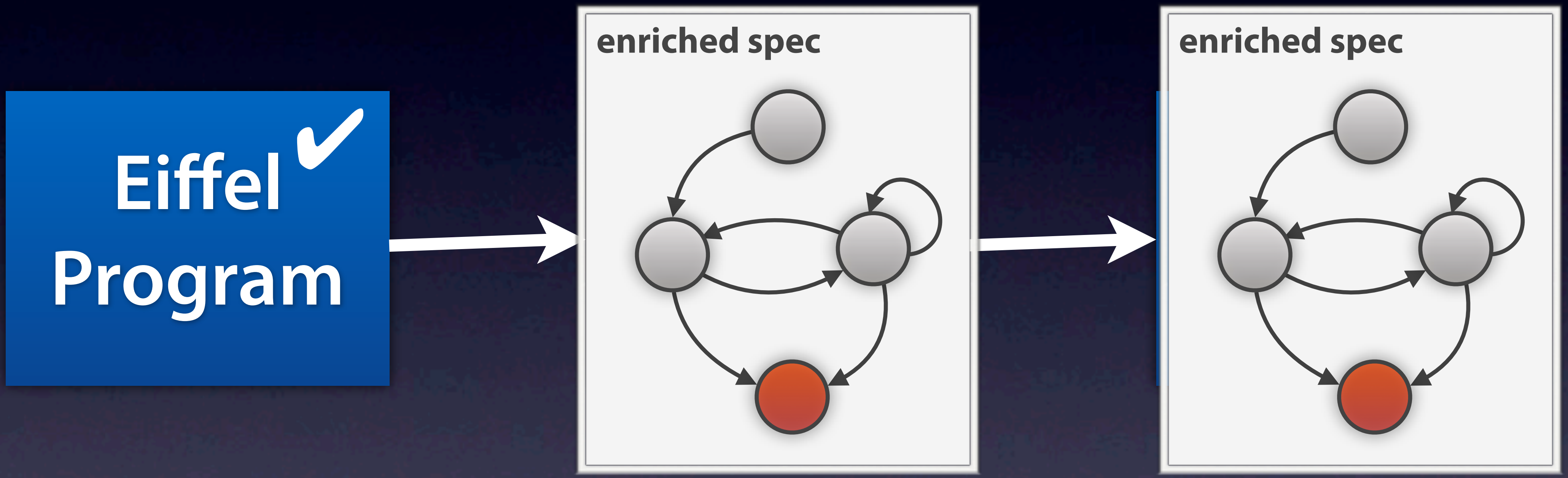


Eiffel Program

enriched spec

enriched spec

Compositional Verification

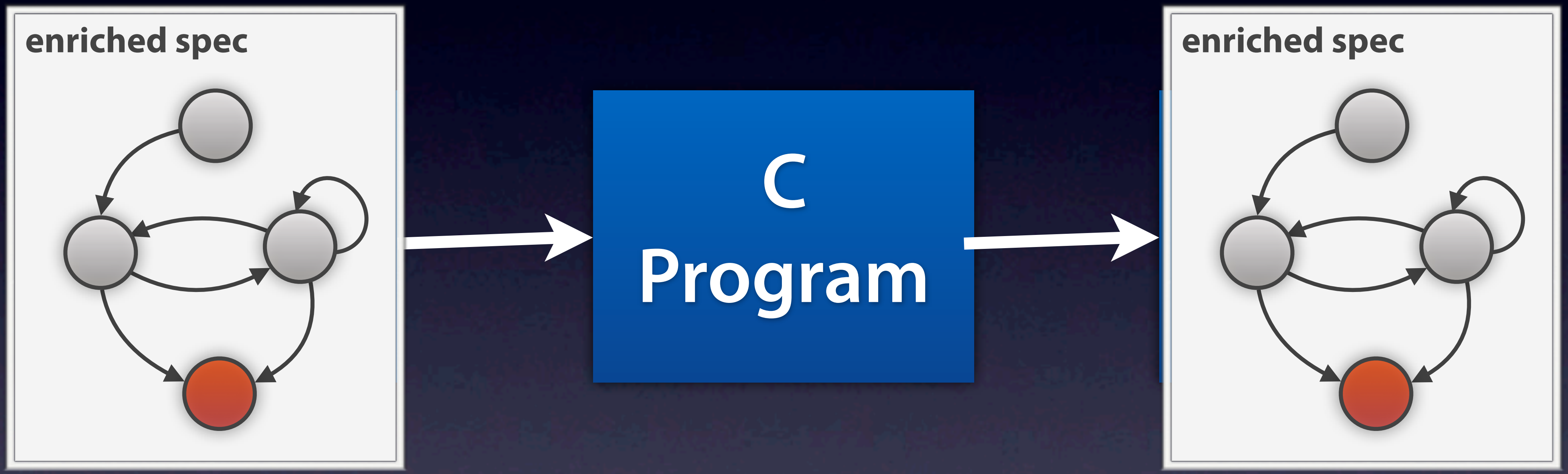


Eiffel ✔
Program

enriched spec

enriched spec

Compositional Verification

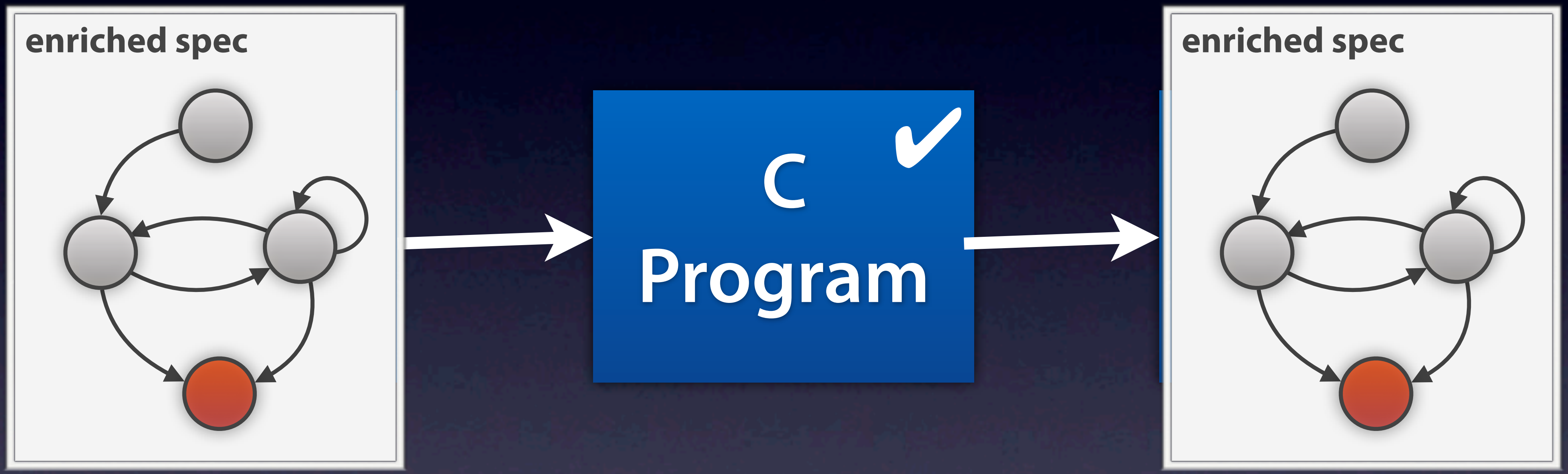


enriched spec

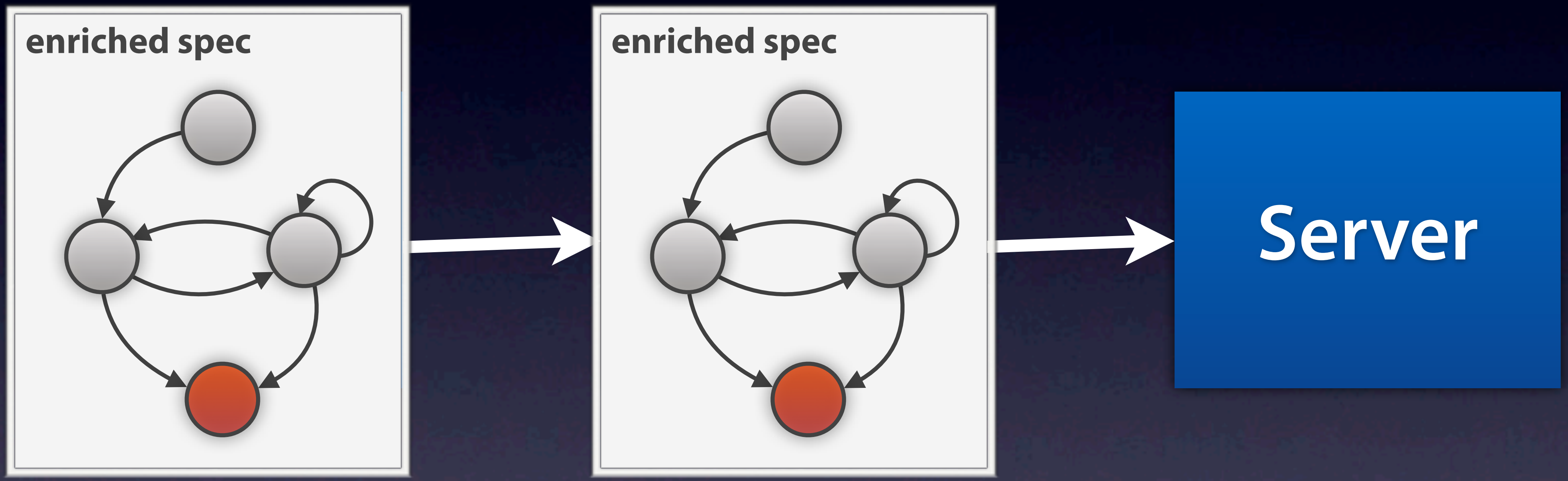
C Program

enriched spec

Compositional Verification



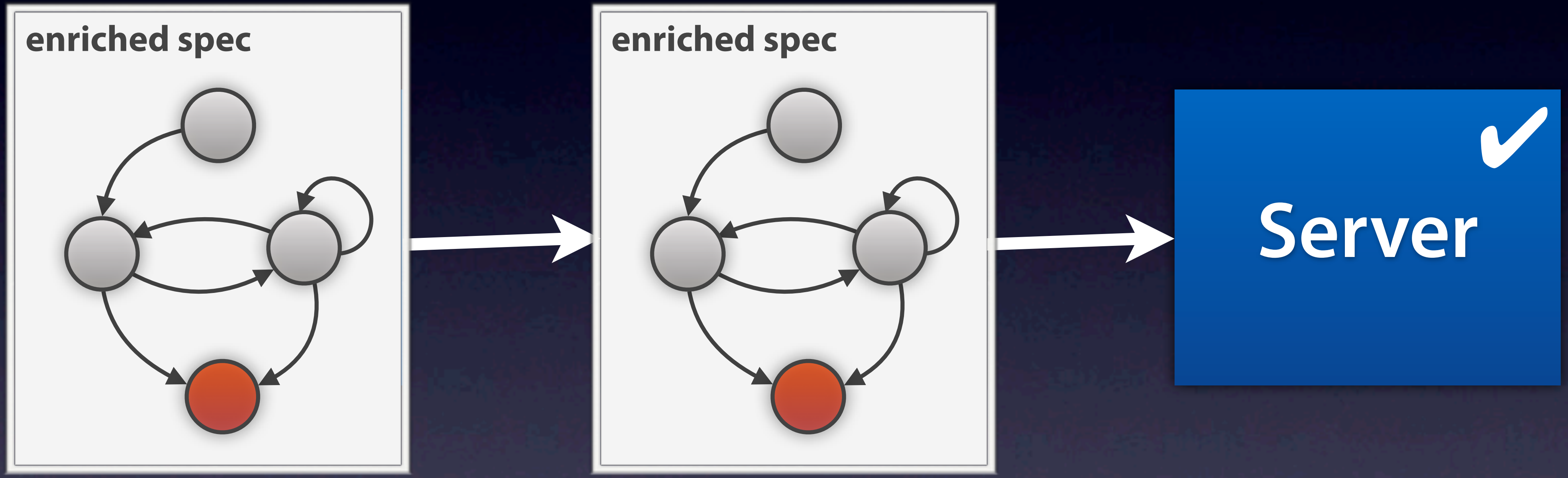
Compositional Verification


enriched spec

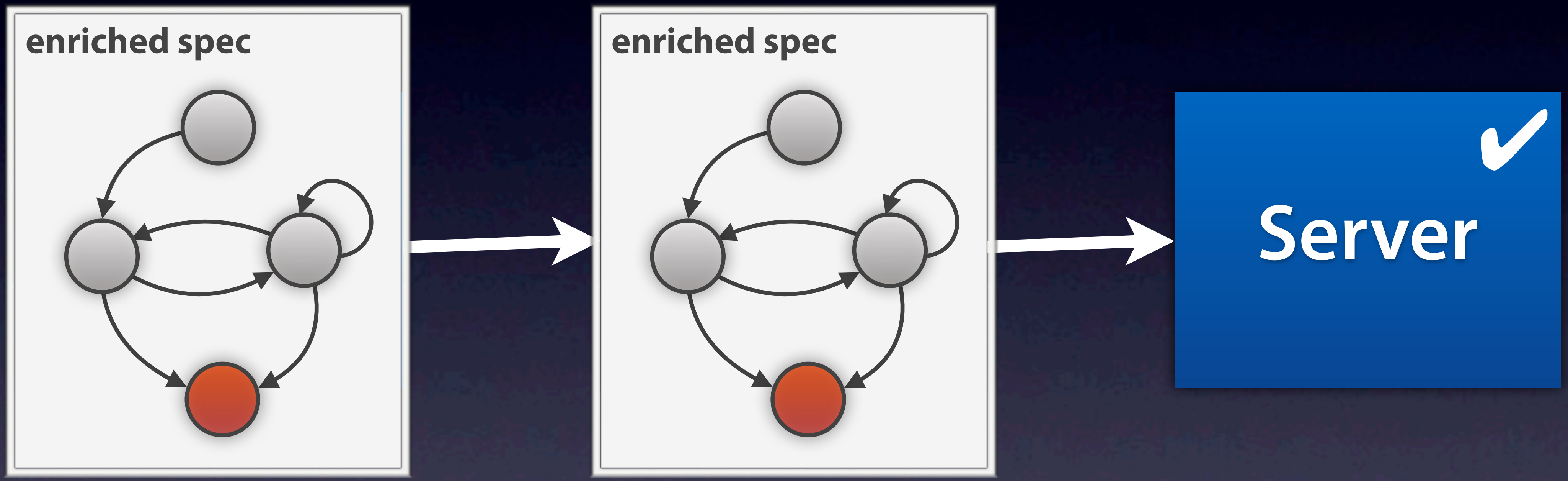
enriched spec

Server

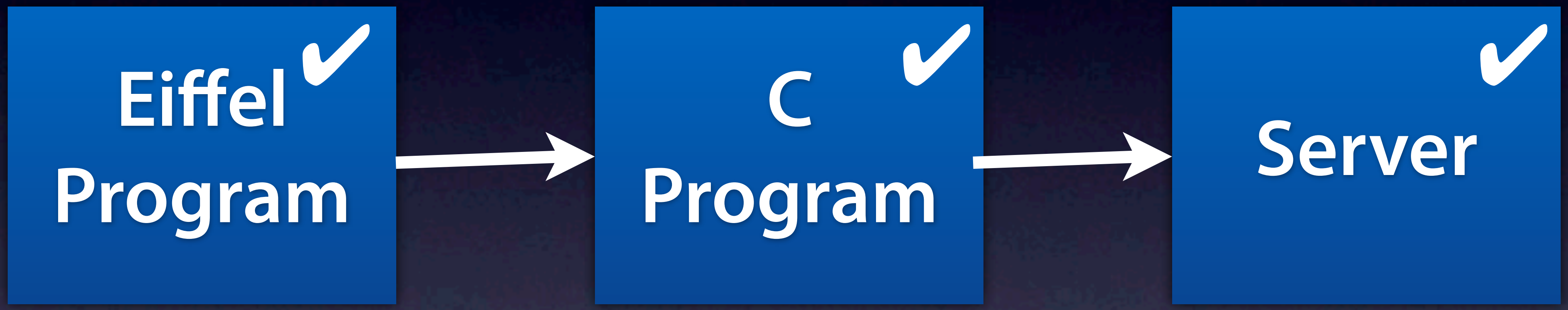
Compositional Verification



Compositional Verification



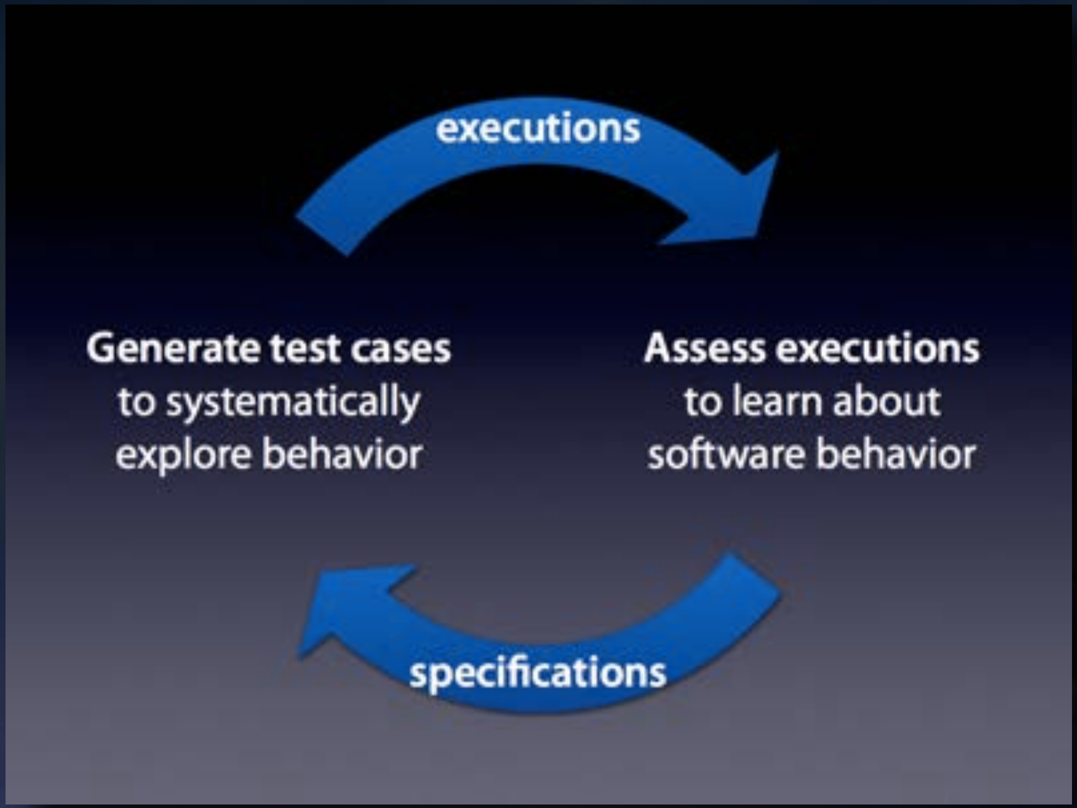
Compositional Verification



Eiffel Program ✓

C Program ✓

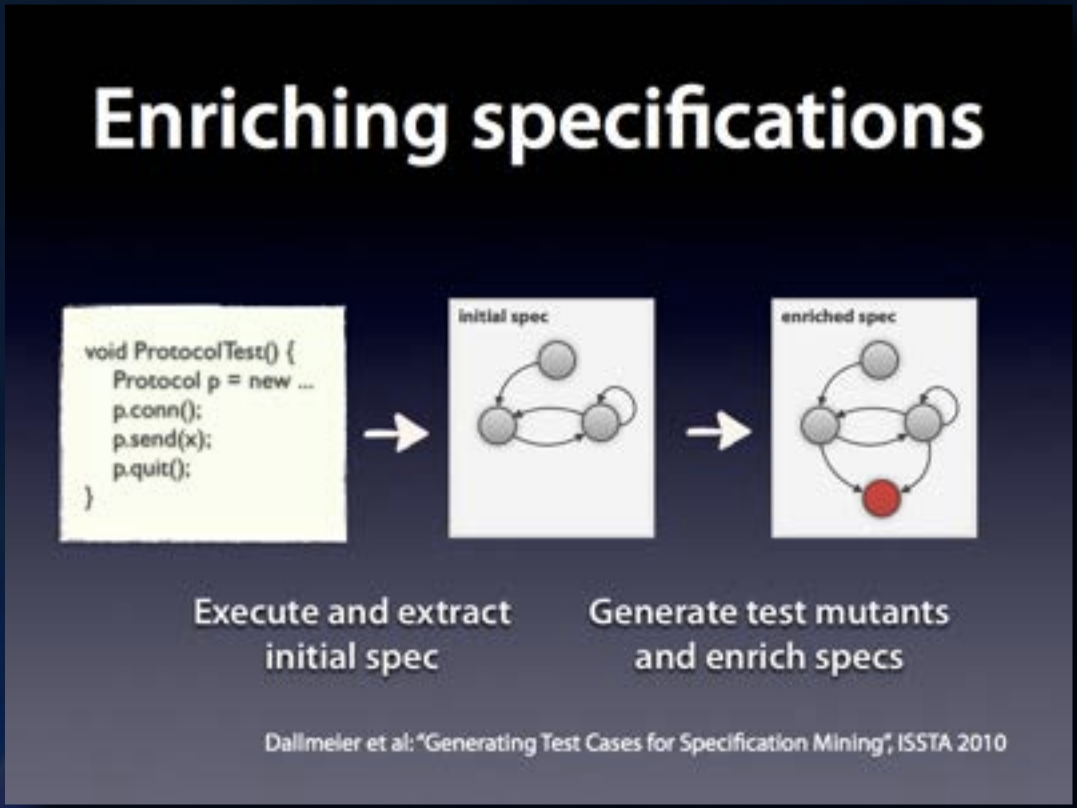
Server ✓



Generate test cases to systematically explore behavior

Assess executions to learn about software behavior

Experimental analysis



Enriching specifications

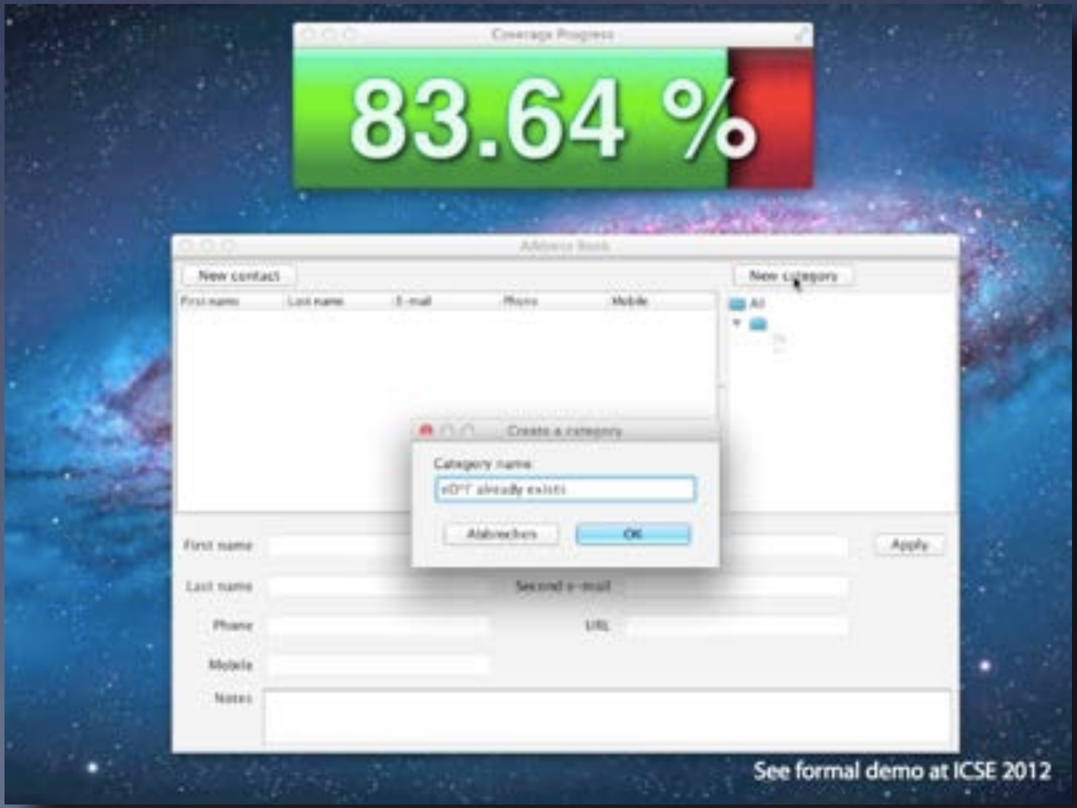
```
void ProtocolTest() {
    Protocol p = new ...
    p.conn();
    p.send(x);
    p.quit();
}
```

Execute and extract initial spec

Generate test mutants and enrich specs

Dallmeier et al: "Generating Test Cases for Specification Mining", ISSTA 2010

Complete behavior

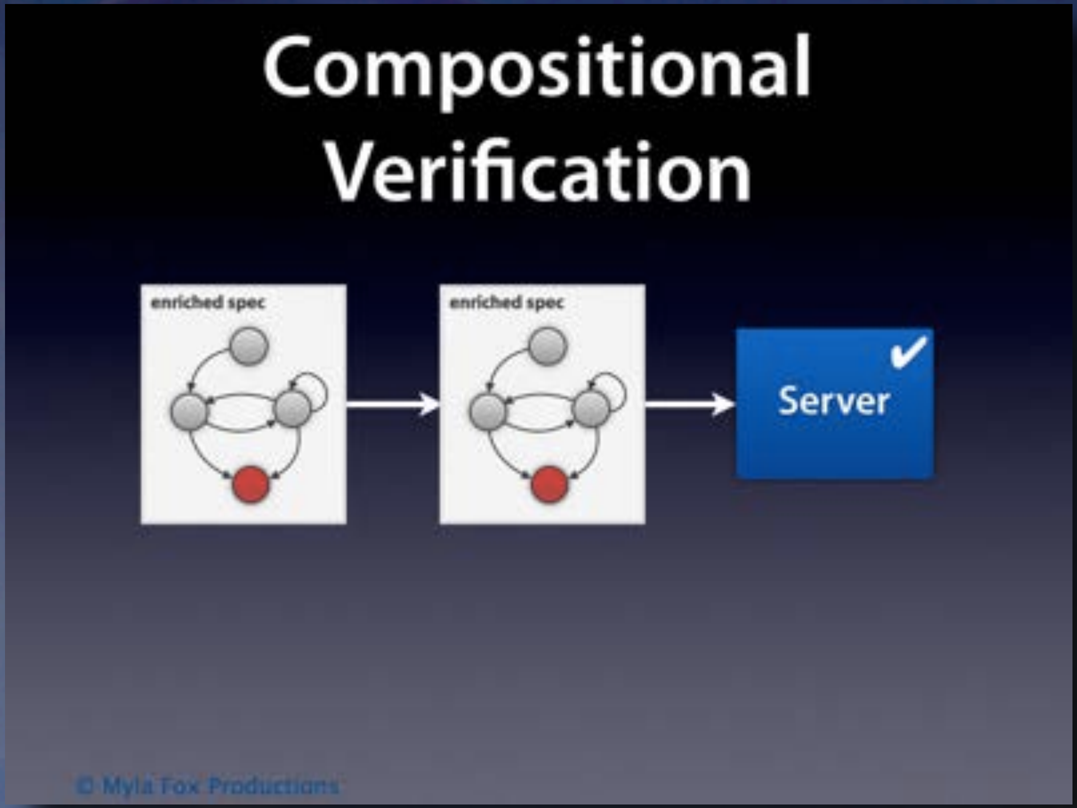


83.64 %

Category name: ɤD*ɪ' already exists

See formal demo at ICSE 2012

Real behavior



Compositional Verification

© Myla Fox Productions

Reliable software