Automating Test Automation

Saurabh Sinha

(Joint work with Suresh Thummala, Nimit Singhania, and Satish Chandra)
What is Test Automation?

1. Launch the application through the link
   http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
What is Test Automation?

1. Launch the application through the link http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as "MySQL" at the “Title” Edit field and select “All” by Drop down list and then Click on the search Button

3. Select a title from the list of all Search results displayed and then click either on the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on the “Add to Shopping Cart” Button

```java
public void setUp() throws Exception {
    driver = new FirefoxDriver();
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
}
```

```java
public void test() throws Exception {
    driver.get("/online-bookstore/Default.jsp");
    driver.findElement(By.name("name")).clear();
    driver.findElement(By.name("name")).sendKeys("mysql");
    driver.findElement(By.cssSelector("input[type=\"submit\"]")).click();
    driver.findElement(By.xpath("//tr[2]/td[2]/a/font")).click();
    driver.findElement(By.name("Login")).clear();
    driver.findElement(By.name("Login")).sendKeys("guest");
    driver.findElement(By.name("Password")).clear();
    driver.findElement(By.name("Password")).sendKeys("guest");
    driver.findElement(By.cssSelector("input[type=\"submit\"]")).click();
    ...
```
Is Test Automation a Significant Problem?

- Real applications can have thousands of manual tests (a product with 30,000 manual tests), requiring regression cycles that can run into a few months

- Efficient and effective regression testing requires automated test execution

- But, test automation comes with costs
  - **Initial cost**: automation is time-consuming and requires specialized skills
  - **Maintenance cost**: small changes in the user interface can break the automated scripts
Test Automation Techniques

- **Record and replay?**
  - Fragile, difficult to maintain scripts

- **Programming in Java or VBScript**
  - Expensive and requires coding expertise

- **Keyword-driven framework?**
  - Effort in developing a general framework

- **High-level scripting language?**
  - Limited expressivity
1. Launch the application through the link http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

1. Verify “User Information” and Following “Item” Details of Selected Book, Details Order #, Item Price, Quantity, Total
A Manual Test Case and Tool-Agnostic Representation

1. Launch the application through the link
   http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

1. Verify “User Information” and Following “Item” Details of Selected Book, Details Order #, Item Price, Quantity, Total

Action-Target-Data (ATD) Tuples

<goto, http://godel.in.ibm.com:8080/online-bookstore/default.jsp, >
A Manual Test Case and Tool-Agnostic Representation

1. Launch the application through the link 
   http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

1. Verify “User Information” and Following “Item” Details of Selected Book, Details Order #, Item Price, Quantity, Total

Action-Target-Data (ATD) Tuples

• <goto, http://godel.in.ibm.com:8080/online-bookstore/default.jsp, >

• <enter, title, MySQL>
• <select, category, all>
• <click, search, >
1. Launch the application through the link http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

1. Verify “User Information” and Following “Item” Details of Selected Book, Details Order #, Item Price, Quantity, Total

Action-Target-Data (ATD) Tuples

- <goto, http://godel.in.ibm.com:8080/online-bookstore/default.jsp, >
- <select, category, all, >
- <click, search, >
- <select, title, >
- <click, login, >
- <enter, quantity, 1, >
- <click, add to shopping cart, >

Data can be read from external files
Conditional tuples can be specified
Challenge: Identifying Relevant Segments

1. Launch the application through the link http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

1. Verify “User Information” and Following “Item” Details of Selected Book, Details Order #, Item Price, Quantity, Total

- **Problem**
  - A manual test step can include multiple segments combined using conjunctions
  - Splitting simply based on conjunctions results in invalid segments
  - E.g.: Step 4 results in an invalid segment “password”, which does not contain any verb

- **Solution**
  - Enumerate all possible candidate segmentations and explore each candidate systematically
  - Backtrack if a chosen candidate does not lead to the interpretation of the entire test case
Challenge: Filtering Irrelevant Segments

1. Launch the application through the link http://godel.in.ibm.com:8080/online-bookstore/Default.jsp

2. Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

3. Select a title from the list of all Search Results displayed and then click either on the image of the book or on Name of the Book

4. Enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

1. Verify “User Information” and Following “Item” Details of Selected Book, Details Order #, Item Price, Quantity, Total

- **Problem**
  - A test step can contain valid ATD tuples that do not correspond to actions on the user interface

- **Solution**
  - Use the application under analysis as a test oracle and ignore irrelevant ATD tuples
  - Proceed automatically with the next ATD tuple of the test step
Challenge: Disambiguating Targets

- **Problem**
  - The target of a test step may match multiple elements in the user interface
  - Reason: testers often do not mention the target properly or the user interface contains multiple elements with the same label

- **Solution**
  - Explore all candidate target elements systematically
  - Backtrack if a chosen candidate prohibits interpretation of the entire test case

Two UI elements labeled “Users”
Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button.
Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

Enter the intended book search name as “MySQL” at the “Title” Edit

Select “Category” as “All”

Drop down list

Then Click “Search” Button

1 <Enter, Title, Book>
2 <Enter, Title, MySQL>
3 <Enter, Title, Name>
4 ...

Segmentation and Enumeration of Alternative Interpretations
Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

Segmentation and Enumeration of Alternative Interpretations

Enter the intended book search name as “MySQL” at the “Title” Edit

select “Category” as “All”

Drop down list

then Click “Search” Button

1 <Select, Category, All>

1 <Enter, Title, Book>
2 <Enter, Title, MySQL>
3 <Enter, Title, Name>
4 ...

© 2012 IBM Corporation
Segmentation and Enumeration of Alternative Interpretations

Enter the intended book search name as “MySQL” at the “Title” Edit field and select “Category” as “All” by Drop down list and then Click “Search” Button

1 2 3 4

ε

Enter the intended book search name as “MySQL” at the “Title” Edit

Select “Category” as “All”

Drop down list

then Click “Search” Button

1 <Enter, Title, Book>
2 <Enter, Title, MySQL>
3 <Enter, Title, Name>
4 ...

1 <Select, Category, All>

1 <Drop, Down, List>
2 <Drop, List, Down>

1 <Click, Search, Button>
2 <Click, Search, _>
3 <Click, Button, Search>
4 <Click, Button, _>
Synthesis of the Tool-Agnostic Representation

Use the application as oracle
1. Can perform ATD on the current page
2. Subsequent steps succeed

If the exploration of a candidate flow fails, backtrack to explore an alternative flow

1 <Enter, Title, Book>
2 <Enter, Title, MySQL>
3 <Enter, Title, Name>
4 ...

1 <Select, Category, All>

1 <Drop, Down, List>
2 <Drop, List, Down>

1 <Click, Search, Button>
2 <Click, Search, _>
3 <Click, Button, Search>
4 <Click, Button, _>
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
   4.1.2 password “guest”
   4.1.3 click login
4.2.1 enter login “guest”
   4.2.2 password “guest”, and click login
4.3.1 enter login “guest” and password “guest”
   4.3.2 click login
4.4.1 enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
      ATDTuple: <enter, login, guest>
   4.1.2 password “guest”
   4.1.3 click login
4.2.1 enter login “guest”
4.2.2 password “guest”, and click login
4.3.1 enter login “guest” and password “guest”
4.3.2 click login
4.4.1 enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
     ATDTuple: <enter, login, guest>
   4.1.2 password “guest”
     ATDTuple: none
   4.1.3 click login
   4.2.1 enter login “guest”
   4.2.2 password “guest”, and click login
   4.3.1 enter login “guest” and password “guest”
   4.3.2 click login
   4.4.1 enter login “guest” and password “guest”, and click login
5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

---

**Backtracking: An Illustrative Example**

- **Root**
  - SL4.1
    - 4.1.1
    - 4.1.2
    - 4.1.3
  - SL4.2
    - 4.2.1
    - 4.2.2
  - SL4.3
    - 4.3.1
    - 4.3.2
  - SL4.4
    - 4.4.1

- No action in Segment 4.1.2
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”  
     ATDTuple: <enter, login, guest>
   4.1.2 password “guest”  
     ATDTuple: none
   4.1.3 click login
   4.2.1 enter login “guest”
   4.2.2 password “guest”, and click login
   4.3.1 enter login “guest” and password “guest”  
   4.3.2 click login
   4.4.1 enter login “guest” and password “guest”  
     and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
4. enter login “guest” and password “guest”, and click login
4.1.1 enter login “guest”
   ATDTuple: <enter, login, guest>
4.1.2 password “guest”
   ATDTuple: none
4.1.3 click login
4.2.1 enter login “guest”
   ATDTuple: <enter, login, guest>
4.2.2 password “guest”, and click login
   ATDTuple: <click, login, >
4.3.1 enter login “guest” and password “guest”
4.3.2 click login
4.4.1 enter login “guest” and password “guest”, and click login
5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button

Login does not succeed because password is not entered
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
       ATDTuple: <enter, login, guest>
   4.1.2 password “guest”
       ATDTuple: none
   4.1.3 click login
   4.2.1 enter login “guest”
       ATDTuple: <enter, login, guest>
   4.2.2 password “guest”, and click login
       ATDTuple: <click, login, >
   4.3.1 enter login “guest” and password “guest”
   4.3.2 click login
   4.4.1 enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
   5.1.1 enter quantity “1”
   5.1.2 click on “Add to shopping cart” button
   5.2.1 enter quantity “1” and click on cart button

Login does not succeed because password is not entered
4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
     ATDTuple: <enter, login, guest>
   4.1.2 password “guest”
     ATDTuple: none
   4.1.3 click login
   4.2.1 enter login “guest”
     ATDTuple: <enter, login, guest>
   4.2.2 password “guest”, and click login
     ATDTuple: <click, login, >
   4.3.1 enter login “guest” and password “guest”
   4.3.2 click login
   4.4.1 enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
   5.1.1 enter quantity “1”
     ATDTuple: <enter, quantity, 1>
   5.1.2 click on “Add to shopping cart” button
   5.2.1 enter quantity “1” and click on cart button

Login does not succeed because password is not entered

Target resolution fails
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
       ATDTuple: <enter, login, guest>
   4.1.2 password “guest”
       ATDTuple: none
   4.1.3 click login
   4.2.1 enter login “guest”
       ATDTuple: <enter, login, guest>
   4.2.2 password “guest”, and click login
       ATDTuple: <click, login, >
   4.3.1 enter login “guest” and password “guest”
   4.3.2 click login
4.4.1 enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
   5.1.1 enter quantity “1”
       ATDTuple: <enter, quantity, 1>
   5.1.2 click on “Add to shopping cart” button
   5.2.1 enter quantity “1” and click on cart button
       ATDTuple: <enter, quantity, 1> <click, cart, >

Target resolution fails

Login does not succeed because password is not entered
Backtracking: An Illustrative Example

4. enter login “guest” and password “guest”, and click login
   4.1.1 enter login “guest”
       ATDTuple: <enter, login, guest>
   4.1.2 password “guest”
       ATDTuple: none
   4.1.3 click login
   4.2.1 enter login “guest”
       ATDTuple: <enter, login, guest>
   4.2.2 password “guest”, and click login
       ATDTuple: <click, login, >
   4.3.1 enter login “guest” and password “guest”
   4.3.2 click login
   4.4.1 enter login “guest” and password “guest”, and click login

5. Enter the Quantity “1” and Click on “Add to Shopping Cart” Button
   5.1.1 enter quantity “1”
       ATDTuple: <enter, quantity, 1>
   5.1.2 click on “Add to shopping cart” button
   5.2.1 enter quantity “1” and click on cart button
       ATDTuple: <enter, quantity, 1>
       <click, cart, >
Technical Issues

- State restoration problem

- Optimizations for intelligently exploring the search space
  - Look-ahead static checking
  - Local backtracking
  - Active Learning

- Guided automation for incomplete manual test cases
  - Addresses the issue of incomplete manual test cases
    - Steps do not have all necessary information
    - Missing test steps
      - Login to the application as guest
  - Highlights the failing test step and seeks human feedback
    - Reuses the feedback in the future to resolve similar steps
ATA Architecture

- Manual Test Case
- Backtracking Engine
- Learning Engine
- Natural Language Processor
- Runtime Interpreter
- User Interface
- NLP Software
- Rational Functional Tester
- Application under test
- Automatically Interpretable ATD Tuples
Empirical Evaluation

- Empirical Studies
  - Three web applications:
    - Two open-source web applications: *BookStore* and *BugTracker*
    - One enterprise application
  - Research questions
    - How often can ATA automatically interpreted manual test steps, with no human intervention?
    - How often does ATA effectively reuse the human feedback received for the steps that it cannot interpret?
    - Do optimizations help improve the efficiency?

- Evaluation in a Production Environment
  - IBM-internal enterprise web application
RQ1: Interpreting Manual Test Steps

- Total number of manual tests: 33 (159 action steps, 134 verification steps)
- Steps that required human intervention:
  - Action steps: 22 / 159 (14%)
  - Verification steps: 31 / 134 (23%)
- Steps with no human intervention:
  - Action steps: 137 / 159 (86%)
  - Verification steps: 103 / 134 (77%)

<table>
<thead>
<tr>
<th>Tests</th>
<th>Test Steps</th>
<th>Successful Steps</th>
<th>Modified Steps</th>
<th>Reused Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Action</td>
<td>Verification</td>
<td>Action</td>
<td>Verification</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BookStore</td>
<td>10</td>
<td>49</td>
<td>57</td>
<td>32 (65%)</td>
</tr>
<tr>
<td>BugTracker</td>
<td>10</td>
<td>48</td>
<td>49</td>
<td>26 (54%)</td>
</tr>
<tr>
<td>App</td>
<td>13</td>
<td>62</td>
<td>28</td>
<td>42 (68%)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>159</td>
<td>134</td>
<td>100 (63%)</td>
</tr>
</tbody>
</table>
Summary and Future Work

- A technique for automating test automation that
  - Uses a combination of natural-language processing, backtracking exploration, runtime interpretation, and learning
  - Uses the application as the oracle to determine the correctness of a potential interpretation

- Overcomes many of the limitations of conventional automation techniques

- Empirical results indicate that the technique can be effective in practice

- Evidence from an evaluation in a production environment indicates substantial productivity improvement

- Further evaluation in production environments
- Persistent state restoration
- Improved change resiliency

- Why do such large manual test suites exist? What features do they cover? Can the tests be reduced without compromising “coverage”?