Automatically Evaluating the Efficiency of Search-Based Test Data Generation (for Relational Database Schemas)

Cody Kinneer

SEKE 2015

July 7, 2015
Random Testing

Easy to implement — and yet not always very effective!
Search-Based Testing
Search-Based Testing

Often much more effective than random testing
Performance of SBST
Performance of SBST

Fitness Function
Performance of SBST

Data Generator

Fitness Function
Performance of SBST

Data Generator

Fitness Function

Restart Rule

Stop Rule

Search Budget
Performance of SBST

- Data Generator
- Stop Rule
- Fitness Function
- Restart Rule

Search Budget
Performance of SBST

- Data Generator
- Fitness Function
- Restart Rule
- Stop Rule
- Search Budget
How do parameter values influence the efficiency of SBST?
Performance of SBST $O()$
Performance of SBST $O(\text{?})$
Performance of SBST

$O(\text{?})$

Analytical
Performance of SBST

$O(\, ? \, )$

×Analytical
Performance of SBST

$O(\cdot)$

×Analytical  Empirical
Performance of SBST

$O(\cdot)$

- Analytical: ✗
- Empirical: ✓
Doubling Experiment
Doubling Experiment

Input
Doubling Experiment

Input

Time = 14.98
Doubling Experiment

Time = 14.98
Doubling Experiment

Time = 14.98
Doubling Experiment

Input
Time = 14.98

Input
Time = 31.45
Doubling Experiment

Input

Time = 14.98

Input

Time = 31.45
Doubling Experiment

Input

Time = 14.98

Input

Time = 31.45

Ratio ≈ 2
Doubling Experiment

Input
Time = 14.98

Input
Time = 31.45

Ratio \approx 2

Linear \quad \text{O}(n)
Doubling Experiment
Doubling Experiment

Time = 12.63
Doubling Experiment

Input

Time = 12.63

Input

Time = 51.48
Doubling Experiment

- Input 1: Time = 12.63
- Input 2: Time = 51.48
- Ratio ≈ 4
Doubling Experiment

Input

Time = 12.63

Input

Time = 51.48

Ratio \approx 4

Quadratic \sim O(n^2)
Doubling Experiment

Input

Input
Doubling Experiment

Time = 11.23
Doubling Experiment

Time = 11.23

Time = 89.72
Doubling Experiment

Input

Time = 11.23

Input

Time = 89.72

Ratio ≈ 8
Doubling Experiment

Input

Time = 11.23

Input

Time = 89.72

Ratio ≈ 8

Cubic — $O(n^3)$
Relational Databases

Deployment Locations for Databases
Relational Databases

Deployment Locations for Databases

Database Application Server

PostgreSQL
Relational Databases

Deployment Locations for Databases

- Database Application Server
- Mobile Phone or Tablet
- Office and Productivity Software

PostgreSQL
SQLite
HyperSQL
Relational Databases

Deployment Locations for Databases

- Database Application Server
- Mobile Phone or Tablet
- Office and Productivity Software

PostgreSQL
SQLite
HyperSQL

Government
Relational Databases

Deployment Locations for Databases

- Database Application Server
- Mobile Phone or Tablet
- Office and Productivity Software

Government

Astrophysics

PostgreSQL

SQLite

HyperSQL
Database Schemas

PostgreSQL

Relational Database Management System
Database Schemas

E-commerce

PostgreSQL

Relational Database Management System
Database Schemas

PostgreSQL

Relational Database Management System

E-commerce

Schema
Database Schemas

E-commerce

PostgreSQL

Relational Database Management System

Schema
State
Database Schemas

E-commerce

Relational Database Management System

PostgreSQL

Schema

State

Integrity Constraints
Database Schemas

PostgreSQL

Relational Database Management System

E-commerce

Schema

State

Integrity Constraints

PRIMARY KEY
Database Schemas

Relational Database Management System

E-commerce

PostgreSQL

Schema

State

Integrity Constraints

- PRIMARY KEY
- FOREIGN KEY
- Arbitrary CHECK
Database Schemas

E-commerce

Relational Components

Schema

State

Relational Database Management System

PostgreSQL
Database Schemas

Relational Database Management System

E-commerce

Relational Components

Tables

Schema

State

PostgreSQL
Database Schemas

PostgreSQL

Relational Database Management System

E-commerce

Relational Components

Schema

Tables

Rows

State
The Data Warehouse Institute reports that North American organizations experience a $611 billion annual loss due to poor data quality.
The Data Warehouse Institute reports that North American organizations experience a $611 billion annual loss due to poor data quality.

Scott W. Ambler argues that the “virtual absence” of database testing — the validation of the contents, schema, and functionality of the database — is the primary cause of this loss.
The Data Warehouse Institute reports that North American organizations experience a $611 billion annual loss due to poor data quality.

Scott W. Ambler argues that the “virtual absence” of database testing — the validation of the contents, schema, and functionality of the database — is the primary cause of this loss.

Past papers presented SchemaAnalyst, a search-based system for testing the complex integrity constraints in relational schemas.
Method of Approach

SchemaAnalyst Execution
Method of Approach

Coverage Criterion

SchemaAnalyst Execution
Method of Approach

Coverage Criterion

Data Generator

SchemaAnalyst Execution
Method of Approach

Coverage Criterion

Data Generator

Database Schema

SchemaAnalyst Execution
Method of Approach

Coverage Criterion

Data Generator

Database Schema

SchemaAnalyst Execution

Test Suite
Method of Approach

Coverage Criterion
Data Generator
Database Schema
SchemaAnalyst Execution
Runtime
Method of Approach

Coverage Criterion

Data Generator

Database Schema

Doubler Choice

SchemaAnalyst Execution

Provides Schema

Schema Doubler

Runtime
Method of Approach

Coverage Criterion

Data Generator

Database Schema

Doubler Choice

SchemaAnalyst Execution

Provides Schema

Schema Doubler

Runtime

Convergence Algorithm

Continue?
Doubling Schemas

<table>
<thead>
<tr>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Column 1</td>
</tr>
<tr>
<td>▶ Column 2</td>
</tr>
<tr>
<td>▶ ...</td>
</tr>
<tr>
<td>▶ Column n</td>
</tr>
</tbody>
</table>
Doubling Schemas

Table

- Column 1
- Column 2
- ...
- Column n
### Doubling Schemas

#### Table

<table>
<thead>
<tr>
<th>Column 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column 2</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Column n</td>
</tr>
</tbody>
</table>

- NOT NULL
- PRIMARY KEY
## Doubling Schemas

A table with columns:

- Column 1
- Column 2
- ...
- Column n

**Properties:**

- **UNIQUE**
- **NOT NULL**
- **PRIMARY KEY**
# Doubling Schemas

<table>
<thead>
<tr>
<th>UNIQUE</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Column 1</td>
<td></td>
</tr>
<tr>
<td>Column 2</td>
<td>. . .</td>
</tr>
<tr>
<td>Column n</td>
<td></td>
</tr>
</tbody>
</table>
Doubling Schemas

### Table
- Column 1
- Column 2
- ...
- Column n

- **UNIQUE**
- **NOT NULL**
- **FOREIGN KEY**
- **PRIMARY KEY**
- **CHECK**
Doubling Schemas

Table

- Column 1
- Column 2
- ...
- Column n

- UNIQUE
- NOT NULL
- FOREIGN KEY
- PRIMARY KEY
- CHECK
Experiments

Experimental Parameters
Experiments

Coverage Criterion

Experimental Parameters
Experiments

- Experimental Parameters
  - Coverage Criterion
  - Data Generator
Experiments

Experimental Parameters

- Coverage Criterion
- Data Generator
- Doubling Technique
- Database Schema
Experiments

Over 2,000 unique combinations of parameters!
Experiments ran on HPC cluster with 3,440 cores

Over 2,000 unique combinations of parameters!
<table>
<thead>
<tr>
<th>Schema</th>
<th>Tables</th>
<th>Columns</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioSQL</td>
<td>28</td>
<td>129</td>
<td>186</td>
</tr>
<tr>
<td>Cloc</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>iTrust</td>
<td>42</td>
<td>309</td>
<td>134</td>
</tr>
<tr>
<td>JWhoisServer</td>
<td>6</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>NistWeather</td>
<td>2</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>NistXTS7</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NistXTS749</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>RiskIt</td>
<td>13</td>
<td>57</td>
<td>36</td>
</tr>
<tr>
<td>UnixUsage</td>
<td>8</td>
<td>32</td>
<td>24</td>
</tr>
</tbody>
</table>
# Relational Schemas

<table>
<thead>
<tr>
<th>Schema</th>
<th>Tables</th>
<th>Columns</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioSQL</td>
<td>28</td>
<td>129</td>
<td>186</td>
</tr>
<tr>
<td>Cloc</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>iTrust</td>
<td>42</td>
<td>309</td>
<td>134</td>
</tr>
<tr>
<td>JWhoisServer</td>
<td>6</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>NistWeather</td>
<td>2</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>NistXTS7</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NistXTS749</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>RiskIt</td>
<td>13</td>
<td>57</td>
<td>36</td>
</tr>
<tr>
<td>UnixUsage</td>
<td>8</td>
<td>32</td>
<td>24</td>
</tr>
</tbody>
</table>
## Relational Schemas

<table>
<thead>
<tr>
<th>Schema</th>
<th>Tables</th>
<th>Columns</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioSQL</td>
<td>28</td>
<td>129</td>
<td>186</td>
</tr>
<tr>
<td>Cloc</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>iTrust</td>
<td>42</td>
<td>309</td>
<td>134</td>
</tr>
<tr>
<td>JWhoisServer</td>
<td>6</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>NistWeather</td>
<td>2</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>NistXTS7</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NistXTS749</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>RiskIt</td>
<td>13</td>
<td>57</td>
<td>36</td>
</tr>
<tr>
<td>UnixUsage</td>
<td>8</td>
<td>32</td>
<td>24</td>
</tr>
</tbody>
</table>
## Relational Schemas

<table>
<thead>
<tr>
<th>Schema</th>
<th>Tables</th>
<th>Columns</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioSQL</td>
<td>28</td>
<td>129</td>
<td>186</td>
</tr>
<tr>
<td>Cloc</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>iTrust</td>
<td>42</td>
<td>309</td>
<td>134</td>
</tr>
<tr>
<td>JWhoisServer</td>
<td>6</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>NistWeather</td>
<td>2</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>NistXTS7</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NistXTS749</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>RiskIt</td>
<td>13</td>
<td>57</td>
<td>36</td>
</tr>
<tr>
<td>UnixUsage</td>
<td>8</td>
<td>32</td>
<td>24</td>
</tr>
</tbody>
</table>
Empirical Results

Doubled

- UNIQUEs
- NOT NULLs
- CHECKs
Empirical Results

Doubled
- UNIQUEs
- NOT NULLs
- CHECKs

699 Experiments
Empirical Results

Doubled
- UNIQUEs
- NOT NULLs
- CHECKs

699 Experiments
8% Stopped
## Empirical Results

<table>
<thead>
<tr>
<th>Doubled</th>
<th>699 Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIQUEs</td>
<td>8% Stopped</td>
</tr>
<tr>
<td>NOT NULLs</td>
<td>20% $O(1)$ or $O(\log)$</td>
</tr>
</tbody>
</table>
Empirical Results

699 Experiments

- UNIQUEs: 8% Stopped
- NOT NULLs: 20% $O(1)$ or $O(\log n)$
- CHECKs: 72% $O(n)$ or $O(n \log n)$
Empirical Results

- Doubled UNIQUEs
- Doubled NOT NULLs
- Doubled CHECKs

- 699 Experiments
- 8% Stopped
- 20% $O(1)$ or $O(\log)$
- 72% $O(n)$ or $O(n \log n)$

SchemaAnalyst $\in O(n)$ for constraints studied
Empirical Results

Doubled

- Tables
Empirical Results

Doubled Tables

467 Experiments
Empirical Results

Doubled

- Tables

467 Experiments

56% Stopped
## Empirical Results

<table>
<thead>
<tr>
<th>Doubled</th>
<th>Tables</th>
<th>467 Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>56% Stopped</td>
</tr>
<tr>
<td></td>
<td></td>
<td>72 O(n^2)</td>
</tr>
</tbody>
</table>
### Empirical Results

<table>
<thead>
<tr>
<th></th>
<th>Doubled Tables</th>
<th>467 Experiments</th>
<th>56% Stopped</th>
<th>72 $O(n^2)$</th>
<th>10 $O(n^3)$</th>
</tr>
</thead>
</table>

- **Doubled Tables**: 467 Experiments
- **56% Stopped**: 72 $O(n^2)$
- **10 $O(n^3)$**
## Empirical Results

<table>
<thead>
<tr>
<th>Doubled Tables</th>
<th>467 Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>56% Stopped</td>
<td>72 $O(n^2)$</td>
</tr>
<tr>
<td>10 $O(n^3)$</td>
<td></td>
</tr>
</tbody>
</table>

$\text{SchemaAnalyst} \in O(n^3)$ or worse for tables
Empirical Results

- Doubled
  - Columns
Empirical Results

Doubled

- Columns

467 Experiments
Empirical Results

- Doubled
  - Columns

- 467 Experiments
- 203 Stopped
### Empirical Results

<table>
<thead>
<tr>
<th>Doubled</th>
<th>467 Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>203 Stopped</td>
</tr>
<tr>
<td></td>
<td>208 $O(n)$ or $O(n \log n)$</td>
</tr>
</tbody>
</table>
Empirical Results

Doubled

- Columns

467 Experiments

- 203 Stopped

- 208 $O(n)$ or $O(n \log n)$

- 28 $O(n^2)$ and 2 $O(n^3)$
Empirical Results

- Doubled
  - Columns

467 Experiments
- 203 Stopped
- 208 $O(n)$ or $O(n \log n)$
- 28 $O(n^2)$ and 2 $O(n^3)$

$\text{SchemaAnalyst} \in O(n^3)$ or worse for columns
Adequacy Criteria
Adequacy Criteria

More effective criteria require additional runtime
Data Generator

More effective generators can also be more efficient
Search-based test data generation is often highly effective, but worst-case time complexity unknown.
Key Contributions

Search-based test data generation is often highly effective, but worst-case time complexity unknown

A technique for automated doubling experiments
Key Contributions

Search-based test data generation is often highly effective, but worst-case time complexity unknown

A technique for automated doubling experiments

Emprical suggestions for worst-case time complexity
Key Contributions

- Search-based test data generation is often highly effective, but worst-case time complexity unknown
- A technique for automated doubling experiments
- Empirical suggestions for worst-case time complexity
- Tradeoffs in search-based test data generation
Key Contributions

- Search-based test data generation is often highly effective, but worst-case time complexity unknown
- A technique for automated doubling experiments
- Empirical suggestions for worst-case time complexity
- Tradeoffs in search-based test data generation
- https://github.com/kinneerc/ExpOse