The right choice matters!
SMT solving substantially improves model-based debugging of spreadsheets.

Simon Außerlechner, Sandra Fruhmann, Wolfgang Wieser, Birgit Hofer, Raphael Spörk, Clemens Mühlbacher, and Franz Wotawa
Fault Localization in Spreadsheets

- What is Spreadsheet fault localization?
- Why important?
  - Spreadsheet users outnumber programmers
  - Basis for decisions, but error prone
  - Hard to debug spreadsheets
- Attempts to use model-based debugging
  Limitations
  - No support for real numbers
  - Solving time
Outline

1. Existing Work
2. SMT Solving
3. Benchmarks
4. Evaluation
5. Future Work
6. Summary
Existing work

- Abreu et al.: “Constraint-based Debugging of Spreadsheets” [ARW12]
  Minion constraint solver

- Jannach and Engler: “Toward model-based debugging of spreadsheet programs” [JE10]
  Choco constraint solver

Limitations

- No Support for Real Numbers
- Solving Time


## Running Example

### Faulty Spreadsheet

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>week 1</td>
<td>week 2</td>
<td>Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
<td>23</td>
<td>31</td>
<td>23</td>
<td>15</td>
<td>$345,00</td>
</tr>
<tr>
<td>3</td>
<td>Jones</td>
<td>35</td>
<td>34</td>
<td>69</td>
<td>17</td>
<td>$1,173,00</td>
</tr>
<tr>
<td>4</td>
<td>Total</td>
<td>58</td>
<td>65</td>
<td>92</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: EUSES Spreadsheet Corpus

### Formula View

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>week 1</td>
<td>week 2</td>
<td>Total</td>
<td>$/h</td>
<td>Gross Pay</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
<td>23</td>
<td>31</td>
<td>=SUM(B2)</td>
<td>15</td>
<td>=D2*E2</td>
</tr>
<tr>
<td>3</td>
<td>Jones</td>
<td>35</td>
<td>34</td>
<td>=SUM(B3:C3)</td>
<td>17</td>
<td>=D3*E3</td>
</tr>
<tr>
<td>4</td>
<td>Total</td>
<td>=SUM(B2:B3)</td>
<td>=SUM(C2:C3)</td>
<td>=SUM(D2:D3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: EUSES Spreadsheet Corpus
Test Cases for Spreadsheets

- **Input cells**: cells that do not reference other cells
  \[ I = \{B2=23, C2=31, E2=15, B3=35, C3=34, E3=17\} \]

- **Output cells**: any formula cell, determined by user
  \[ O = \{B4=58, C4=65, D4=123, F2=810, F3=1173\} \]
Constraint-based Debugging (ConBug)

Spreadsheet → Conversion → Constraints → Solve → Diagnoses

- Test case: D4 == 123
- For single faults: SUM(f_D2, f_D3, f_D4) == 1

For complete example:
{D2}, {D4,F2}, ...

Satisfiability modulo theories (SMT) solver

- Based on Boolean satisfiability
- Enhanced by different theories (e.g. Integer, Real)
- Solves decision problem
  - Determine diagnoses via MCSes and MCSes-U algorithms [LS08, LS09]
- Z3 solver
  - Speedup in computation time
  - Handling of real numbers


Spreadsheet Corpora (1)

- Limitations of EUSES spreadsheet corpus [FR05]
  - Many small spreadsheets
  - Missing input values
  - Obsolete file formats
  - No faulty versions

### Spreadsheet Corpora (2)

<table>
<thead>
<tr>
<th></th>
<th>Mod. EUSES corpus</th>
<th>Integer corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong></td>
<td>Filtering from EUSES</td>
<td>Artificially created + existing spreadsheets</td>
</tr>
<tr>
<td><strong>Domain</strong></td>
<td>Reals</td>
<td>Integer</td>
</tr>
<tr>
<td><strong>Mutants</strong></td>
<td>622 mutants</td>
<td>220 mutants</td>
</tr>
<tr>
<td></td>
<td><em>created with spreadsheet mutation operators [AE09]</em></td>
<td></td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>6 to 4,000 formulas</td>
<td>7 to 233 formulas</td>
</tr>
<tr>
<td></td>
<td>AVG: 225 formulas</td>
<td>AVG: 39 formulas</td>
</tr>
</tbody>
</table>

Original spreadsheets + mutants publicly available
Empirical Evaluation

- Runtime comparison
  - Minion vs. Choco vs. Z3 (MCSes and MCSes-U)
  - Integer corpus
  - Integers vs. Real numbers
    - Integer corpus + Real number version

- Reduction quality
  - Modified EUSES corpus
Runtime comparison [in ms]: Choco vs. Minion
Runtime comparison [in ms]: Z3 vs. Choco
Detailed runtime comparison

- Choco slightly faster than Minion

- Z3 faster than Minion and Choco
  - Approx. 6 times faster for Integer corpus

- Integer vs. Real domain
  - Integer spreadsheets approx. 2.6 times faster than Reals
Reduction quality

- Reduction $= 1 - \frac{\text{Cells in diagnoses}}{\text{Formula cells}} \times 100\%$

- Evaluation basis
  - Subset of mod. EUSES corpus
  - 183 of 622 spreadsheets

- Results
  - 12 spreadsheets: timeout 5 minutes
  - $\sim 80\%$ reduction
  - $\sim 500$ ms avg. solving time (Z3)
Future Work and Open Challenges

- Scalability problem (Hypertree complexity)
- User interface and user acceptance study
- Explanations for the user (traceability)
- Dealing with vague user input
Using Z3 as solver …

… makes it possible to use model-based debugging for spreadsheets

- In practice (Real numbers) and
- For interactive debugging (Solving time)