Institute for Software Technology

Graz University of Technology

Comparing Models for Spreadsheet Fault Localization

Birgit Hofer and Franz Wotawa

Abstract

We present a novel dependency-based model that can be used in **Model-Based Software Debugging (MBSD)**. This model allows **improvements** of the **diagnostic accuracy** while keeping the **computation times short**. In an empirical evaluation, we show that **dependency-based models** of spreadsheets whose **value-based models** are often not solvable in an acceptable amount of time can be solved in less than one second. Furthermore, the amount of diagnoses is reduced by 15 % on average when using the novel dependency-based model instead of the original dependency-based model.

1 Model-based Debugging

Spreadsheet + Expect. Output → Conversion → Constraints (Model) → Solve → Diagnoses

	A	B	C	D
1	**Item**	**P1**	**P2**	**Total**
2	**Units**	100	300	400
3	**Price**	20	40	35
4	**Income**	2 000	12 000	14 000
5	**Costs**	500	600	500
6	**Profit**	1 500	11 400	13 500

Should be 12 900

	B	C	D
1	**P1**	**P2**	**Total**
2	100	300	=SUM(B2:C2)
3	20	40	=D4/D2
4	=B3*B2	=C3*C2	=SUM(B4:C4)
5	500	600	=SUM(B5:B5)
6	=B4-B5	=C4-C5	=D4-D5

AB(D2) ∨ *behavior(D2)*
AB(D3) ∨ *behavior(D3)*
AB(B4) ∨ *behavior(B4)*
…

Test case:
D6 == 12 900
B6 == 1 500
…

For single faults:
SUM(AB(D2),AB(D3),…)==1

Single Faults:
{D5}
{D6}

Double Faults:
{D3,D4}
…

2 Existing Behavior Models

AB(cell) ∨ *behavior(cell)*

Value-Based Model [1,2,3]
- \+ exact, few diagnoses
- – computation time
- – Reals: lacking support

- D2 = B2 + C2
- D3 = D4 / D2
- B4 = B3 * B2 …

Solve → 2 Diagnoses

Dependency-Based Model [4]
- \+ fast
- \+ only Boolean
- – many diagnoses

- ok(B2) ∧ ok(C2) → ok(D2)
- ok(D4) ∧ ok(D2) → ok(D3)
- ok(B3) ∧ ok(B2) → ok(B4) …

Solve → 5 Diagnoses

3 Improving the Dependency-Based Model

Use ↔ instead of →

- ok(B2) ∧ ok(C2) ↔ ok(D2)
- ok(D4) ∧ ok(D2) ↔ ok(D3)
- ok(B3) ∧ ok(B2) ↔ ok(B4) …

Solve → 2 Diagnoses

4 Coincidental Correctness (Fault Masking)

Functions and operations in spreadsheets where faults could be masked:

- Conditional like IF-function
- Abstraction function like MIN, MAX, COUNT
- Boolean
- Multiplication by zero
- Power with 0 or 1 as base number or 0 as exponent

Formulas containing any of these functions and/or operations must be modeled with →, all other formulas can be modeled with ↔.

5 Empirical Evaluation

Model	63 spreadsheets	31 spreadsheets
Number of single fault diagnoses		
Value-Based	4.0	-
Dependency-Based	13.2	45.0
Improved Dep.-Based	11.0	38.6
Runtime		
Value-Based	56,818.8 ms	> 20 minutes
Dependency-Based	32.0 ms	187.4 ms
Improved Dep.-Based	31.6 ms	164.8 ms

6 Conclusions

Improved Dependency-Based Model:
- − Still more diagnoses than value-based model
- \+ 15 % less diagnoses than original dependency-based model
- \+ Real time applicable
- \+ Arbitrary solver (only Boolean needed)
- \+ Debugging of spreadsheets containing Real numbers
- \+ Correct/wrong info for output cells instead of concrete values

References

[1] R. Abreu, B. Hofer, A. Perez, and F. Wotawa: “Using constraints to diagnose faulty spreadsheets.” *Software Quality Journal, pp. 1–26,* 2014.
[2] D. Jannach and T. Schmitz, “Model-based diagnosis of spreadsheet programs - A constraint-based debugging approach.” *Automated Software Engineering, Springer, pp. 1-40, 2014.*
[3] S. Ausserlechner et al.: “The Right Choice Matters! SMT Solving Substantially Improves Model-Based Debugging of Spreadsheets.” QSIC 2013: 139-148
[4] F. Wotawa, “On the Relationship between Model-Based Debugging and Program Slicing.” Artificial Intelligence, 135, 125–143, 2002.

Acknowledgments

The research herein is partially conducted within the competence network Softnet Austria II (www.soft-net.at, COMET K-Projekt) and funded by the Austrian Federal Ministry of Economy, Family and Youth (bmwfj), the province of Styria, the Steirische Wirtschaftsförderungsgesellschaft mbH. (SFG), and the city of Vienna in terms of the center for innovation and technology (ZIT).

Contact: bhofer@ist.tugraz.at, wotawa@ist.tugraz.at

ECAI – 21st European Conference on Artificial Intelligence
August, 18 - 22, 2014 – Prague, Czech Republic