Tool-Supported Fault Localization in Spreadsheets: Limitations of Current Evaluation Practice

Birgit Hofer, Franz Wotawa
Dietmar Jannach, Thomas Schmitz
Kostyantyn Shchekotykhin
An Overview of Limitations of Current Evaluation Practice

1. Lack of benchmarks systems
2. Usability and user acceptance
3. Field research

Focus: Approaches for automated fault localization
Benchmark Systems – Current Situation

- There is no public data set for spreadsheet fault localization

- Researcher create own benchmark systems
  - Take existing corpus (e.g. EUSES [FR05]) or collect individual spreadsheets
  - Apply mutation operators, e.g. [AE09] on them or manually inject faults


Some Examples I

- Hofer et. al [HRW13]
  - “… we are evaluating the … approaches by means of the EUSES spreadsheet corpus. We skipped around 240 Excel 5.0 spreadsheets that are not compatible with our implementation, …
  - we removed all spreadsheets containing less than 5 formulas …
  - automatically created up to five first-order mutants. A mutant of a spreadsheet is created by randomly choosing a formula cell of the spreadsheet and applying a mutation operator on it. According to the classification of spreadsheet mutation operators of Abraham and Erwig, we used the following mutation operators …”

- Jannach and Schmitz [JS14]
  - “For the performance analysis, we selected a number of artificial and real-world spreadsheets in which we manually injected faults.”


Some examples II

- Abraham and Erwig [AE08]
  - “... we use **spreadsheets** that have been **used in previous empirical studies**. The spreadsheets have been **picked** to include as many different kinds of formulas, and formulas with branching ...”
  - We generate mutant spreadsheets by **seeding faults** in the original spreadsheets using the **mutation operators** given in Table 1. The mutation operators have been **designed to reflect errors** reported in spreadsheet literature ...”

- Außerlechner et al. [AFW13]
  - “Since MINION is not able to deal with Real numbers ..., we **created a specific spreadsheet corpus** that contains spreadsheets with Integer values only ... Whereas some of the spreadsheets are **artificially created**, 21 spreadsheets are **real-life** programs ... “


Current Situation - Consequences

- Each research group uses own data set
  - rarely made publicly available
  - often made to fit the evaluated approach
  - comparison of approaches difficult
We need a corpus that contains …

- Real world spreadsheets
- Large spreadsheets, not toy examples
- Spreadsheets with real faults, not only seeded faults
- Input-/output relations that reveal the fault
Ways to get there

- Laboratory: spreadsheet construction exercises
  - Excellent starting point: Kooper Corpus [AP10]
    - Larger spreadsheets
    - Different domains and exercises
- Real life

Usability and User Acceptance

- Mostly offline experiments
- Information from the user required, e.g.
  - Correctness of values
  - Expected values
  - Specification of several test cases

→ Is a user willing / able to provide these inputs?

→ User studies are necessary to answer these questions.
Field research

- **Setting**
  - Laboratory experiments vs. everyday use

- **Participant**
  - Students vs. managers

- **Scenario**
  - Artificial problem vs. real problem
Proposals for future work

→ Improve comparability and reproducibility
   Develop common benchmark system

→ Focus on usability and user acceptance
   Make user studies

→ Focus on real life scenarios (not only laboratory experiments)
   Make field research, questionnaires ...