Towards Automated Analysis of R Programs

Anton Xue[1], Ross Mawhorter[2], Gian Pietro Farina[3], Stephen Chong[4]
[1]Yale University, [2]Harvey Mudd College, [3]University at Buffalo, [4]Harvard University

Motivation and Background

- The R statistical and graphical language is **widely used** by data scientists
  - 200k+ public GitHub repositories (August 2018)
- R supports many potentially **hazardous features**
  - Lazy side-effecting arguments, first-order environments, dynamic typing
- Many R users are not computer scientists, **may not understand nuances**
- R **lacks sufficient formalization** and has **little formal methods supports**
  - Prior work: Morandat et al., *Evaluating the Design of the R Language* [ECOOP12]

A Statistical Language?

The New York Times

Opinion

The Excel Depression

By Paul Krugman

April 18, 2013

470

In this age of information, math errors can lead to disaster. NASA's Mars Orbiter crashed because engineers forgot to convert to metric measurements; JPMorgan Chase's "London Whale" venture went bad in part because modelers divided by a sum instead of an average. So, did an Excel coding error destroy the economies of the Western world?

Market Watch

Close to 90% of spreadsheet documents contain errors, a 2008 analysis of multiple studies suggests. "Spreadsheets, even after careful development, contain errors in 1% or more of all formula cells," writes Ray Panko, a professor of IT management at the University of Hawaii and an authority on bad spreadsheet practices. "In large spreadsheets with thousands of formulas, there will be dozens of undetected errors."

Forbes

JPMorgan Chase JPM +1.75% lost more than $6 billion in its "London Whale" incident, in part due to Excel spreadsheet errors (including alleged copying and pasting of incorrect information from multiple spreadsheets). In a sad twist of fate, the British bank Barclays sent an offer to purchase another firm in 2008 that hid—instead of deleted—nearly 200 spreadsheet cells, resulting in unnecessary losses.

Our Contributions

- Formalization of a subset of R, called Simple-R
  - Language syntax
  - Execution semantics
- Evaluation to see how well Simple-R against real-world R programs
- Development towards analysis tools for R
  - Preliminary symbolic execution engine
  - Future plans: gradual typing system