Spreadsheet error detection with a type system

1. Motivation

- Panko, R. (2016). What We Don't Know About Spreadsheet Errors Today: The Facts, Why We Don't Believe Them, and What We Need to Do. arXiv preprint arXiv:1602.02601.

1.1 Popular spreadsheet errors

- logical errors (45%): mathematical and domain knowledge
- mechanical errors (23%): pointing to wrong cell and typing errors
- omission errors (31%): leaving out a necessary component

1.2 Facts

- 14 laboratories spreadsheet development studies involving 967 individuals working alone on a variety of tasks. These numbers come from the Panko spreadsheet research website. The average across these studies is a cell error rate of 3.9%
- the probability of an error increases rapidly when there are many calculations that depend on precedent cells

2. Literature

- See bibtex file
- To be summarized (For each one, present the motivation, research question, method, evaluation and key contribution)

3. Method description

- Problem Description: Can we detect the mechanical errors, omission errors and part of logical errors in spreadsheet using static analysis, like type system?

3.1 Definition

3.1.1 Syntax of spreadsheets

- `Cell expressions`: e ::= \blank | \error | v | a | ωn(e1, . . . , en)
- `Spreadsheet`: s :: (a1, e1) ; . . . ; (am, em) i != j => ai != aj
- note: e is expression, \blank is blank cell, \error is error cell, v is value, a is address, ω is operations, n is the number of parameters required by the operation.

3.1.2 Basic Types

- `Basic Types`: Dependent Type, AND Type, and OR Type.
- `Dependent Type`: Cells with value v and header t are in type t[v], e.g. type for cell B3 is week[week1]. It can be recursively defined. e.g. type for cell B7 is week[week1[28]]
- `AND Type`: Cells with one more types are in type t1[v]&t2[v], e.g. type for cell B7 is week[week1[28]]&people[Johnson[28]].
- `OR Type`: Cells with operations referred to other n cells will have a type t1[v]|t2[v]...|tn[v], e.g. type for cell B12 is week[week1[23]]&people[Green[23]] | week[week1[35]]&people[Jones[35]] | ... | week[week1[37]]&people[Edwards[37]].

3.2 Type Check

3.2.1 Judgements

1. Every value that does not have a header is a well-typed.
2. If a cell has value v and header t, then it t[v] is well-typed.
3. For AND Type, if there is no common ancestor, it is well-typed.
4. For OR Type, if there is a common header ancestor, it is well-typed.

3.2.2 Type Inference

4. Simple Example

	A	B	C	D	E
1	Hours Worked in December				
2		week			
3	people	week 1	week 2	week 3	week 4
4	Green	23	31	25	=AVERAGE(B4
5	Jones	35	34	33	=AVERAGE(B5
6	Smith	25	26	27	=AVERAGE(B6
7	Johnson	28	30	21	=AVERAGE(B7
8	White	45	10	14	=AVERAGE(B8
9	Edwards	37	38	40	=AVERAGE(B9
10					
11					
12	Weekly Total	=SUM(B4:B10)	=SUM(C4:C10)	=SUM(D4:D9)	=SUM(E4:E9)
13	Max Hours	=MAX(B4:B9)	=MAX(C4:C9)	=MAX(D4:D9)	=MAX(E4:E9)

- By spatial analysis and operational semantics of each formula, we can know the "type" for each cell.
- For cell B12, by spatial analysis, we know that the type is WeeklyTotal[193] & week[week1[193]]. Since there is no common ancestor of WeeklyTotal[193] and week[week1[193]], with Judgement 3, we know it is well-typed.
- For cell B12, by analyzing the operational semantics, we know that the type is week[week1[23]]&people[Green[23]] | week[week1[35]]&people[Jones[35]] | ... | week[week1[37]]&people[Edwards[37]]. Since there is common anscestor week, thus with Judgement 4, we know it is well-typed.
- If there is a cell with formula "A5 + C4", via analysis, we know it is typed as people[Jones]|week[week2[31]]. Since there is no common ancestor, it is ill-typed and it will be marked as a potential wrong cell.

6. Expectation of Project

6.1 Implementation

6.1.1 Parser

- http://ewbi.blogs.com/develops/popular/excelformulaparsing.html

6.1.2 Type System

- `Basic`: Abraham, R., & Erwig, M. (2007). UCheck: A spreadsheet type checker for end users. Journal of Visual Languages & Computing, 18(1), 71-95.
- `Advanced`: Cunha, J., Fernandes, J. P., Mendes, J., & Saraiva, J. (2015). Embedding, evolution, and validation of model-driven spreadsheets. Software Engineering, IEEE Transactions on, 41(3), 241-263..

6.1.3 Spatial Analysis

- `Basic`: Abraham, R., & Erwig, M. (2004, September). Header and unit inference for spreadsheets through spatial analyses. In Visual Languages and Human Centric Computing, 2004 IEEE Symposium on (pp. 165-172). IEEE.
- `Advanced`: Chambers, C., & Erwig, M. (2009). Combining spatial and semantic label analysis.

6.1.4 Evaluation

- `Spreadsheet Corpus`: http://openscience.us/repo/spreadsheet/euses.html

6.2 Report

6.2.1 Literature Review

- A survey: See bibtex file
- Possible improvement: Extend typing rules with operational semantics of more spreadsheet functions; Automatic error fixing (creating a fixing language (DSL) and do syntactical heuristic search).)

6.2.2 Report on implementation and evaluation

- A short paper

7. Plan

- Finish the implementation of basic type system and basic spatial analysis by the end of March;
- Extend the system in proposed direction till the mid of April;
- Report drafting in the left two weeks.