



UCheck: A spreadsheet type checker for end users[☆]

Robin Abraham, Martin Erwig*

School of EECS, Oregon State University, USA

Received 4 February 2005; received in revised form 12 June 2006; accepted 20 June 2006

Abstract

Spreadsheets are widely used, and studies have shown that most end-user spreadsheets contain non-trivial errors. Most of the currently available tools that try to mitigate this problem require varying levels of user intervention. This paper presents a system, called UCheck, that detects errors in spreadsheets automatically. UCheck carries out automatic header and unit inference, and reports unit errors to the users. UCheck is based on two static analyses phases that infer header and unit information for all cells in a spreadsheet.

We have tested UCheck on a wide variety of spreadsheets and found that it works accurately and reliably. The system was also used in a continuing education course for high school teachers, conducted through Oregon State University, aimed at making the participants aware of the need for quality control in the creation of spreadsheets.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Spreadsheet; Unit; Type; Automatic error detection; Debugging; End-user software engineering

1. Introduction

Studies have shown that each year, tens of millions of professionals and managers create hundreds of millions of spreadsheets [1], thereby making spreadsheets the most widely used programming environment [2]. End-user programmers do not have a sound background in software design, development, or maintenance. As a result, existing spreadsheets contain

[☆]This work is partially supported by the National Science Foundation under the grant ITR-0325273 and by the EUSES Consortium (<http://EUSESconsortium.org/>).

*Corresponding author. Tel.: +1 541 737 8893; fax: +1 541 737 3014.

E-mail addresses: abrahamo@eecs.oregonstate.edu (R. Abraham), erwig@eecs.oregonstate.edu (M. Erwig).

errors at an alarmingly high rate [3,1,4,5]. Some studies even report that 90% or more of real-world spreadsheets contain errors [6]. In many instances, spreadsheet errors have resulted in huge financial losses to the organizations concerned [7–13]. Many of these “horror stories” have been documented in [14].

Our goal is to enable end users to develop and maintain reliable spreadsheets. To this end we have designed and implemented a *unit reasoning* system called UCheck that allows end users to identify and correct errors in their spreadsheets. The general idea behind the unit reasoning approach is to exploit information in spreadsheets about labels and headers to check the consistency of cell data and formulas.

UCheck basically consists of two components.

1. A formal reasoning system for detecting unit errors that is based on header information for a spreadsheet [15].
2. A header inference system that automatically determines header information for spreadsheets [16].

The unit system uses dependent units, multiple units, and unit generalization to classify the contents of spreadsheets and to check the consistent usage within formulas. Using units, which are based on values in spreadsheets, allows content classification on a more fine-grained level than types do. Moreover, we can communicate with the users in terms of objects contained in the spreadsheet, without having to resort to the abstract concept of types. The advantage of this approach is that it brings the strengths of static type checking to spreadsheets without end users having to pay the cost of learning about type systems.

Consider, for example, the spreadsheet in Fig. 1. We can observe that cell C4 does not just contain a number. In the context of this spreadsheet, it represents the number of oranges harvested in June because of the corresponding column and row headers. These kinds of relationships between labels and cells are determined automatically by the system.

Another example is the cell E4. Its unit is inferred to be June and Fruit. This inference is made from the fact that the formula contains a SUM function and aggregates over Apple, Orange, and Plum, which are all identified as being labeled Fruit. Now if the range in the SUM formula was A4:D4 instead, the formula would be aggregating over Apple, Orange,

	A	B	C	D	E	F
1		Fruit				
2	Month	Apple	Orange	Plum	Total	
3	May	4	5	6	=SUM(B3:D3)	
4	June	7	7	8	=SUM(B4:D4)	
5	July	5	5	0	=SUM(B5:D5)	
6	Total	=SUM(B3:B5)	=SUM(C3:C5)	=SUM(D3:D5)	=SUM(B6:D6)	
7						

Fig. 1. Inferring units from headers.

Download English Version:

<https://daneshyari.com/en/article/524579>

Download Persian Version:

<https://daneshyari.com/article/524579>

[Daneshyari.com](https://daneshyari.com)