

AN ABSTRACT OF THE THESIS OF

Jay W. Summet for the degree of Master of Science in Computer Science presented on July 23, 2001. Title: End-User Assertions: Propagating their Implications.

Abstract approved: _____
Margaret M. Burnett

Spreadsheet languages are the most commonly used end-user programming paradigm, yet spreadsheets commonly contain errors. Research shows that a significant number of spreadsheets (20%-40%) created by end users contain errors. In an attempt to reduce this error rate, this work presents an assertion propagation system for an end-user spreadsheet programming language, along with proofs of correctness, and complexity analysis. In addition to the traditional benefits of assertions (dynamic error checking and the documentation of programmer assumptions) this system deductively propagates the implications of assertions. This propagation adds two benefits, the cross-checking of program logic, and additional immediate visual feedback about the range of behavior of the program code for the end-user.

End-User Assertions: Propagating their Implications

by
Jay W. Summet

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirement for the
degree of

Master of Science

Presented July 23, 2001
Commencement June 2002

Master of Science thesis of Jay W. Summet presented on July 23, 2001.

APPROVED:

Major Professor, representing Computer Science

Chair of Department of Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Jay W. Summet, Author

ACKNOWLEDGEMENT

This work was supported in part by NASA Space Grant (NGT540022), and a NSF ITR grant (#0082265). The material in the ITR grant proposal (which I assisted in writing) served as a basis for the introduction and related works sections of this thesis. The graphical user interface which allows users to specify assertions and for the system to display system generated assertions was designed and implemented by Christine Wallace. Chris also ran a protocol analysis experiment to evaluate the usability of assertions with end users as part of her thesis work, and provided many suggestions during conversations about assertion behavior and user interaction. I am in debt to the entire Forms/3 group, both current and past, for the many years of work designing and implementing the Forms/3 language, without which this work would be homeless. Mark, Laura, Chris, Dan, Josh, Miguel, Andy and Bing were especially helpful in gleefully bringing implementation bugs to my attention. Special thanks to Dr. Erwig and Dr. Cook for their help with abstract interpretation and assertion behavior respectively, and to Dr. Hal Parks of the OSU Mathematics Department for pointers to interval arithmetic. And of course, my major professor Dr. Burnett has been my greatest supporter, helping both guide the actual work presented here and preparing me for the academic life in general. Outside of school, I'd like to thank my Mother and Father, Mhairi Raven and the great folks of the Shire of Coeur Du Val.

TABLE OF CONTENTS

| | <u>Page</u> |
|--|-------------|
| 1. INTRODUCTION..... | 1 |
| 1.1 SPREADSHEET ERRORS AND END-USER SOFTWARE ENGINEERING..... | 1 |
| 1.2 ASSERTIONS AND FORMAL TECHNIQUES..... | 2 |
| 1.3 THE USERS VIEW OF ASSERTIONS..... | 4 |
| 1.4 OVERVIEW OF THIS WORK..... | 7 |
| 2. RELATED WORK..... | 9 |
| 2.1 DERIVING ASSERTIONS COMPARED TO OTHER FORMS OF ANALYSIS AND INTERPRETATION..... | 9 |
| 2.2 SPREADSHEETS AND OTHER END-USER PROGRAMMING SYSTEMS..... | 14 |
| 3. DEDUCTIVE PROPAGATION OF ASSERTIONS..... | 18 |
| 3.1 DEFINITIONS..... | 18 |
| 3.2 FORWARD PROPAGATION OVERVIEW..... | 21 |
| 3.3 LIMITATIONS ON ASSERTION PROPAGATION..... | 27 |
| 3.4 END-USER COMPREHENSION..... | 29 |
| 4. PROPAGATION METHOD AND CORRECTNESS..... | 31 |
| 4.1 CORRECTNESS OF OPERATOR AND OPERAND REPLACEMENT..... | 32 |
| 4.2 ASSERTION-SPECIFIC OPERATOR CORRECTNESS..... | 33 |

TABLE OF CONTENTS (Continued)

| | <u>Page</u> |
|--|-------------|
| 5. ALGORITHM COMPLEXITIES..... | 48 |
| 5.1 NORMAL USAGE COST OF ASSERTION PROPAGATION..... | 49 |
| 5.2 CAUSES OF WORST CASE COMPLEXITIES..... | 50 |
| 5.3 ANALYSIS OF WORST CASE COMPLEXITIES..... | 52 |
| 6. FUTURE WORK..... | 54 |
| 6.1 IMPROVEMENTS IN PROPAGATION..... | 54 |
| 6.2 IMPROVEMENTS IN COLLABORATION..... | 56 |
| 6.3 KEEPING FORMS/3 LAZY..... | 57 |
| 6.4 SUPPORTING TEMPORAL AND REGION-BASED PROGRAMMING..... | 58 |
| 7. CONCLUSION..... | 61 |
| BIBLIOGRAPHY..... | 63 |

LIST OF FIGURES

| <u>Figure</u> | <u>Page</u> |
|---|-------------|
| 1. A temperature conversion ($^{\circ}\text{F}$ to $^{\circ}\text{C}$) spreadsheet at three points in a modification task of reversing the conversion ($^{\circ}\text{C}$ to $^{\circ}\text{F}$). As initially given to the user (a), showing the system's response (b) to the modifications of the guard on cell input_temp to range from zero to 100, and the final spreadsheet after all modifications have been made (c). | 5 |
| 2. This dialog which displays both the assertion propagated by Forms/3 and the user specified assertion, is viewed by clicking on a guard. At this point in the modification task, the dialog is indicating that Forms/3 does not agree with the user supplied range. The user correctly interprets this to mean that there is a problem with their formula for the output_temp cell. | 6 |
| 3. An example from [Ernst et al. 1999] showing likely invariants inferred by their system at the end of a program which "sums the values in array B (of length N) into result variable S". The results above were inferred by observing the instrumented programs behavior on "100 randomly generated arrays of length 7 to 13, in which each element was a random number in the range -100 to 100, inclusive". | 10 |
| 4. These tables, from [Jeffords and Heitmeyer 1998], illustrate the SRC mode transition table in tabular form (top) for an automobile cruise control system which was produced by the system designer or programmer and the corresponding table (bottom) showing assertions in the form of entry conditions, exit sets, and invariants generated by their algorithm over the course of four iterations. | 11 |

LIST OF FIGURES (Continued)

| <u>Figure</u> | | <u>Page</u> |
|---------------|--|-------------|
| 5. | The above example shows a portion of a specification file for the STeP system taken from [Bjørner et. al. 1995]. | 12 |
| 6. | Correspondence between the Cousot's abstract interpretation framework (left) and assertion propagation. While assertion propagation and abstract interpretation share similarities, assertions are not generated by abstraction from the static semantics of the program. | 13 |
| 7. | An example of Excel's data validation dialog. | 14 |
| 8. | A simple example showing forward propagation of a range assertion. A user specified assertion is indicated by the stick-figure icon, while the system generated assertion is indicated by the computer. | 22 |
| 9. | This figure demonstrates the difficulties of propagating assertions through formulas with shared dependencies. The actual implementation does not produce the assertion that is marked with the ERROR arrow. | 28 |
| 10. | In this example, both input assertions are violated (as indicated to the user by red "conflict" ovals around the values), but because the value in the Output cell could be produced by values which would not violate the input assertions, (e.g. $20 - 16 = 4$) the output assertion is not violated. | 32 |

LIST OF FIGURES (Continued)

| <u>Figure</u> | <u>Page</u> |
|---|-------------|
| 11. The three cases of range overlap, and their handling under the range merge algorithm. | 37 |
| 12. The results of propagating sub-assertions through a division operator. The range sub-assertion on cell C accepts numbers from negative infinity to -0.5, and then (continued off-screen) from 0.2 to positive infinity. | 41 |
| 13. Because the range of possible values for the SafetyCalcs cell is below twenty, the System_Safe? cell will always have a true value (as indicated by the “True” Boolean sub-assertion displayed above it). The IF expression in the Output cell only propagates the assertion on the THEN expression (from the Pressure cell) because the predicate will always be true. | 46 |
| 14. This figure shows how the original formula (a) of the Painted_Gizmos cell can have references substituted (b), and through symbolic evaluation (c,d) be simplified to remove shared dependencies. | 54 |
| 15. A simple example of the difficulties presented by the IF operator. | 55 |
| 16. A grid cell in Forms/3. The four cells on the lower right share a formula (displayed). | 60 |

LIST OF TABLES

| <u>Table</u> | | <u>Page</u> |
|--------------|---|-------------|
| 1. | Special cases handled by the assertion propagation algorithms | 42 |
| 2. | Return values of the Forms/3 IF operator | 45 |
| 3. | Return values of the assertion-IF operator | 45 |
| 4. | List of variables used in Chapter 5 | 48 |