Tool Support for Checking the Real-World Validity of Software

Jian Xiang, John Knight, Kevin Sullivan

Department of Computer Science

University of Virginia

Charlottesville, VA USA

{Jian,Knight,Sullivan}@cs.virginia.edu

*Abstract*— Logic interpretations

Keywords— Prototype, logic interpretation, real-world types, software assurance

# Introduction

### Problem with uninterpreted logic

### Introduction of interpretation and interpreted formalism

### Purpose of the prototype

### Outline of the contents

# Background

## Interpreted formalism

### Interpretation

### Interpreted formalism

### Real-world type system

## Pragmatics of interpreted formalism

### Development

### Error anaylsis techniques

# Requirements of the prototype

## Developing interpreted formalism

### Manual creation

### Synthesis framework

## Analysis techniques

### Real-world constraint checking

### Interval analysis

### Runtime Assertions

### Targeted Inspection

# Design of the Toolset

## Overview

### Figure of the prototype

## Use of the prototype

### Important Operations

### Analysis Technique

### Developing Interpretations

## Language specific decision

### Typed Java Elements

### Type Conversion

## Implementation

### Representation for interpreted formalism

### Error checking implementation

#### ASTVisitor

#### Possible Erroreous Statement

### Synthesis implementation

# Experiences

## Case study

## Observations

# Interfaces

## Main interface

## Error detection interface

# Related Work

# Conclusion

##### References

1. L. Jiang, and Z. Su. “Osprey: a practical type system for validating dimensional unit correctness of C programs,” in Proceedings of the 28th international conference on Software engineering (ICSE '06). ACM, New York, NY, USA, 2006, pp. 262-271.