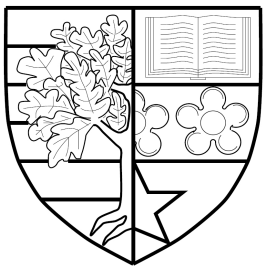
FROM FORMAL SPECIFICATION TO FULL PROOF: A STEPWISE METHOD

by

Lavinia Burski



Submitted for the degree of
Doctor of Philosophy

Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University

March 2016

Abstract

Proving formal specifications in order to find logical errors is often a difficult and labour-intensive task. This thesis introduces a new and stepwise toolkit to assist in the translation of formal specifications into theorem provers, using a number of simple steps based on the MathLang Framework. By following these steps, the translation path between a Z specification and a formal proof in Isabelle could be carried out even by one who is not proficient in theorem proving.

Acknowledgements

I dedicate this thesis to my loving and supportive boyfriend, Jeff.

Contents

List of Tables

List of Figures

Todo list

Acronyms

ASM Abstract state machine.

CGa Core Grammatical aspect.

DRa Document Rhetorical aspect.

GPSa General Proof Skeleton aspect.

Gpsa General Proof Skeleton aspect.

GpsaOL General Proof Skeleton ordered list.

Hol-Z Hol-Z.

IEC International Electrotechnical Commission.

MathLang MathLang framework for mathematics.

PPZed Proof Power Z.

SIL Safety Integrity Levels.

SMT Satisfiability Modulo Theories.

TSa Text and Symbol aspect.

UML Unified Modeling Language.

UTP Unifying theories of programming.

ZCGa Z Core Grammatical aspect.

ZDRa Z Document Rhetorical aspect.

ZMathLang MathLang framework for Z specifications.

Glossary

computerisation The process of putting a document in a computer format.

formal methods Mathermatically rigorous techniques and tools for the specification, design and verification of software and hardware systems.

formalisation The process of extracting the essence of the knowledge contained in a document and providing it in a complete, correct and unambiguous format.

halfbaked proof The automatically filled in skeleton also known as the Half-Baked Proof.

partial correctness A total correctness specification [P] C [Q] is true if and only if, whenever C is executed in a state satisfying P and if the execution of C terminates, then the state in which Cs execution terminates satisfies Q.

semi-formal specification A specification which is partially formal, meaning it has a mix of natural language and formal parts.

total correctness A total correctness specification [P] C [Q] is true if and only if, whenever C is executed in a state satisfying P, then the execution of C terminates, after C terminates Q holds.

Chapter 1

Background

Introduction stating formal methods are a type of mathematics. Explanation of where formal languages came from in mathematics etc

1.1 History of mathematics

Intro....

1.1.1 Right from the begininning

- logic and aristotle
- Frege Grundgesetze + Cantor + Russel discovered paradoxes
- Russel inventing type theory
- Zermelo added axiomisation
- Fraenkel + skolem extended to ZF set theory (which Z is based on)

1.1.2 Computerisation of Maths and Proof Systems

- typesetting systems like LaTeX
- proof assistants and automated theorem provers
- Semantical oriented document representations like OpenMath and OMDoc

1.1.3 Conclusion

In summary....

1.2 MathLang for mathematics

Intro....

1.2.1 Overview and Goals

1.2.2 Detailed information on CGa

- Reference Zenglars quote
- Weak type theory into CGa

1.2.3 Detailed information on DRa

- relations
- instances

1.2.4 Detailed information on skeletons

1.2.5 information on TSa

1.2.6 A full worked examples in mathlang

1.2.7 MathLang for Z

- [2] states what ro do to make formal methods industrial strength
- [1] stating in future work mathlang should be developed to cope with more mathematics (formal spec is a type of mathematics)
- diagram of math text to theorem prover using mathlang + diagram of specification to theorem prover using mathlang

1.2.8 Conclusion

1.3 History of formal methods and formal languages

I don't know if history of formal methods should be a seperate section or keep it as a subsection under 'history of mathematics'.

- definitions of 'formal language', 'formal method' and 'formal specification'
- the first formal language is thought to be used by Frege in his Begriffsschift (1879), Begriffschift meaning 'concept of writing' described as 'formal language of pure thoguht'
- broad histroy of formal methods
  - 1940's, Alan Turing annotated the properties of program states to simplify the logical analysus of sequential programs
  - 1960's Floyd, Hoare and Naur recommended using axiomatic techniques to prove programs meet their specification.
  - 1970's Dijkstra used formal caluculus to aid development of non-deterinist programs
- Formal methods today
- why use formal methods in industry (design errors like Therac-25 1985, NASAs Checkout Launch and Control System (CLCS) cancelled 9/2002, , added level of rigor)
- types of formal methods (Z, B method, ABS)
- Success of formal methods (B27 Traffic Control System, SHOLIS project, Data Acquisition, Monitoring and Commanding of Space Equipment)
- Weakness of formal methods (Low-level ontologies, Limited Scope,Cost,Poor tool feedback)

- What needs to be done to make formal methods industrial strength?
  - Bridge gap between real world and mathematics
  - Mapping from formal specifications to code (preferably automated)
  - Patterns identified
  - Level of abstraction should be supported
  - Tools needed to hide complexity of formalism
  - Provide visualization of specifications
  - Certain activities not yet formulizable methods
  - No one model has been identified which should be used for software)

ZMathLang covers items 1, 3, 5, 6, 7(semi formal spec)

1.3.1 Conclusion

1.4 Z Syntax and semantics

1.5 Proving systems for Z

1.5.1 Levels of Rigor

- Level 1 represents the use of mathematical logic to specify the system.
- Level 2 uses pencil-and-paper proofs.
- Level 3 is the most rigorous application of formal methods.

1.6 Conclusion

Bibliography

[1] R. Lamar. *A Partial Translation Path from MathLang to Isabelle*. PhD thesis, Heriot-Watt University, 2011.

[2] C. V. Stringfellow. Formal methods presentation. September 2016.