UNIVERSITY NAME

DOCTORAL THESIS

Thesis Title

Author:
John SMITH

Supervisor:
Dr. James SMITH

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Research Group Name
Department or School Name

April 8, 2021

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY NAME

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Thesis Title

by John SMITH

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don’t forget to include your project advisor…

Contents

List of Figures

List of Tables

For/Dedicated to/To my…

Chapter 1

Introduction

1.1 Safety-critical systems

- present the context of safety-critical systems, examples of safety-critical systems
- present the context of formal methods, program proofs
- present the fact that it is a hot topic
- present the context of neuroprotheses and the need for a safe design of digital circuits

Then, present the structure of the memoir and the content of the different chapters. The structure of the memoir follows the different steps necessary to establish the proof of semantic preservation. Each chapter corresponds to a step.

Chapter 2

The HILECOP methodology

2.1 Model-based System Engineering and the HILECOP methodology

- present the principles of Model-based System Engineering
- present the domain of formal methods
- present the HILECOP methodology

2.2 Verifying the HILECOP methodology

- RESEARCH QUESTION (A.K.A THE PROBLEM):
  CAN WE PROVE THAT THE HILECOP MODEL-TO-TEXT TRANSFORMATION IS SEMANTIC-PRESERVING?
- ADDITIONAL QUESTIONS:
  + How do we prove that?
  + What are the similarities with compiler verification?
  + What are the specificities?
- present the task motivation: the creation of safe digital circuits; first step, the behavior of the input model must be preserved in the output VHDL program
- present the model-to-text transformation, i.e zoom in on the step in the HILECOP methodology figure (but not too much since a whole chapter will be dedicated to the transformation)

➔ talk about the existing HILECOP software, and give views showing what an input model looks like
➔ transition idea: "inspired by the work on compiler verification, we had an idea about how to prove that the transformation is semantic-preserving, and which steps to follow"

Chapter 3

Preliminary Notions

- Present the Coq proof assistant and the concepts underlying the system
- Present the inductive type system
- Present what are dependent types
- Present the proof system (including tactics)

Chapter 4

Implementation of the HILECOP high-level models

RESEARCH QUESTION OF THE CHAPTER:

WHICH FORMAL SEMANTICS SHOULD WE CONSIDER IN THE CASE OF HILECOP HIGH-LEVEL MODELS?

4.1 General facts about Petri nets

- Make a general presentation of Petri nets.
  ➔ maybe put the general presentation of PNs in appendix, and focus here on the PNs semantics.
- Literature review on PNs semantics.
  ➔ don't forget to mention linear logic when talking about PNs semantics

➔ CONCLUSION: "we choose to adopt the formal semantics set in previous work".
➔ insist on the fact that most of the formalization is not my contribution, but comes from previous works

4.2 Synchronously executed Petri Nets

- Informal presentation of SITPNs
  ➔ check out what we did for SEFM and PNSE

4.3 Formal definition of SITPNs & semantics

- Present the formal structure of SITPNs, main definitions, and semantics
  ➔ check out SEFM and PNSE, and the last formalization of SITPN semantics (with the new definition of the set of *Fired* transitions)
  ➔ it's better to present the changes made in the semantics in the chapter about the proof
- Present the contribution to formalizing the notion of well-defined SITPN, and motivate the reason of the contribution

➔ the motivation is as follows: if sitpn is not well-defined, then the transformation returns an error + well-definition otherwise cannot prove behavior similarity

4.4 Coq implementation of SITPNs

- Talk about the dependtly-typed version
- Give the SITPN structure + SITPN state structure
- Give the state transition relation
- Give the full execution relation
- Talk about the implementation of the set of fired transitions

CONCLUSION OF THE CHAPTER:

- We will consider the formal semantics defined in previous theses to perform the proof of semantic preservation
- Reminder of the contributions: an operational formalization of the set of fired transitions + formalization of conflict resolution and well-defined SITPNs + Coq implementation

Chapter 5

$\mathcal{H}$-VHDL: a target hardware description language

RESEARCH QUESTION OF THE CHAPTER:

WHICH FORMAL SEMANTICS OF VHDL WILL MOST SUIT OUR NEEDS?

5.1 Presentation of the VHDL language

- Present the main concepts of VHDL
- Present the informal semantics of VHDL, i.e as described in the *Language Reference Manual* (LRM):
  - elaboration
  - simulation
- Present what use is made of VHDL in HILECOP
  - show our simulation cycle
  - present the P and T designs, and maybe a very high-level view of the transformation
  - show a figure of the P and T designs interfaces

  ➔ Don't go in too much details while presenting the use of VHDL in HILECOP. Details of which VHDL constructs are used in our programs will be given in the next section, and details on the transformation will be given in the next chapter.

5.2 Choosing a semantics for VHDL

- Our needs for a formal semantics
  ➔ reminder of the final goal of the thesis
  ➔ Do not describe what the semantics must look like, but rather remind the purpose of the semantics in relation to the PhD goal.
- Research question: which semantics should we consider?
  ➔ express the trade-off between
  1. using existing tools

2. creating our own tools

- Formalization depends on our needs
  - ➔ present our needs
  - ➔ present the qualifying criterions, and talk about the mandatory hypotheses about stable signals
    ➔ Talking about the mandatory hypotheses first here. Decide the level of precision we want our semantics to have.

- Presentation of the literature works and relations to our needs
  $\boxed{\text{CONCLUSION:}}$ "We choose to set up our own semantics."

5.3 A natural semantics for $\mathcal{H}$-VHDL

➔ Refer to appendix Reminder on natural semantics for a presentation of natural semantics on a very simple imperative programming language.

- Abstract syntax
- Semantical domains
- Elaboration
- Simulation

➔ Add examples of derivation of elaboration and simulation rules + P and T design in concrete syntax and abstract syntax.

5.4 Coq implementation of $\mathcal{H}$-VHDL

- Give some interesting parts of the code (maybe implementation of Δ and σ environment)
- Discussing improvements with dependent-types?
- Give some inductive types, simulation relations maybe?
- Give metrics about the code
- Give link to Git repo

CONCLUSION OF THE CHAPTER:

Contributions:

- setting a semantics for $\mathcal{H}$-VHDL, fitting our needs closely, but based on previous formalization works
- implementing the simulation semantics + abs. syntax in Coq

Chapter 6

The HILECOP model-to-text transformation

RESEARCH QUESTION OF THE CHAPTER:

IS THERE A PROPER WAY TO EXPRESS OUR (I.E, HILECOP'S) TRANSFORMATION FUNCTION?

6.1 Expressing transformation functions

- Literature review on compiler verification, model-to-model and model-to-text transformation verification
- Additional questions that the lit. review results try to give an answer to:
  - How to implement the transformation function to ease proof verification?
  - How to implement the transformation function to make it extensible? modular? (thinking about new elements added to the source model)

6.2 Informal presentation of the transformation

- High-level presentation of the transformation
  ➔ use the figures of the 2nd CSI
- point out the particularities of HILECOP's transformation function compared to what have been presented in the lit. review.

  ➔ don't know if the lit. review will give enough material to do so

  ➔ are there cases of model-to-text transformation close enough to our case to enable comparison?
- Give an example of transformation in the form of schematic figure.

CONCLUSION OF THE SECTION: Hard to express our trans. fun. in a way that facilitate compositional reasoning (if compositional reasoning is at all possible), partly because component instances are connected. However, the lit. review pointed out the necessity to keep a binding between elements of the source models and elements of the target programs (introduction of the γ binder) to enable state comparison, i.e on which relies the proof of semantic preservation.

6.3 The transformation algorithm

➔ maybe talk about the fact that the transformation algorithm has been established by observing the behavior of the current HILECOP implementation, and through discussions with the HILECOP designer (a.k.a David Andreu)

- Give the model-to-text transformation algorithm
  ➔ give the alg. as established through our meetings, or maybe use a simpler alg. (without the generate infos part, maybe too much implementation oriented)
- Give the formal specification of the transformation, i.e which structural properties between SITPN elements and $\mathcal{H}$-VHDL constructs
  ➔ don't know if I'll have the time to do that, but it would be great to have it, and use it in the proof. Each time the proof says "by construction", then the reader can refer to the formal spec
- Give a concrete example of generated $\mathcal{H}$-VHDL code from a simple SITPN

6.4 Coq implementation of the HILECOP model-to-text transformation

➔ talk about the state-and-error monad, the γ binder, the use of generic list functions (fold, map, filter, iter...)

CONCLUSION OF THE CHAPTER: Our transformation function is not really expressible in a compositional way as P and T instances are bound together. It's a specific case of model-to-text transformation. However, as much as the comparison with other works of transformation verification holds, we have been trying to inspire ourselves from the literature to implement the HILECOP model-to-text transformation function in Coq.

Chapter 7

Proving semantic preservation in HILECOP

RESEARCH QUESTION OF THE CHAPTER:

WHAT IS THE STRATEGY TO PROVE THAT THE HILECOP MODEL-TO-TEXT TRANSFORMATION FUNCTION IS SEMANTIC-PRESERVING?

➔ Refer to appendix Reminder on induction principles for a presentation of induction principles that will be used through the proof.

7.1 Tranformation functions and proof strategies

➔ maybe explain the difference between compiler verification and compiler validation, and stress the fact that we do verification
➔ maybe to close to the first part of X. Leroy's article on CompCert

- Lit. review on proof strategies
  - are there usual proof strategies?
  - do they apply in our case?
- Present the main results (mostly coming from the first part of X. Leroy's article on CompCert):
  - proofs are based on execution steps, and state comparison
  - ...

7.2 Behavior preservation theorem

- preliminary definitions for the proof (+ notations to follow the proof)
  ➔ illustrate the state similarity relation
- definitions of main thms + proofs (proofs refer to lemmas that will be presented in the next section)
- illustrating the main thms (execution steps figure)

7.3 Proving semantic preservation

- Give the proof (the whole proof? maybe a bit long, some part as appendices?)
  1. Initial states
  2. First step
     ➔ not sure it is really relevant, as it uses most of the material coming from the "Initial states" part and the "Rising edge" part
  3. Rising edge
  4. Falling edge
- Illustrate some point of the proof (determine which points) with figures
  ➔ illustrate the part that says "by construction", linking PN parts to $\mathcal{H}$-VHDL parts

7.4 A detailled proof: equivalence of fired transitions

➔ don't know if this is a separate section or if it must be a part of the previous section

- Informal presentation of the proof
- formal proof
- bug detection (illustrated, show the figure)

7.5 Verification of the proof of behavior preservation

- Present the interesting points in the verification of the proof with Coq

CONCLUSION OF THE CHAPTER:

- Remind of the main proof strategy
- Remind of the bug detection, the changes on the SITPN and $\mathcal{H}$-VHDL semantics that are consequences of the proof
- State of the proof verification. How far am I right now?

Chapter 8

Conclusion

8.1 Contributions

- Contributions to SITPN
- Contributions to VHDL
- Contributions to transformation functions
- Proof
  - Bug detections + changes in semantics
  - How far are we to complete the verification?

8.2 Improvements and perspectives

➔ if proof not yet verified, first thing is not complete this job!

- improvements on the implementation of $\mathcal{H}$-VHDL, get closer to the formal def., more use of dependent types
- proving the semantic preservation thm in its existential version, i.e:
  - the generated design is always elaborable
  - the generated design is always simulable, i.e, it will never produce errors as long as the input model verifies some properties (liveness, boundedness…)
- consider the whole HILECOP high-level model
  - macroplaces
  - GALS

Appendix A

Reminder on natural semantics

Appendix B

Reminder on induction principles

- Present all the material that will be used in the proof, and that needs clarifying for people who do not come from the field (e.g, automaticians and electronicians)
  - structural induction
  - induction on relations
  - ...

Bibliography

Arnold, A. S. et al. (Mar. 1998). “A Simple Extended-Cavity Diode Laser”. In: *Review of Scientific Instruments* 69.3, pp. 1236–1239. URL: http://link.aip.org/link/?RSI/69/1236/1.

Hawthorn, C. J., K. P. Weber, and R. E. Scholten (Dec. 2001). “Littrow Configuration Tunable External Cavity Diode Laser with Fixed Direction Output Beam”. In: *Review of Scientific Instruments* 72.12, pp. 4477–4479. URL: http://link.aip.org/link/?RSI/72/4477/1.

Wieman, Carl E. and Leo Hollberg (Jan. 1991). “Using Diode Lasers for Atomic Physics”. In: *Review of Scientific Instruments* 62.1, pp. 1–20. URL: http://link.aip.org/link/?RSI/62/1/1.