Karlsruhe Institute of Technology

Online Neural Network-based Language Identification

Master's Thesis of

Daniel H. Draper

at the Department of Informatics
Institute for Anthropomatics and Robotics

Reviewer:	Dr.-Ing. Sebastian Stüker
Second reviewer:	
Advisor:	M.Sc. Markus Müller

12. December 2016 – 11. May 2017

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 12th of May, 2017

. . . . . .

(Daniel H. Draper)

Abstract

Zusammenfassung

Contents

List of Figures

List of Tables

1 Preliminary Definitions

In the following chapter we want to define and explain terms, models and logic used throughout this thesis. We skip the explanation of first-order logic, which we presume to be known by the reader.

Bibliography

[ABH+07] Wolfgang Ahrendt, Bernhard Beckert, Reiner Hähnle, Philipp Rümmer, and Peter H. Schmitt. Verifying object-oriented programs with KeY: A tutorial. In *5th International Symposium on Formal Methods for Components and Objects, Amsterdam, The Netherlands*, volume 4709 of *LNCS*, pages 70–101. Springer, 2007.

[AD94] Rajeev Alur and David L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235, 1994.

[BHS07] Bernhard Beckert, Reiner Hähnle, and Peter H. Schmitt, editors. *Verification of Object-Oriented Software: The KeY Approach.* LNCS 4334. Springer-Verlag, 2007.

[CFH+03] Edmund Clarke, Ansgar Fehnker, Zhi Han, Bruce Krogh, Olaf Stursberg, and Michael Theobald. Verification of Hybrid Systems Based on Counterexample-Guided Abstraction Refinement. In Hubert Garavel and John Hatcliff, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2619 of *Lecture Notes in Computer Science*, pages 192–207. Springer Berlin Heidelberg, 2003.

[CW98] Ana Cavalcanti and Jim Woodcock. ZRC – A Refinement Calculus for Z. *Formal Aspects of Computing*, 10(3):267–289, 1998.

[DB14] John Derrick and Eerke A. Boiten. *Refinement in Z and Object-Z.* Springer London, 2014.

[GZ14] Bin Gu and Liang Zou. A refinement calculus for hybrid systems. In *Engineering of Complex Computer Systems (ICECCS), 2014 19th International Conference on*, pages 176–185, Aug 2014.

[Hen00] Thomas A. Henzinger. The theory of hybrid automata. In M.Kemal Inan and RobertP. Kurshan, editors, *Verification of Digital and Hybrid Systems*, volume 170 of *NATO ASI Series*, pages 265–292. Springer Berlin Heidelberg, 2000.

[MLP12] Stefan Mitsch, Sarah M. Loos, and André Platzer. Towards formal verification of freeway traffic control. In Chenyang Lu, editor, *ICCPS*, pages 171–180. IEEE, 2012.

[MV92] Carroll Morgan and Trevor Vickers. *On the Refinement Calculus.* Springer London, 1992.

[Pla08] André Platzer. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*, 41(2):143–189, 2008.

[Pla10] André Platzer. *Logical Analysis of Hybrid Systems*. Springer, Pittsburgh, 2010.

[Pla15a] André Platzer. Guide for Keymaera Hybrid Systems Verification Tool. http://symbolaris.com/info/KeYmaera-guide.html, 2015.

[Pla15b] André Platzer. KeYmaera: A Hybrid Theorem Prover for Hybrid Systems. http://symbolaris.com/info/KeYmaera.html, 2015.

[RT13] Matthias Rungger and Paulo Tabuada. Abstracting and refining robustness for cyber-physical systems. *CoRR*, abs/1310.5199, 2013.

[RyS96] Jean-François Raskin and Pierre yves Schobbens. State Clock Logic: a Decidable Real-Time Logic. In *HART'97, LNCS 1201*, pages 33–47. Springer-Verlag, 1996.

[STW14] Steve Schneider, Helen Treharne, and Heike Wehrheim. The behavioural semantics of Event-B refinement. *Formal Aspects of Computing*, 26(2):251–280, 2014.

[Wol] Wolfram Research, Inc. Mathematica. https://www.wolfram.com.

Glossary

Cyber-Physical System[CPS] is a system describing motions or evolutions in which a physical aspect is being controlled by a computer/computer program. In this thesis equivalent to the notion of Hybrid Systems.

Differntial Dynamic Logic[DDL] is the logic in which we express the safety properties for our CPS, and it also includes a syntax to express differential equations.

Dynamic Logic is the logic we use to express the safety property for our CPS.

Glue is the relation between the values in the world of reals and the discrete world. For us, refers to a way of gaining the corresponding value in the other world from a given value.

Hook is the concrete instruction at which the control program is executed when describing a hybrid system as a hybrid program. This is one or multiple non-deterministic assignments of values, e.g a:=*.

Hook Safety Postcondition is the condition that has to be fullfilled by the value(s) that were assigned in the hook for the safey condition of the whole program to hold true.

Hybrid Automata is a way to model hybrid systems in the form of a non-deterministic automata. Uses the same syntax as finite automata with the addition of differential equations.

Hybrid Program is a way to describe hybrid systems in the form of a program. Expressed in the syntax of regular programs with the extension of differential equations.

Hybrid System is a system in which discrete as well as continuous evolutions are present. E.g, a remote controlled car which can only be accelerated or braked, its movement is continuous and follows continuous differential equations, while the control program is discrete and can only take discrete values (e.g, Acceleration := 1; Acceleration := 2 etc.).

Java Modelling Language is the language we use to express the contracts a certain method or class has to fulfill to be considered correct.

KeY is the tool we use to verify our java control programs as our concrete implementations.

KeYmaera is the tool we use to verify both our remodelled hybrid programs as well as the glue relation.