



Universiteit
Leiden

The Netherlands

Formal models of software-defined networks

Feng, H.

Citation

Feng, H. (2024, December 3). *Formal models of software-defined networks*.
Retrieved from <https://hdl.handle.net/1887/4170508>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4170508>

Note: To cite this publication please use the final published version (if applicable).

Formal Models of Software-Defined Networks

Hui Feng

Formal Models of Software-Defined Networks

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op dinsdag 3 december 2024
klokke 10:00 uur

door

Hui Feng
geboren te Handan, China
in 1993

Promotor: Prof.dr. M.M.Bonsangue

Promotiecomissie: Prof.dr. L. Barbosa (University of Minho, Portugal)
Prof.dr. M. Sirjani (Mälardalen University, Sweden)
Prof.dr. F. de Boer
Prof.dr. J. Kleijn
Prof.dr. A. Plaat
Dr. Q. Chen



Hui Feng was financially supported through the China Scholarship Council (CSC) to participate in the PhD program at Leiden University. Grant number 201706350092.

The research in this thesis was performed at Leiden Institute of Advanced Computer Science at Leiden University, under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

Copyright © 2024 Hui Feng.

To my parents.

“Fear is the main source of superstition, and one of the main sources of cruelty.
To conquer fear is the beginning of wisdom.”

Bertrand Russell, 1872-1970

Contents

1	Introduction	1
1.1	Formal methods in SDN	3
1.2	Research objectives	5
1.3	Publications	8
1.4	Overview	9
2	Software-defined Networks	11
2.1	Traditional networking vs. SDN	11
2.2	Protocols for SDNs	13
2.3	The OpenFlow protocol	14
2.3.1	SDN packets	15
2.3.2	OpenFlow flow tables and flow entries	15
2.3.3	OpenFlow messages	16
2.3.4	Evaluating the OpenFlow protocol	16
2.4	Two examples	17
3	Reo and constraint automata	21
3.1	A short introduction to Reo	21
3.2	Constraint automata	22
3.2.1	Constraint automata with memory	25
3.2.2	Basic and more complex connectors	29
4	A Reo Model of SDNs	37
4.1	Modeling the data plane	37
4.1.1	The Reo connector of a switch	38
4.1.2	Constraint automata for switches	40
4.2	Modeling the control plane	41

4.3	SDN models for two controller algorithms	44
4.4	Related work	48
4.5	Conclusion	49
5	Implementing Reo into Promela	51
5.1	A short introduction to Promela	51
5.2	Symbolic constraint automata	53
5.3	From Reo to Promela	61
5.3.1	Implementing Reo ports in Promela	61
5.3.2	Implementing Reo connectors in Promela	62
5.3.3	Other Reo connectors in Promela	66
5.4	A case study: verifying a SDN	68
5.4.1	A Promela SDN model via symbolic constraint automata	69
5.4.2	Verification and simulation	72
5.5	Related work	75
5.6	Conclusion	76
6	Concurrent NetKAT with Ports	79
6.1	Introduction	79
6.2	NetKAT	81
6.3	NetKAT with ports	86
6.4	NetKAT automata with I/O ports	90
6.4.1	From symbolic constraint automata to ioNKA	93
6.4.2	Composing ioNKA	95
6.5	Conclusion and future work	97
7	Towards causality reasoning for SCA	99
7.1	Introduction	99
7.2	Preliminaries	102
7.3	A Railway crossing Example	105
7.4	A Language-based causal model	106
7.5	Computing causes	110
7.6	Experimental evaluation	112
7.7	Extensions	114
7.8	Conclusions	115

8 Conclusions **117**

8.1 Main contributions 118

8.2 Future research directions 120

Samenvatting

Summary

Acknowledgements

Curriculum Vitae

Publication List

CONTENTS
