

MASTER'S THESIS

Verification tools for communication protocol design

Choy, Wai Hing

Date of Award:
2000

[Link to publication](#)

General rights

Copyright and intellectual property rights for the publications made accessible in HKBU Scholars are retained by the authors and/or other copyright owners. In addition to the restrictions prescribed by the Copyright Ordinance of Hong Kong, all users and readers must also observe the following terms of use:

- Users may download and print one copy of any publication from HKBU Scholars for the purpose of private study or research
- Users cannot further distribute the material or use it for any profit-making activity or commercial gain
- To share publications in HKBU Scholars with others, users are welcome to freely distribute the permanent URL assigned to the publication

Verification Tools for Communication Protocol Design

CHOY Wai Hing

A thesis submitted in partial fulfillment of the requirements
for the degree of
Master of Philosophy

January 2000

Hong Kong Baptist University

Abstract

We describe the work for developing a verification tool for communication protocol design. In our project, the communication protocol is modeled as CSP processes and the design is captured by the system in a graphical notation of CSP. The CSP description will be reasoned by the theorem prover HOL90. In this thesis, we explore the way to reason the CSP expressions with the theorem prover and show how the reasoning can be performed in an automated fashion.

Contents

Declaration	i
Abstract	ii
Acknowledgments	iii
List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Thesis organization	3
2 CSP and protocol verification	5
2.1 Communicating Sequential Process	6
2.1.1 CSP description	6
2.1.2 A simple example of CSP description	8
2.2 Analysis of CSP processes	10
2.2.1 Denotational semantics and process refinement	10
2.2.2 Process algebra	13
2.3 Application of CSP in protocol design	15
2.4 Discussion	17

3	The HOL theorem prover	19
3.1	Higher order logic and HOL theorem prover	20
3.1.1	Higher order logic	20
3.2	The meta language - ML	21
3.2.1	HOL system	22
3.3	Theorems and proofs in HOL	23
3.3.1	Forward proof	24
3.3.2	Goal directed proof	24
3.4	Embedding other formalism	26
3.5	Simplifier	26
4	CSP_G , a graphical description technique	29
4.1	Prefix construction	31
4.2	Sequential composition	32
4.3	Choice	32
4.4	Parallel composition	33
4.5	Hiding	34
4.6	Parenthesis	34
4.7	Machine-readable CSP	35
5	Mechanizing CSP in HOL	38
5.1	Embedding the syntax of CSP	39
5.2	Fix-point theory	42
5.2.1	Partial orders	42
5.2.2	Complete partial order	43
5.2.3	Fix-point theory in the context of CSP processes	44
5.3	Mechanizing the fix-point theory	44
5.4	Treatment of communication channel	46
5.5	Proving CSP laws	46

6	Implementing integrated CSP_G editor/verification tools	50
6.1	Implementation of GCSPED	51
6.1.1	Organization of GCSPED modules	52
6.1.2	Design issues of GCSPED::DRAW_DESK	54
6.1.3	File format of CSP_G file	55
6.2	Implementation of the CSP_G to CSP_M translator	59
6.2.1	Lexical analysis for CSP_G specification	59
6.2.2	Pre-processing of CSP_G specification	61
6.3	Implementation of CSPCONVERT	65
6.3.1	Parser implementation	66
6.3.2	Setup the environment	66
6.4	Implementation of VERIFIER	67
7	Application example	71
7.1	Definition of <i>BUFFER</i> specification in CSP_G	71
7.2	Definition of <i>BUFFER</i> implementation in CSP_G	73
7.3	Convert the CSP_G specification into a CSP_M specification	76
7.4	Verification of <i>BUFF</i> implementation	77
8	Conclusion	87
A	proof of CSP law	88
B	Proof of “Buffer” implementation	106
C	CSP laws	109
	Bibliography	118
	Curriculum Vitae	120