

RIJKSUNIVERSITEIT GRONINGEN

Design and Verification of Lock-free Parallel Algorithms

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. F. Zwarts,
in het openbaar te verdedigen op
vrijdag 15 april 2005
om 16.15 uur

door

Hui Gao

geboren op 8 september 1969
te Jiangsu, China

Promotores: Prof. dr. W.H. Hesselink
Prof. dr. ir. J.F. Groot

Beoordelingscommissie: Prof. dr. M. Herlihy
Prof. dr. W. Fokkink
Prof. dr. S.D. Swierstra

ISBN 90-367-2231-4



The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

Acknowledgements

Finally, I wish to express my gratitude to all persons who gave me the possibility to finish this thesis, though there are so many people that I can't mention all of them right here.

First, I would like to express my thanks to my former advisors Prof. Dr. H.G. Dehling and Prof. Dr. A.C. Hoffmann, and Prof. W. Schaafsma for creating the job opportunity for me to start the research as a Ph.D candidate in University of Groningen, in February 2000.

Then I would like to thank my present advisor Prof. Dr. Wim H. Hesselink for taking me over in February 2001, after both of my former advisors left the university of Groningen. In the following four years, he kept an eye on the progress of my work and was always available when I needed his advice. He has always been a great source of wisdom, knowledge and experience. My second advisor, Prof. Dr. Jan Friso Groote came to direct my research one year later. His critical comments, valuable hints and creative suggestions helped me to overcome many obstacles in my research. I am deeply indebted to my two advisors. The combination of their technical and editorial advice substantially contributed to the completion of this thesis. I am really glad and lucky that I have them both as my mentors.

I am very grateful to the members of the reading committee, Prof. Dr. Maurice Herlihy, Prof. Dr. Wan Fokkink, and Prof. Dr. Doaitse Swierstra, for reviewing the manuscript of my thesis and giving me many valuable comments.

Thanks also go to the chairman of Fundamental Computing Science group, Prof. Dr. Gerard R. Renardel de Lavalette, who was always willing to provide a solid financial support for me to attend summer schools and conferences.

I am grateful to all the members in our reading club, Wim H. Hesselink, Jan Eppo Jonker, Jan H. Jongejan, Hendrik Wietze de Haan and Roland Veen, for innumerable discussions, and valuable comments on all the drafts of the papers presented in this thesis.

I also want to extend my thanks to my former and present office mates, Daniel Straumann, Nico Kruithof, Hendrik Wietze de Haan and Marek Burza, for providing a good working atmosphere and sharing of tips, teas, cookies, and etc. Especially, I want to thank Hendrik Wietze de Haan for helping me to translate the summary of this thesis into Dutch. Big thanks to our secretaries, E.D. Elshof, D.J. Hansen and Y.G.M. Vergnes, for their kind assistance with various practical problems.

Many thanks goes to all my friends who've stood by me through all the years.

Last, but not least, I would like to give my special thanks to my family. It was my wife, Lixia Tang, who inspired me to study in the Netherlands five years ago. It was my son, Shenghan Gao, who frequently corrected the pronunciation of my English. My wife and my son's unconditional and endless love as well as their steady and substantial support are the main source of my life. My warm thanks go to my two elder sisters, Bi Gao and Hua Gao, for their kind concern and having taken care of my parents for so many years. My parents, Zhongnan Gao and Juru He, receive my deepest gratitude and love for always being there for me. Indeed, without love, inspiration, encouragement, and optimism from them, I would never have been able to finish this work.

Contents

Acknowledgements	iv
1 Introduction	1
1.1 Shared memory architectures	2
1.1.1 The programming model	3
1.1.2 Synchronization	4
1.2 Correctness	8
1.2.1 The temporal logic	8
1.2.2 Safety property	9
1.2.3 Liveness property	10
1.3 Verification	11
1.3.1 Model checking	11
1.3.2 Theorem proving	12
1.4 Introduction to PVS	14
1.4.1 The PVS specification language	14
1.4.2 The PVS prover	17
1.4.3 Experiences with PVS	17
1.5 Overview of the thesis	19
2 Lock-free dynamic hash tables with open addressing	21
2.1 Introduction	22
2.2 The interface	25
2.3 The algorithm	28
2.3.1 Hashing	29
2.3.2 Tagging of values	30

2.3.3	Data structure	30
2.3.4	Primary procedures	32
2.3.5	Memory management and concurrent migration	38
2.4	Correctness (Safety)	47
2.4.1	Main properties	48
2.4.2	Intuitive proof	49
2.4.3	The model in PVS	50
2.5	Correctness (Progress)	53
2.5.1	The easy part of progress	54
2.5.2	Progress of newTable	54
2.5.3	The failure of wait-freedom	55
2.6	Conclusions	56
3	A formal reduction for lock-free parallel algorithms	57
3.1	Introduction	58
3.2	Lock-free transformation	59
3.3	Reduction	60
3.3.1	Observed Specification	61
3.3.2	Refinement mappings	61
3.3.3	Correctness	62
3.4	A lock-free pattern	63
3.4.1	Simulation	65
3.4.2	Refinement	67
3.5	Large object	68
3.6	Conclusions	70
4	A general lock-free algorithm using compare-and-swap	72
4.1	Introduction	73
4.2	Synchronization primitives	75
4.3	The lock-free implementation using CAS	76
4.4	Correctness	78
4.4.1	Invariants	79
4.4.2	Refinement	80
4.4.3	Progress	81

4.5	Conclusions	83
5	Lock-free parallel garbage collection by mark&sweep	84
5.1	Introduction	85
5.2	Specification	88
5.3	A higher-level implementation	93
5.3.1	Data Structure	94
5.3.2	Algorithm	96
5.4	Correctness	105
5.4.1	The main loop	105
5.4.2	Safety properties	106
5.4.3	Liveness	108
5.5	The low-level lock-free implementation	113
5.6	Practical experiment	114
5.7	Conclusions	116
A	For lock-free dynamic hash tables	118
A.1	Invariants	118
A.2	Dependencies between invariants	129
B	For lock-free parallel GC	137
B.1	Invariants	137
B.2	Dependencies between invariants	141
B.3	The low-level lock-free algorithm	143
B.3.1	Data Structure	143
B.3.2	Algorithm	144
	Bibliography	151
	Summary	157
	Samenvatting	162