

DIRECTED UNFOLDING

*Reachability Analysis of Concurrent Systems
& Applications to Automated Planning*

by

Sarah Louise Hickmott

Thesis submitted for the degree of

Doctor of Philosophy

in

School of Electrical and Electronic Engineering
Faculty of Engineering
Computer and Mathematical Sciences

University of Adelaide, Australia

2008

This page left blank.

Abstract

The factored state representation and concurrency semantics of Petri nets are closely related to those of classical planning models, yet automated planning and Petri net analysis have developed independently, with minimal and mainly unconvincing attempts at cross-fertilisation. This thesis exploits the relationship between the formal reachability problem, and the automated planning problem, via Petri net unfolding, which is an attractive reachability analysis method for highly concurrent systems as it facilitates reasoning about independent sub-problems. The first contribution of this thesis is the theory of *directed unfolding*: controlling the unfolding process with informative strategies, for the purpose of optimality and increased efficiency. The second contribution is the application of directed unfolding to automated planning.

Inspired by well-known planning heuristics, this thesis shows how problem specific information can be employed to guide unfolding, in response to the formal problem of developing efficient, directed reachability analysis methods for concurrent systems. Complimenting this theoretical work, this thesis presents a new forward search method for partial order planning which can be exponentially more efficient than state space search.

Our suite of planners based on directed unfolding can perform optimal and suboptimal classical planning subject to arbitrary action costs, optimal temporal planning with respect to arbitrary action durations, and address probabilistic planning via replanning for the most likely path. Empirical results reveal directed unfolding is competitive with current state of the art automated planning systems, and can solve Petri net reachability problems beyond the reach of the original “blind” unfolding technique.

This page left blank.

For my parents.

Declaration

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

Sarah Hickmott

Acknowledgements

I would not have been able to complete this thesis without support from many people and in particular, supervision from Langford White, Ian Fuss and Sylvie Thiébaux. Lang provided me with an endless stream of ideas, but equally importantly, supported me to take my own direction with this project, and consistently conveyed his belief in me and this work. Ian challenged my ideas and worked to broaden my perspective and assist me in seeing a bigger picture. Sylvie provided invaluable guidance and encouragement; her energy and determination both assisted and inspired me. Without the knowledge, experience and generous support of each of these people I would not have been able to complete this work; my gratitude for each person's contribution is heartfelt.

Throughout this thesis there are significant contributions from Blai Bonet, Patrik Haslum and Jussi Rintanen, my co-authors in conjunction with Sylvie and Lang, on publications relating to this work. I am grateful for and appreciative of the insight, knowledge and enthusiasm Blai, Patrik and Jussi brought to this research. Patrik also read and provided useful feedback on this thesis for which I am thankful.

Over the last four years I have had the opportunity to discuss my work with many scientists, but am particularly grateful to Alban Grastien, Jinbo Huang, Subbarao Kambhampati, Sophie Pinchinat and David Smith for fruitful discussions; I also thank Stefan Schwoon for his assistance with MOLE.

This work has been financially supported by the University of Adelaide (UofA) Faculty of Engineering, Mathematical and Computer Sciences and the National ICT Australia (NICTA) via the Dynamic Planning, Optimisation and Learning Project (DPOLP). I would like to thank all members of DPOLP, for interesting discussions and useful responses to this work, in particular Jonathon Billington and Guy Gallasch for their technical advice, and Sanjeev Naguleswaran for valuable guidance. I thank Bob Williamson for helping to facilitate my extended visits to Canberra. I also wish to express my gratitude to Owen Thomas and Olivier Buffet, for programming support beyond the call of duty.

Finally, I take this opportunity to thank my friends, who have strongly supported and patiently endured me, throughout my increasing absorption in this project. And last but definitely not least I want to thank my family, who always stand by me, and have shared the difficulty as well as the excitement of this journey.

This page left blank.

Contents

1	Introduction	1
1.1	Initial Motivation	3
1.2	Contribution	5
1.3	Overview	7
I	Directed Unfolding	11
2	Reachability Analysis	13
2.1	System Modelling	14
2.1.1	Restricted State-Transition System	14
2.1.2	Concurrent System	17
2.2	The Reachability Problem	17
2.2.1	Definition	18
2.2.2	Relevance: Automated Planning, Formal Verification and Diagnosis	18
2.2.3	Optimal Solution	19
2.2.4	Partially Ordered Solution	20
2.2.5	Forward State Space Search	20
	Optimal Solution	20
2.3	The State Explosion Problem	21
2.3.1	Alleviating the Problem	23
	Symbolic Algorithms	24

	Partial Order Methods	25
	Decomposition Techniques	26
	Abstraction and Symmetry Reduction	27
2.3.2	Heuristic Search	28
	Heuristic State Space Search	29
2.4	Directed Unfolding: Facilitating Optimality and Improving Efficiency . . .	30
2.5	Conclusion	31
3	Unfolding a Petri Net	33
3.1	Petri net	33
3.1.1	Place Transition Net	34
3.1.2	General Assumptions	36
3.2	Unfolding a Place Transition Net	37
3.2.1	Unfolding: Representation and Method	38
	Branching Process	38
	Configuration	41
	Possible Extensions	42
	The Unfolding Algorithm	42
3.2.2	A Complete Finite Prefix of the Unfolding	43
	The ERV Unfolding Algorithm	44
	MOLE: An Implementation of The ERV Algorithm	44
3.3	The Reachability Problem	45
3.3.1	Connection with Unfolding	46
3.3.2	Complexity	47
3.3.3	On-the-fly Reachability Analysis via Unfolding	49
	The ERV-Fly Algorithm	51
	MOLE: An Implementation of the ERV-Fly Algorithm	51
3.4	Conclusion	52
3.4.1	Personal Contribution	53

4	Directed Unfolding	55
4.1	Reconsidering Adequate Orders	58
4.1.1	A Semi-Adequate Order on Configurations	58
4.2	Directing the Unfolding for Optimality	60
4.2.1	A Notion of Optimality	60
4.2.2	Notions of Cost	64
4.2.3	Optimal Cost Reachability Analysis	65
	Additive Cost	65
	Parallel Cost	66
4.3	Directing the Unfolding with Heuristics	72
4.3.1	Direct Translation	73
4.3.2	Generic Framework for Heuristic Guidance	73
4.3.3	Specific Instantiations	75
	Heuristic Guidance with Additive Cost Function	75
	Heuristic Guidance with Parallel Cost Function	77
	Summary	78
4.4	Size of the Finite Prefix	78
4.5	Heuristic Functions	80
4.5.1	Heuristic Functions For Additive Cost	81
4.5.2	Heuristic Function For Parallel Cost	84
4.6	Experimental Results	84
4.6.1	Petri Net Benchmarks	85
4.6.2	Random Problems	86
4.6.3	Planning Benchmarks	90
4.7	Conclusion	93
4.7.1	Personal Contribution and Collaboration	94

II	Planning Via Directed Unfolding	97
5	Automated Planning	99
5.1	Automated Planning	100
5.1.1	Practical and Theoretical Motivation	100
5.1.2	Domain Independent Planning	101
5.1.3	Conceptual Model of the Planning Problem	102
5.2	Classical Planning	103
5.2.1	Representation	105
	STRIPS Representation	105
5.2.2	Analysis	108
	State Space Planning	108
	Plan Space Planning	109
	Partial Order Planning and The Least Commitment Principle	110
	Planning As Propositional Satisfiability	112
	Graphplan	113
	Heuristic State Space Planning	115
	A Revival of Partial Order Planning?	115
5.3	Extending the Classical Planning Problem	116
5.3.1	Action Costs	116
	Additive Cost of a Plan	116
	Parallel Cost of a Plan	117
5.3.2	Probabilistic Action Effects	117
5.4	Conclusion	119
6	Planning Via Directed Unfolding	121
6.1	Translating a Planning Problem to a PT-Net System	122
6.1.1	Establishing 1-safe Planning Operators	125
	Example	126

Equivalence	126
6.1.2 Eliminating Negative Preconditions	129
Example	129
Equivalence	129
6.1.3 Mapping to PT-Net System	131
Example	131
Correctness	131
6.1.4 Limitations of Translation	133
Size of Translation	133
Notion of Concurrency	134
6.2 Planning as Reachability Analysis	136
6.2.1 Planning via Directed Unfolding	136
6.2.2 Probabilistic Action Effects	137
6.3 Comparison with Classical Planning Methods	139
6.4 The PUP SUITE	141
6.4.1 Artificial Problems	144
6.4.2 IPC Benchmarks	145
Translation Time	147
Suboptimal Classical Planning	147
Optimal Classical Planning	150
Optimal Temporal Planning	156
Replanning	156
Plan Flexibility	159
6.5 Conclusion	162
6.5.1 Personal Contribution and Collaboration	163
7 Conclusions	165
Bibliography	171

This page left blank.