

Contents

Abstract	v
Resumen	vii
Resum	ix
Acknowledgements	xi
Agradecimientos	xiii
List of Figures	xxi
I Introduction	xxiii
1 Introduction: Motivation and Contributions	1
1.1 Concurrent Programming Languages	1
1.2 Program Analysis of Concurrent Languages	3
1.3 Main Goals and Contributions	6
1.4 Organization of the Thesis	9
2 Preliminaries	11

2.1	The Syntax and the Semantics of CSP	11
2.2	Formal Definition and Basic Terminology of Petri Nets	18
3	Static Slicing of Explicitly Synchronized Languages	21
3.1	Context-Sensitive Synchronized Control Flow Graph	25
3.1.1	An Algorithm to Generate the CSCFG	31
3.1.2	Using the CSCFG for Program Slicing	37
3.2	Static Slicing of CSP Specifications	38
3.3	Implementation	42
3.3.1	Architecture of SOC	44
3.3.2	Benchmarking the Slicer	47
3.4	Related work	50
4	Tracking CSP Computations	53
4.1	Tracking Computations	56
4.2	Instrumenting the Semantics for Tracking	58
4.3	Implementation	64
4.3.1	Architecture of CSP-Tracker	65
4.3.2	Using CSP-Tracker	67
4.4	Related Work	69
5	From CSP Specifications to Petri Nets	71
5.1	Equivalence between CSP and Petri Nets	72
5.2	Transformation of a CSP Specification into an Equivalent PN	73
5.3	Tuning the Generated Petri Net	87

Contents

5.4	Implementation	89
5.5	Related Work	92
6	Dynamic Slicing Techniques for Petri Nets	93
6.1	Dynamic Slicing of Petri Nets	94
6.2	Extracting Slices from Traces	96
6.3	Related Work	99
7	Conclusions and Future Work	101
7.1	Conclusions	101
7.2	Future Work	104
	Bibliography	105
	A Papers of the Thesis	115
	II Papers	119