

## CONTENTS

<b>Title Page</b>	<b>ii</b>
<b>Approval Page</b>	<b>iii</b>
<b>Statement Page</b>	<b>iv</b>
<b>Dedication Page</b>	<b>v</b>
<b>Motto Page</b>	<b>vi</b>
<b>PREFACE</b>	<b>vii</b>
<b>ABSTRACT</b>	<b>xxiii</b>
<b>I INTRODUCTION</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problem Formulation . . . . .	6
1.3 Problem Limitations . . . . .	7
1.4 Research Objectives . . . . .	7
1.5 Research Advantages . . . . .	8
1.6 Research Originality . . . . .	8
1.7 Research Contributions . . . . .	9
<b>II BRIEF REVIEW OF RELATED WORKS</b>	<b>10</b>
2.1 <i>Model Checking</i> with PROMELA and SPIN . . . . .	10
2.2 <i>Java</i> and <i>Model Checking</i> . . . . .	13
2.3 Semantics Equivalence . . . . .	15
2.4 Translation and Transformation . . . . .	17
<b>III FUNDAMENTAL THEORY</b>	<b>26</b>
3.1 Source-to-source Translation . . . . .	26
3.2 PROMELA . . . . .	27
3.3 SPIN . . . . .	34
3.4 <i>Java</i> . . . . .	36
3.4.1 The Construction of <i>Java</i> Program . . . . .	36

3.4.2	Multithreading . . . . .	48
3.4.3	Threads in <i>Java</i> . . . . .	48
3.4.4	Synchronization . . . . .	50
3.5	Concurrent Execution . . . . .	52
3.6	Finite Automata . . . . .	53
3.6.1	Deterministic Finite Acceptor . . . . .	53
3.7	Program Graph . . . . .	54
3.7.1	Substatements . . . . .	55
3.8	Equivalence of Program graph . . . . .	62
3.8.1	Interleaving . . . . .	63
3.9	Handshaking . . . . .	64
3.10	Channel System . . . . .	65
<b>IV</b>	<b>RESEARCH METHOD</b>	<b>67</b>
4.1	Literature Study . . . . .	67
4.2	Defining A Construct Association . . . . .	68
4.3	Defining the Two Composition Operators . . . . .	68
4.4	Proving the Equivalence between Model and Program . . . . .	69
4.5	Developing P2J Code Translator . . . . .	69
4.6	Testing P2J Code Translator . . . . .	69
<b>V</b>	<b>CONSTRUCT ASSOCIATION</b>	<b>71</b>
5.1	The Constructs of PROMELA . . . . .	72
5.2	The Proof of Construct Association Correctness . . . . .	74
5.2.1	<b>Expression</b> . . . . .	75
5.2.2	<b>Assignment</b> . . . . .	77
5.2.3	<b>Communications</b> . . . . .	78
5.2.4	<b>Weak Atomicity</b> . . . . .	81
5.2.5	<b>Strong Atomicity</b> . . . . .	84
5.2.6	<b>Selection <i>iffi</i></b> . . . . .	88
5.2.7	<b>Deterministic repetition for</b> . . . . .	98
5.2.8	<b>Repetition - <i>dood</i></b> . . . . .	101
5.2.9	<b>Exception Handling</b> . . . . .	111
5.3	The Equivalence of PROMELA Model and <i>Java</i> Program . . . . .	114
5.4	Parallelism in PROMELA Model and <i>Java</i> Program . . . . .	122

5.4.1	Interleaving Representation in DAG . . . . .	122
<b>VI</b>	<b>P2J CODE TRANSLATOR</b>	<b>126</b>
6.1	Specification of P2J Code Translator . . . . .	126
6.2	Design of P2J Code Translator . . . . .	127
6.2.1	Procedure <i>preprocess</i> . . . . .	130
6.2.2	Procedure <i>Parse</i> . . . . .	131
6.2.3	Procedure <i>Process.parse</i> . . . . .	132
6.2.4	Procedure <i>Statement.parse</i> . . . . .	134
6.2.5	Procedure <i>Translate</i> . . . . .	140
<b>VII</b>	<b>TRANSLATION RESULT AND DISCUSSION</b>	<b>144</b>
7.1	Structure of the Result . . . . .	144
7.1.1	Class Header . . . . .	144
7.1.2	Class Channel . . . . .	149
7.2	Results of Translation . . . . .	156
7.2.1	Model for <i>Mutual Exclusion</i> . . . . .	156
7.2.2	Model of <i>Peterson</i> . . . . .	161
7.2.3	Model of Dijkstra Semaphore using Rendezvous . . . . .	162
<b>VIII</b>	<b>CONCLUSIONS</b>	<b>163</b>
	<b>REFERENCE</b>	<b>166</b>
	<b>APPENDIX</b>	<b>174</b>
<b>A</b>	<b>The List of Grouping the Statements of PROMELA</b>	<b>175</b>
A.1	<i>Metaterms</i> . . . . .	175
A.2	<i>Declarators</i> . . . . .	175
A.3	<i>Control flow</i> . . . . .	176
A.4	<i>Basic statement</i> . . . . .	177
A.5	<i>Predefined</i> . . . . .	177
<b>B</b>	<b>PROMELA Grammar Definition</b>	<b>179</b>
<b>C</b>	<b>The Tool of P2J Code Translator</b>	<b>181</b>
C.1	The Structure . . . . .	181

<b>D</b>	<b>The Complete Result of Translation</b>	<b>228</b>
D.1	Mutual Exclusion Model . . . . .	228
D.1.1	Header Implementation . . . . .	228
D.1.2	Class UserA . . . . .	229
D.1.3	Class UserB . . . . .	229
D.2	Peterson Solution Model . . . . .	230
D.2.1	Header Interface . . . . .	230
D.2.2	Header Implementation . . . . .	230
D.2.3	Class threadMX . . . . .	231
D.2.4	Class init . . . . .	232
D.3	Model Dijkstra Semaphore using Rendezvous . . . . .	233
D.3.1	Channel Launcher . . . . .	233
D.3.2	Channel Interface . . . . .	233
D.3.3	Class dijsktra . . . . .	233
D.3.4	Class user . . . . .	234
D.3.5	Class init . . . . .	235