



Universiteit
Leiden
The Netherlands

Team automata : a formal approach to the modeling of collaboration between system components

Beek, M.H. ter

Citation

Beek, M. H. ter. (2003, December 10). *Team automata : a formal approach to the modeling of collaboration between system components*. Retrieved from <https://hdl.handle.net/1887/29570>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/29570>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/29570> holds various files of this Leiden University dissertation.

Author: Beek, Maurice H. ter

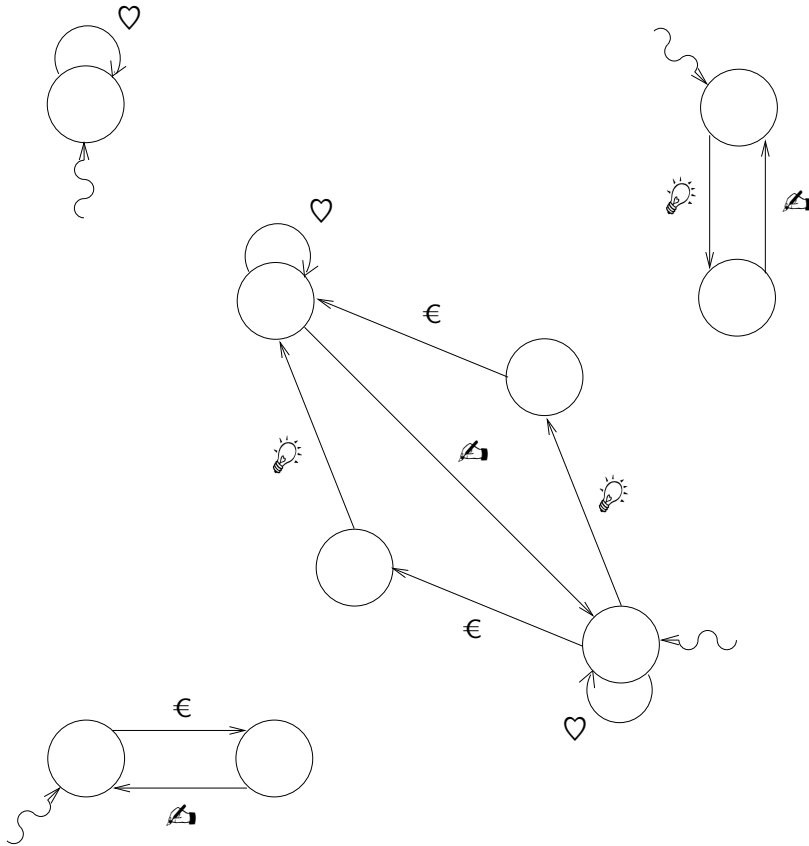
Title: Team automata : a formal approach to the modeling of collaboration between system components

Issue Date: 2003-12-10

Team Automata

A Formal Approach to the Modeling of
Collaboration Between System Components

Maurice H. ter Beek



Team Automata

A Formal Approach to the Modeling of Collaboration Between System Components

Proefschrift

ter verkrijging van

de graad van Doctor

aan de Universiteit Leiden,

op gezag van de Rector Magnificus Dr. D.D. Breimer,

hoogleraar in de faculteit der Wiskunde en

Natuurwetenschappen en die der Geneeskunde,

volgens besluit van het College voor Promoties

te verdedigen op woensdag 10 december 2003

te klokke 15.15 uur

door

Maurice Henri ter Beek

geboren te 's-Gravenhage in 1972

Promotiecommissie

Promotor: Prof.dr. G. Rozenberg

Copromotor: Dr. H.C.M. Kleijn

Referent: Prof.dr. C.A. Ellis (University of Colorado at Boulder, U.S.A.)

Overige leden: Prof.dr. Th. Bäck

Prof.dr. G. van Dijk

Prof.dr. J.N. Kok

Prof.dr. M. Koutny (University of Newcastle upon Tyne, U.K.)



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

voor John en Lia

Acknowledgments

I would never have become the person I am without the continuous and unconditional love and support of my parents... pa en ma, *bedankt!*

Part of the research for this thesis was conducted outside of Leiden, most notably in Pisa and Budapest. In Pisa I was initially hosted by Fabrizio Luccio at the Department of Computer Science of the University of Pisa and later by Stefania Gnesi at the Institute of Science and Information Technology of the National Research Council. In Budapest I was hosted by Erzsébet Csuha-Jarjú at the Computer and Automation Research Institute of the Hungarian Academy of Sciences. I am very grateful for the enduring hospitality and friendship provided by my colleagues at these institutes.

Notwithstanding my frequent trips abroad, the bulk of the research for this thesis was of course carried out in Leiden at LIACS. During all the years I spent there as a member of the Theoretical Computer Science group, my trips back to Leiden have always remained something to look forward to. For this I thank my former group members and other colleagues at LIACS.

I must admit that during the last few years the progress of my thesis has been (too) frequently the subject of conversation between me and my friends. In fact, I suspect some of them to be more relieved than me now that it is finished! But seriously, the genuine interest of my friends has always stimulated me enormously and for this I thank them all very much. I consider myself lucky to have too many friends to list them here one by one. Let me make one exception and thank Vincent for a friendship that goes beyond brotherhood.

Finally, the person that has supported me most during all the years I have worked on this thesis — and that continues to do so in my daily life — is *mijn lief*. Nadia, *ti amo!*

Table of Contents

1. Introduction	11
2. Preliminaries	23
3. Automata	29
3.1 Automata, Computations, and Behavior	29
3.2 Properties of Automata	34
3.2.1 Reduced Versions	34
3.2.2 Enabling	50
3.2.3 Determinism	55
4. Synchronized Automata	59
4.1 Definitions	59
4.1.1 Synchronized Automata	60
4.1.2 Subautomata	64
4.2 Projecting	68
4.3 Iterated Composition	74
4.4 Synchronizations	84
4.4.1 Free	85
4.4.2 Action-Indispensable	85
4.4.3 State-Indispensable	86
4.4.4 Free, Action-Indispensable, and State-Indispensable...	86
4.5 Predicates of Synchronizations	87
4.6 Effect of Synchronizations	90
4.6.1 Top-Down Inheritance of Properties	95
4.6.2 Bottom-Up Inheritance of Properties	103
4.6.3 Conclusion	110
4.7 Inheritance of Synchronizations	111
5. Team Automata	115
5.1 Definitions	115
5.1.1 Component Automata	116

5.1.2	Team Automata	118
5.1.3	Subteams	121
5.2	Iterated Composition	123
5.3	Synchronizations	126
5.3.1	Peer-to-Peer	128
5.3.2	Master-Slave	130
5.3.3	A Case Study	134
5.3.4	Peer-to-Peer and Master-Slave	137
5.4	Predicates of Synchronizations	140
5.4.1	Homogeneous Versus Heterogeneous	147
5.5	Effect of Synchronizations	149
5.5.1	Top-Down Inheritance of Properties	150
5.5.2	Bottom-Up Inheritance of Properties	153
5.6	Inheritance of Synchronizations	155
5.7	Conclusion	160
6.	Behavior of Team Automata	163
6.1	Behavior of Finite Component Automata	163
6.2	Team Behavior Versus Component Behavior	165
6.2.1	From Team Automata to Component Automata	166
6.2.2	From Component Automata to Team Automata	172
6.3	Shuffles	181
6.3.1	Definitions	182
6.3.2	Basic Observations	183
6.3.3	Commutativity and Associativity	193
6.3.4	Conclusion	205
6.4	Synchronized Shuffles	206
6.4.1	Definitions	207
6.4.2	Basic Observations	211
6.4.3	Commutativity and Associativity	215
6.4.4	Conclusion	227
6.5	Team Automata Satisfying Compositionality	228
7.	Team Automata, I/O Automata, Petri Nets	233
7.1	I/O Automata	234
7.1.1	Definitions	234
7.1.2	Iterated Composition	237
7.1.3	Synchronizations	239
7.1.4	Behavior	239
7.1.5	Conclusion	241
7.2	Petri Nets	243

7.2.1	Vector Actions and Vector Team Automata	244
7.2.2	Effect of Vector Synchronizations	249
7.2.3	Vector Controlled Concurrent Systems	251
7.2.4	Individual Token Net Controllers	254
7.2.5	Conclusion	274
8.	Applying Team Automata	277
8.1	Groupware Architectures	278
8.1.1	Team Automata as Architectural Building Blocks	278
8.1.2	GROVE Document Editor Architecture	280
8.1.3	Conclusion	282
8.2	Team-Based Model Development	283
8.2.1	A Conflict-Free Cooperation Strategy	283
8.2.2	Teams in the Conflict-Free Strategy	286
8.2.3	Teams Modeled by Team Automata	289
8.2.4	Conclusion	291
8.3	Spatial Access Control	291
8.3.1	Access Control	292
8.3.2	Authorization and Revocation	297
8.3.3	Meta Access Control	301
8.3.4	Conclusion	306
9.	Discussion	309
	Bibliography	313
	List of Figures	321
	List of Symbols	325
	Index	333

