



Universiteit
Leiden
The Netherlands

Model checking of component connectors

Izadi, M.

Citation

Izadi, M. (2011, November 6). *Model checking of component connectors*. IPA Dissertation Series. Retrieved from <https://hdl.handle.net/1887/18189>

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/1887/18189>

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

Model Checking of Component Connectors

Mohammad Izadi

Model Checking of Component Connectors

proefschrift

ter verkrijging van

de graad van Doctor aan de Universiteit Leiden,

op gezag van de Rector Magnificus prof. mr. P.F. van der Heijden,

volgens besluit van het College voor Promoties

te verdedigen op dinsdag 6 december 2011

klokke 13:45 uur

door

Mohammad Izadi

geboren te Najafabad (Iran)

in 1972

Promotiecommissie

Promotors: Prof. Dr. Farhad Arbab
Prof. Dr. Ali Movaghar (*Sharif University of Technology, Iran*)
Co-promotor: Dr. Marcello M. Bonsangue
Overige leden: Prof. Dr. Joost N. Kok
Prof. Dr. Frank de Boer
Dr. Natallia Kokash
Dr. Marjan Sirjani (*Reykjavik University, Iceland*)



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

Copyright © 2011 Mohammad Izadi
IPA Dissertation Series 2011-22

To my faithful lovely wife,
Faeze
*and **for** our beloved children,*
*our daughter, **Negar***
*and our little son, **Nikan***

Contents

1	Introduction	1
1.1	Research Context and Main Question	2
1.2	This Thesis	3
1.3	Related Work	4
1.4	Thesis Outline, Contributions, and Results	6
1.5	Research History and Publications	9
2	Context and Backgrounds	13
2.1	Component Based Systems and Coordination	14
2.2	Formal Verification and Its Methods	16
2.2.1	Deductive Verification	17
2.2.2	Model Checking	17
2.2.3	Combining Deduction and Model Checking	18
2.3	Advanced Techniques of Model Checking	19
2.3.1	Automata Theoretic Model Checking	19
2.3.2	On-the-Fly Model Checking	20
2.3.3	Symbolic Model Checking	21
2.3.4	Compositional Minimization	22
2.4	The Rules of Temporal Logics	22
3	Formal Modeling of Component Connectors	25
3.1	Reo: A Channel Based Coordination Language	26
3.1.1	Reo Primitives	26
3.1.2	Compositional Connectors	27
3.2	Basic Theory of Constraint Automata	29
3.2.1	Timed Data Streams	29
3.2.2	Constraint Automata: the Operational Semantics of Reo	31
3.2.3	Composing of Constraint Automata	33
3.3	Other Semantic Models for Reo	37
3.3.1	Co-algebraic Model of Connectors	37
3.3.2	Connector Coloring Models	38
3.3.3	Intentional Automata	38
3.3.4	Guarded and Reo Automata	40
3.3.5	Process Algebraic and Structural Operational Semantics	41

3.4	Tool Support for Reo	41
4	Fair Component Connectors	43
4.1	Introduction	44
4.2	Streams and Languages of Records	50
4.2.1	Bidirectional Translation of Record and TDS-Languages	51
4.3	Büchi Automata of Records	52
4.3.1	Büchi Automata: A Review	53
4.3.2	Büchi Automata on Streams of Records	56
4.3.3	Recasting Constraint Automata into Büchi Automata	60
4.4	Modeling Fair Reo Connectors	61
4.5	Composition of Büchi Automata of Records	62
4.5.1	Product and Join	63
4.5.2	Splitting the Join	67
4.5.3	Hiding of Port Names	70
4.6	Fair Constraint Automata	71
5	Context Dependent Connectors	75
5.1	Introduction	76
5.2	Guarded Languages and Augmented Büchi Automata	77
5.3	Modeling Reo connectors by ABARs	80
5.4	Composing ABAR Models	84
5.4.1	Product and Join	84
5.4.2	Hiding of Port Names	85
5.4.3	Splitting the join	90
5.5	Context Dependent Fair Constraint Automata	91
6	Model Checking	95
6.1	Record-based linear-time temporal logic	96
6.1.1	Some useful encodings	98
6.1.2	Specifying Reo connectors	99
6.2	From formulas to automata: model checking	101
6.3	On-the-fly translation	106
6.3.1	A description of the algorithm	106
6.3.2	The algorithm in detail	108
6.3.3	The ABAR defined by the algorithm	110
6.3.4	Proof of the correctness	111
7	A Reo Model Checker	115
7.1	Binary Decision Diagrams	116
7.2	Encoding ABARs as BDDs	119
7.2.1	Symbolic Join	122
7.3	Property Specification by BDD	125
7.4	A symbolic model checking algorithm	126
7.5	Experimental results	128

7.5.1	Dining philosophers	128
7.5.2	Mutual Exclusion	130
7.5.3	Discussion	132
8	Compositional Reduction	135
8.1	Introduction	136
8.2	Failure based equivalence of constraint automata	137
8.3	Congruency Results for Joining of Constraint Automata	140
8.4	Congruency Results for Hiding Names	144
8.5	Linear Temporal Logic of Constraint Automata	147
8.6	Reduction Algorithms	152
8.7	Compositional Model Checking	154
8.8	Case studies	157
9	Conclusions and Future Work	163
9.1	Results and Conclusions	164
9.2	Future Work	164
	Appendices	179
A	Abstract	179
B	Samenvatting (dutch)	181
C	Curriculum Vitae	183

List of Figures

3.1	Some useful channel-types in Reo	27
3.2	Exclusive router (a) and shift-lossy (b) channels designed by primitive channels of Reo [19]	29
3.3	Constraint automata for some basic channels in Reo [30]	32
3.4	Joining of constraint automata models of two FIFO1 channels	35
3.5	Hiding of port B in constraint automaton of Figure 3.4(c)	36
3.6	Intentional automaton model of a synchronous channel [52].	39
4.1	A Büchi automaton for L in Example 4.10	53
4.2	A Büchi automaton for $\bar{L}$ in Example 4.10	54
4.3	A generalized Büchi automaton with the set of accepting sets $\mathcal{F} = \{\{q_1\}, \{q_2\}\}$	55
4.4	BAR models of basic Reo channels: a) Sync channel b) SyncDrain channel, c) Filter channel, (d) ND-LossySync channel, and (e) FIFO1 channel.	57
4.5	A duplicator channel and its BAR model	58
4.6	An (unfair) merger channel and its BAR model	58
4.7	Two visibly equivalent Büchi automata of records.	59
4.8	Models of a <i>non-deterministic</i> lossy synchronous channel by a) a constraint automaton and b) a Büchi automaton of records.	60
4.9	Models of a fair non-deterministic lossy synchronous channel with a) a weak fairness condition, b) a strong fairness condition.	62
4.10	Models of a merger connector: (a) unfair version, (b) fair version	63
4.11	Composing two FIFO1 channels	65
4.12	Direct and indirect joining of two FIFO1 buffers	68
4.13	The resulting BAR after hiding B in Figure 4.12(e).	71
4.14	The resulting BAR after eliminating τ -transitions in Figure 4.13.	71
5.1	A BAR model of a FIFO2 channel and its canonical ABAR.	80
5.2	Three ABAR models of the context dependent lossy synchronous channel	82
5.3	The ABAR model of a fair closed system of a context dependent lossy synchronous channel and its environment	83
5.4	Models for a Reo synchronous channel (Sync) from source node B to sink C : (a) Its BAR model; (b) The canonical ABAR model for (a); and (c) The more explicit ABAR model.	83

5.5	The composition of the ABAR models of a context dependent lossy synchronous channel and a synchronous channel	86
5.6	The composition of two context dependent lossy synchronous channels. . . .	87
5.7	The composition of a context dependent lossy synchronous channel with a FIFO1 channel.	88
5.8	The composition of a FIFO1 channel with a context dependent lossy synchronous channel.	89
5.9	The composition of a synchronous channel with a FIFO1 channel.	89
5.10	The composition of a FIFO1 with a synchronous channel.	90
5.11	Direct and indirect joining of two FIFO1 buffers modeled by ABARs	91
6.1	ABAR models of some basic Reo connectors: (a) Sync channel, (b) Context-Dependent LossySync channel, and (c) FIFO1 channel.	100
7.1	Binary decision tree for switching function $f = z_1 \wedge (\neg z_2 \vee z_3)$ [29].	117
7.2	Binary decision diagram for switching function $f = z_1 \wedge (\neg z_2 \vee z_3)$ [29]. . .	118
7.3	A synchronous channel and its ABAR model	120
7.4	BDD representation of a synchronous channel: (a) ports, (b) states, initial states, final states and (c) transition relation.	121
7.5	(a) FIFO1 channel, (b) its ABAR model, and BDD representation of (c) ports and states, (d) initial states and final states and (e) transition relation.	123
7.6	(a) Join of a synchronous channel and a FIFO1 channel, (b) its ABAR and BDD representation of (c) ports, (d) states, initial states, final states and (e) transition relation.	124
7.7	(a) A Büchi automaton and (b) an ABAR for $\langle r \rangle (A \wedge B)$	126
7.8	BDD representation for the ABAR equivalent of $\langle r \rangle (A \wedge B)$ (a) states, (b) initial states, (c) final states, and (d) transition relation.	127
7.9	ABAR models of some Reo channels where $\mathcal{D} = \{d\}$	129
7.10	Coordination pattern for two philosophers in the dining philosophers problem	130
7.11	Behavior of a philosopher in ABAR terms	130
7.12	Coordination pattern for two processes in mutual exclusion for $k = 1$	131
7.13	Behavior of a process in ABAR terms	132
8.1	(a) Dining philosophers scenario in Reo and (b) a chopstick, (c) minimized constraint automaton for a chopstick and (d) a philosopher	155
8.2	(a) A resource allocation system, (b) constraint automaton model of a process, (c) constraint automaton model of the coordinator	157
8.3	Inres protocol architecture (the connectors are Reo primitive channels)	158

List of Tables

6.1	Definitions of New_1 , New_2 and $Next_1$ functions.	108
7.1	State space generation results for the dining philosophers problem	129
7.2	Model checking time (sec) for n dining philosophers	131
7.3	State space generation results for the mutual exclusion problem	132
7.4	Model checking time (sec) for the mutual exclusion problem	132
8.1	Number of reachable states for the Inres protocol system.	159
8.2	Number of reachable states for the resource allocation system.	160

