

BISIMULATIONS AND PREBISIMULATIONS FOR TIMED AUTOMATA

SHIBASHIS GUHA



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
DECEMBER 2016**

BISIMULATIONS AND PREBISIMULATIONS FOR TIMED AUTOMATA

by

SHIBASHIS GUHA

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the



**Indian Institute of Technology Delhi
DECEMBER 2016**

Certificate

This is to certify that the thesis titled **Bisimulations and Prebisimulations over Timed Automata** being submitted by **Shibashis Guha** for the award of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of bona fide work carried out by him under my guidance and supervision at the **Department of Computer Science and Engineering, Indian Institute of Technology Delhi**. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

S. Arun-Kumar

Professor

Department of Computer Science
and Engineering

Indian Institute of Technology Delhi
New Delhi, India 110 016

Acknowledgments

Writing the acknowledgement is possibly one of the most challenging tasks since this is the part of the thesis that finds the maximum readership. I take this opportunity to thank my advisor Prof. S. Arun-Kumar who allowed me to pursue my doctoral research under his supervision. It is not only that I learnt various technical skills and subtleties from him but his perspective on research and academia has broadened my horizon and helped me recognize academics and research as a way of life. He has always been very supportive and liberal about letting me choosing my research problems and towards my visits and internships in other institutes and working with other collaborators.

Chinmay Narayan was a good friend since my M.Tech days. We joined the PhD programme in IIT Delhi on the same day. He was always there as a wall, on whom I could bounce my ideas and in return get some excellent feedback. His immaculate style of formalizing things helped me learn a great deal. With him I could possibly talk about anything and everything. I would remember him as someone who would constantly encourage me during those bad days when things appeared to move very slowly and sometimes even refused to make any progress. I would also remember him as a friend who likes singing in a way that warrants that nobody would like to hear him twice.

I thank Prof. Srinath Srinivasa who was my research advisor during my M.Tech at IIIT Bangalore. His students, including me, used to look up to him as a source of inspiration. His constant encouragement towards solving problems introduced me to research and laid the foundation for joining the PhD programme at a later stage.

It was exciting to work with Mihir Mehta on the ReLTS project. I had a wonderful time discussing with him the design and the implementation of the tool. In the verification lab in IIT Delhi, I was fortunate to find Divyanshu with whom I spent hours discussing problems. He had been very helpful in reviewing some of the proofs in this thesis. Also my sincere thanks to Chandrika for giving me valuable feedback while preparing several of my presentations that I made in the course of my PhD.

I am indebted to Prof. Krishna S. of IIT Bombay, who has been very kind to allow me to visit Centre for Formal Design and Verification of Software (CFDVS) at IIT Bombay as and when I wanted to. I also thank Prof. Ashutosh Trivedi of IIT Bombay with whom I had several insightful discussions. Many of these discussions with Prof. Krishna S. and Prof. Ashutosh Trivedi introduced me to various interesting research problems and the current research trends that are aligned with my research interests. I must also mention Prof. Supratik Chakraborty's support for arranging my visit to IIT Bombay on the very first occasion. Without that the successive visits would not have been possible.

I thank Dr. Nathalie Bertrand and Dr. Frederic Herbreteau for hosting me during my visit to INRIA at Rennes and LaBRI at Bordeaux respectively in France. I had a very fruitful time while working with both of them during that visit.

My sincere thanks to Prof. Sanjiva Prasad, Prof. Astrid Kiehn and Prof. Kolin Paul for agreeing to be in my doctoral research committee and providing me with valuable feedback in the course of my PhD. I also thank Prof. Saroj Kaushik, the head of the Department of Computer Science and Engineering at IIT Delhi for her support towards procuring laptop and other grants that made things easier and free from hassles.

Paper work was made a lot easier by the Department staff: Mrs. Renu Bhatia, Mr. Rathinam, Rekha and Hemant. I also thank Mr. Rajinder of the PG section and Mr. Chopra of the accounts section for facilitating paper work on several occasions. Mr. Chandu and Mr. KK, the CFDVS staff at IIT Bombay have also been very kind in terms of the logistics support that I needed. Further, I thank Microsoft Research India for supporting my PhD work through their generous fellowship.

Life in IIT would have been less fun in the absence of Syamantak's jokes, his compulsive laughs, our news gazette Gayathri, Shashank's recommendation of Mehdi Hassan ghazals and the Martin Scorsese movies and intruding into Anamitra's home and ensuring that he gets enough food for us. I also thank Suvam, Brojeshwar, Nisha, Dipanjan, Anirban, Swati, Manoj, Yamuna, Sharat, Namita, Sayak, Anup, Mona, Rudra, Jatin, Arindam,

Sandeep, Rajshekar, Prathamesh, Madhulika, Aditi and Parul for sharing some of the nice moments that I would cherish for long. At IIT Bombay, Hrishikesh, Ramchandra, Shahan, Abhisekh and in France, Paulin, Srinivas, Yogesh and Sagnik were always around whenever there was a need, however small it may be. I thank them all.

Last but not the least, I thank my parents for their enormous support, love and encouragement towards my education and infusing me with all the values, patience and courage that one needs to face various challenges in the journey called life and more specifically in the course of PhD. Their constant endeavour to provide me with the best education had enabled me to enroll in my doctoral studies that culminated in this thesis. I dedicate this thesis to them.

Shibashis Guha

Abstract

Timed automata, introduced in [1] is a popular formalism for modelling and analysing real time systems. During the last two decades, significant progress has been made in both theory and practice on this formalism. Control state reachability in timed automata was proved to be decidable and is PSAPCE-complete [1]. Several tools have been developed for verification of timed automata. Among them UPPAAL [2, 3], KRONOS [4] and CMC [5] are worth mentioning. Several protocols, controllers and real time algorithms have been verified and industrial case studies have been made using these tools [6, 7, 8, 9]. Though timed language equivalence is known to be undecidable for timed automata [1], timed bisimulation equivalence and time abstracted bisimulation equivalence have been proved to be decidable for timed automata [10, 11].

Speed dependent preorders based on bisimulation have been defined in process algebra frameworks to compare the responsiveness of two functionally equivalent systems. A faster than precongruence, based on a timed extension to Milner’s Calculus for Communicating Systems (CCS) [12], has been defined in [13]. The relative efficiency of two processes has been defined based on the number of internal actions performed by each of them in the paper [14]. Performance preorders based on durational actions [15, 16] have also been studied in [17].

In this thesis, we define an ‘at least as fast as’ relation between two timed automata. The relation is based on the notion of bisimulations and prebisimulations. Prebisimulation for timed automata is an unexplored area and to the best of our knowledge, this is the first such instance where decidability of prebisimulation over timed automata is studied. We show the decidability of the timed performance prebisimulation relation for one clock timed automata using a zone graph [18, 19] construction in PTIME. A zone graph induces a coarser abstraction of the state space of the timed automata compared to the more traditional region graph [1] construction. Thus the construction of a zone graph in general leads to a smaller arena for verification and analysis of timed automata.

In the thesis, we show that our construction leads to a unifying approach to decide various equivalence and preorder relations for timed automata. The method also leads to a game characterization of the relations where the game is played on the zone graphs of the two timed automata being compared. The game characterization further produces an infinite game hierarchy that leads to defining an infinite hierarchy of timed relations. Corresponding to the game for timed bisimulation, we also show a method to generate a timed Hennessy-Milner logic [20, 21] based distinguishing formula if the two timed automata are not timed bisimilar.

Checking any timed and time abstracted relation over timed automata, analysis and model checking of timed automata involve construction of a zone graph or the region graph of the timed automata. The size of the zone graph or the region graph is exponential in the number of the clocks of the timed automaton [1]. Hence it is desirable to have a timed automaton with the least possible number of clocks. While the problem of checking whether a timed automaton accepting the same timed language but with fewer clocks exists is known to be undecidable [22], in this thesis, we have described a method, which given a timed automaton A , produces another timed automaton with the smallest number of clocks that is timed bisimilar to A .

Contents

List of Figures	xix
------------------------	------------

List of Tables	xxiii
-----------------------	--------------

1 Introduction	1
1.1 Labelled Transition Systems, Behavioural Equivalences and Preorders . .	3
1.2 Real Time Models and Timed Automata	6
1.3 Clock Reduction	10
1.4 Organization	11
2 Technical Preliminaries	15
2.1 Timed Automata	15
2.1.1 Definitions	16
2.1.2 Semantics	20
2.1.3 Region Automata	23
2.2 Representation of zones	29
2.3 Network of timed automata	32
2.4 Conclusion	35
3 Construction of Zone Graph	37
3.1 Definitions	37
3.2 Location Dependent Maximal Constants Abstraction	39
3.3 Construction of a pre-stable zone graph	42
3.3.1 Creating pre-stable zones	45
3.3.2 Extended Zones	50
3.4 Conclusion	52

4	Deciding Timed and Time Abstracted Relations	53
4.1	Introduction	53
4.2	Equivalences and Preorders for Timed Systems	55
4.3	Strong Bisimulation and its Game Characterization	59
4.4	Game Characterization of Timed Bisimilarity	65
4.4.1	Corner point bisimulation as a game	69
4.4.2	Equivalence between timed bisimulation and corner point bisimulation	77
4.4.3	Synthesis of Distinguishing Formulae	92
4.5	Deciding Time Abstracted Bisimulation	97
4.6	ReLTS: A tool for deciding relations	100
4.6.1	Architecture of ReLTS	101
4.6.2	Example	103
4.6.3	Algorithm	104
4.6.4	Benchmarks	107
4.7	Conclusion	109
5	Timed Performance Prebisimulation	111
5.1	Introduction	111
5.2	Timed Performance Prebisimulation	114
5.3	Closure under parallel composition	115
5.4	Corner Point Prebisimulation	118
5.4.1	Some examples of cp-prebisimulation game	121
5.4.2	Deciding corner point prebisimulation and timed performance prebisimulation	124
5.4.3	The abstraction operation	141
5.5	A non-trivial example	144
5.5.1	Alternating bit protocol	145

5.5.2	Stop-and-Wait ARQ	150
5.5.3	ABP $\preceq$ Stop-and-Wait ARQ	152
5.6	Conclusion	155
6	Game Characterization	157
6.1	Introduction	157
6.2	Game Template	159
6.3	Hierarchy of Timed Games	161
6.3.1	Infinite Game Hierarchy	168
6.4	Conclusion	169
7	Clock Reduction while Preserving Timed Bisimulation	171
7.1	Introduction	171
7.2	Clock Reduction	173
7.3	Conclusion	200
8	Conclusion	203
	Bibliography	209
	Appendix	222
A	List of symbols	223
B	Acronyms	229
	Index	231

List of Figures

1.1	A timed automaton modelling an assembly line component spraying colour on the chassis of an automobile	3
2.1	An example timed automaton	18
2.2	A timed automaton with three clocks	19
2.3	Timed automata with and without invariant	22
2.4	Regions corresponding to $M_x = 3$ and $M_y = 2$	24
2.5	A timed automaton with $M_x = 1$ and $M_y = 1$	25
2.6	numbered regions for $M_x = 1$ and $M_y = 1$	25
2.7	A timed automaton with three clocks	26
2.8	Infinite zone graph without abstraction	29
2.9	(a) timed automaton modelling mouse click (b) timed automaton modelling a user behaviour	33
2.10	A small part of the TLTS for the network of the two automata shown in Figure 2.9	33
3.1	An example to illustrate location dependent maximal abstraction	41
3.2	A timed automaton and the zones for location l_1	44
3.3	For every zone, there exists a hyperplane that bounds it fully from above in a pre-stable zone graph	49
4.1	Strategies of challenger and defender	62
4.2	An example of a strong bisimulation game with four rounds and three alternations	63
4.3	An example of a strong bisimulation game with five rounds and four alternations	64
4.4	s_1 is strongly bisimilar to t_1	64

4.5	A cp-delay from p to p'	68
4.6	An example of a cp-timed bisimulation game. A distinguishing formula is also generated while playing the game if the two timed automata are not cp-bisimilar	72
4.7	Checking cp-bisimulation using zone graph: A' and its zone graph	73
4.8	Checking cp-bisimulation using zone graph: B' and its zone graph	74
4.9	Checking cp-bisimulation using zone graph	76
4.10	Examples illustrating hyperplane bisimilarity	80
4.11	Action a is enabled by the hyperplane $x = 2$ and disabled by the hyper- plane $x = 5$	80
4.12	$\hat{H}$ is reached after H'_{n+1} from q_{2n}	84
4.13	Timed automata states with the delays used in the proof of Lemma 4.4.17	89
4.14	Components of ReLTS	102
4.15	A simple example	103
4.16	Internals of <code>get_distinguishing_formula</code>	103
4.17	finite zone graph after abstraction for timed automaton shown in Figure 2.8(a)	108
5.1	Example: A preorder relation	112
5.2	(a) $A_1 \not\lesssim B_1$ (b) $A_2 \lesssim B_2$	115
5.3	$A_1 \lesssim A_2$ but $A_1 \parallel B \not\lesssim A_2 \parallel B$	116
5.4	$A \lesssim B$ but $A \not\sim_t B$	118
5.5	An equal delay made by defender when challenger moves to $(l_0, x =$ $3 - \delta)$ from the initial state	120
5.6	Examples showing defender may move to entry as well as exit corner point when challenger reaches an exit corner point	120
5.7	Example of corner point prebisimulation game	121
5.8	Two timed automata that are not corner point prebisimilar	123

5.9	Comparing corner point delays	124
5.10	Lemma 5.4.3 does not hold for timed automata with two clocks	140
5.11	Timed automaton for sender in ABP	146
5.12	Timed automaton for channel	147
5.13	Timed automaton for ABP receiver	148
5.14	Part of zone graph corresponding to the timed automaton of the ABP sender	149
5.15	Part of the abstracted zone graph corresponding to the timed automaton of the ABP sender in parallel composition	150
5.16	Timed automaton for sender in ARQ	151
5.17	Timed automaton for receiver in ARQ	152
5.18	Part of the zone graph corresponding to timed automaton of the ARQ sender	153
5.19	Part of the abstracted zone graph corresponding to timed automaton of the ARQ sender in parallel composition	154
6.1	The hierarchy of the timed relations and the corresponding hierarchy of their game characterizations	161
6.2	Time abstracted bisimulation game	164
6.3	Time abstracted delay bisimulation game	164
6.4	Time abstracted observational bisimulation game	166
6.5	Time abstracted observational bisimulation game on an LTS	167
6.6	Relations over timed automata, game characterization and the infinite hi- erarchy of timed games	169
7.1	A timed automaton and the zones for location l_1	175
7.2	Zone Graph for the TA in Figure 7.1	175
7.3	Splitting locations of the TA in Figure 7.1	176
7.4	The two timed automata are timed bisimilar	176
7.5	Merging constraints in stage 3	184

7.6	Colouring clock graph	184
7.7	(a) a graph G (b) TA with clock graph same as G	194

List of Tables

4.1	Rounds of the strong bisimulation game for the example shown in Figure	
4.2		63
4.2	Rounds of the strong bisimulation game for the example shown in Figure	
4.6		73
4.3	All durations in seconds	107
5.1	Moves of the cp-prebisimulation game corresponding to automata shown	
	in Figure 5.7 when challenger chooses to play from the p side in each	
	round, when it is checked if $p \lesssim_{cp} q$	122
5.2	Moves of the cp-prebisimulation game corresponding to automata shown	
	in Figure 5.7 when challenger chooses to play from the q side in each	
	round, when it is checked if $p \lesssim_{cp} q$	122