Lecture Notes in Computer Science 9223

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

More information about this series at http://www.springer.com/series/7407

Frank Drewes (Ed.)

Implementation and Application of Automata

20th International Conference, CIAA 2015
Umeå, Sweden, August 18–21, 2015
Proceedings

Springer

Editor
Frank Drewes
Umeå University
Umeå
Sweden

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-22359-9 ISBN 978-3-319-22360-5 (eBook)
DOI 10.1007/978-3-319-22360-5

Library of Congress Control Number: 2015945122

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

This volume contains the papers presented at the 20th International Conference on Implementation and Application of Automata (CIAA 2015), which was organized by the Department of Computing Science at Umeå University, Sweden, and took place at Umeå Folkets hus during August 18–21, 2015.

The CIAA conference series is the major international venue for the dissemination of new results in the implementation, application, and theory of automata. The previous 19 conferences were held in various locations all around the globe: Blois (2011), Giessen (2014), Halifax (2013), Kingston (2004), London Ontario (WIA 1997, WIA 1996, and 2000), Nice (2005), Porto (2012), Potsdam (WIA 1999), Prague (2007), Pretoria (2001), Rouen (WIA 1998), San Francisco (2008), Santa Barbara (2003), Sydney (2009), Taipei (2006), Tours (2002), and Winnipeg (2010).

The topics of this volume include cover automata, counter automata, decision algorithms on automata, descriptional complexity, expressive power of automata, homing sequences, jumping finite automata, multidimensional languages, parsing and pattern matching, quantum automata, realtime pushdown automata, random generation of automata, regular expressions, security issues, sensors in automata, transducers, transformation of automata, and weighted automata.

In total, 49 papers were submitted by authors in 20 different countries: Brazil, Canada, Czech Republic, Finland, France, Germany, Hungary, India, Israel, Italy, Japan, South Korea, Norway, Poland, Portugal, Russia, South Africa, Sweden, the UK, and the USA. Each of these papers was reviewed by at least three reviewers and thoroughly discussed by the Program Committee, which resulted in the selection of 22 papers for presentation at the conference and publication in this volume. Four invited talks were given by Benedikt Bollig, Christof Löding, Andreas Maletti, and Bruce Watson. In addition to these contributions, the volume contains two short papers about tool demonstrations that were given at the conference.

I am very thankful to all invited speakers, authors of submitted papers, system demonstrators, Program Committee members, and external reviewers for their valuable contributions and help. Without them, CIAA 2015 could not have been realized. The entire process from the original submissions to collecting the final versions of papers was greatly simplified by the use of the EasyChair conference management system.

I would furthermore like to thank the editorial staff at Springer, and in particular Alfred Hofmann and Anna Kramer, for their guidance and help during the process of publishing this volume, and Camilla Andersson at the conference site Umeå Folkets hus for her help with all the practical preparations.

CIAA 2015 was financially supported by (a) the Department of Computing Science at Umeå University, (b) the conference fund of Umeå Municipality, the County Council of Västerbotten and Umeå University, (c) the Faculty of Science and Technology at Umeå University, and (d) the Swedish Research Council, who provided generous funding for invited speakers.

Last but by no means least, I wish to thank the local Organizing Committee consisting of the members of the research group Foundations of Language Processing, namely, Suna Bensch, Henrik and Johanna Björklund, Loek Cleophas, Petter Ericson, Yonas Woldemariam, and Niklas Zechner for their help.

We are now looking forward to CIAA 2016 at Yonsei University, Seoul, in South Korea.

August 2015 Frank Drewes

Organization

CIAA 2015 was organized by the Department of Computing Science at Umeå University, Sweden, and took place at Umeå Folkets hus.

Invited Speakers

Benedikt Bollig	Université Paris-Saclay, France
Christof Löding	RWTH Aachen, Germany
Andreas Maletti	University of Stuttgart, Germany
Bruce Watson	University of Stellenbosch, South Africa

Program Committee

Parosh Aziz Abdulla	Uppsala University, Sweden
Marie-Pierre Beal	Université Paris-Est Marne-la-Vallée, France
Cezar Câmpeanu	University of Prince Edward Island, Canada
Pascal Caron	Université de Rouen, France
Jean-Marc Champarnaud	Université de Rouen, France
David Chiang	University of Notre Dame, USA
Stefano Crespi-Reghizzi	Politecnico di Milano, Italy
Jürgen Dassow	Otto von Guericke University Magdeburg, Germany
Frank Drewes	Umeå University, Sweden
Rudolf Freund	TU Wien, Austria
Yo-Sub Han	Yonsei University, South Korea
Markus Holzer	Justus Liebig University Giessen, Germany
Oscar Ibarra	University of California, Santa Barbara, USA
Helmut Jürgensen	The University of Western Ontario, Canada
Martin Kutrib	Justus Liebig University Giessen, Germany
Andreas Maletti	University of Stuttgart, Germany
Sebastian Maneth	University of Edinburgh, UK
Wim Martens	Bayreuth University, Germany
Denis Maurel	Université François Rabelais Tours, France
Carlo Mereghetti	Università degli Studi di Milano, Italy
Brink van der Merwe	University of Stellenbosch, South Africa
Cyril Nicaud	Université Paris-Est, France
Alexander Okhotin	University of Turku, Finnland
Daniel Reidenbach	Loughborough University, UK
Rogerio Reis	Universidade do Porto, Portugal
Kai T. Salomaa	Queen's University, Canada
Klaus Sutner	Carnegie Mellon University, USA
Sophie Tison	Université de Lille 1, France
György Vaszil	University of Debrecen, Hungary

Heiko Vogler	TU Dresden, Germany
Hsu-Chun Yen	National Taiwan University, Taiwan

Steering Committee

Jean-Marc Champarnaud	Université de Rouen, France
Markus Holzer	Justus Liebig University Giessen, Germany
Oscar Ibarra	University of California, Santa Barbara, USA
Denis Maurel	Université François Rabelais Tours, France
Kai T. Salomaa	Queen's University, Canada
Hsu-Chun Yen	National Taiwan University, Taiwan

External Reviewers

Cyriac Aiswarya
Mohamed Faouzi Atig
Johanna Björklund
Henrik Björklund
Benedikt Bollig
Sabine Broda
Cezar Campeanu
Arnaud Carayol
Yi-Jun Chang
Christian Choffrut
Julien Clément
Maxime Crochemore
Wojciech Czerwiński
Julien David
Toni Dietze
Francesco Dolce
Mike Domaratzki
Marianne Flouret
Travis Gagie
Lukas Holik
Sebastian Jakobi
Artur Jeż
Sang-Ki Ko
Marco Kuhlmann
Ondrej Lengal
Lvzhou Li
Sylvain Lombardy
Eva Maia
Andreas Malcher
Luca Manzoni
Tomas Masopust
Ian McQuillan
Katja Meckel
Ludovic Mignot
Nelma Moreira
Timothy Ng
Kim Nguyen
Florent Nicart
Damien Nouvel
Faissal Ouardi
Balasubramanian Ravikumar
Klaus Reinhardt
Martin Schuster
Shinnosuke Seki
Jari Stenman
Till Tantau
Bianca Truthe
Vojtěch Vorel
Johannes Waldmann
Matthias Wendlandt
Lynette Van Zijl

Organizing Committee

Suna Bensch
Henrik Björklund
Johanna Björklund
Frank Drewes
Loek Cleophas
Petter Ericson
Yonas Woldemariam
Niklas Zechner

Sponsoring Institutions

Department of Computing Science, Umeå University
Faculty of Science and Technology, Umeå University
Umeå Municipality, the County Council of Västerbotten and Umeå University
The Swedish Research Council

Invited Papers

Automata and Logics for Concurrent Systems: Five Models in Five Pages

Benedikt Bollig

LSV, ENS Cachan, CNRS & Inria
bollig@lsv.ens-cachan.fr

Abstract. We survey various automata models of concurrent systems and their connection with monadic second-order logic: finite automata, class memory automata, nested-word automata, asynchronous automata, and message-passing automata.

Resource Automatic Structures for Verification of Boundedness Properties

Extended Abstract

Christof Löding

RWTH Aachen, Germany
loeding@cs.rwth-aachen.de

Automatic structures are (possibly infinite) structures that can be represented by means of finite automata [1, 10]. The elements of the domain of the structure are encoded as words and form a regular language. The relations of the structure are recognized by synchronous automata with several input tapes (the number of the tapes corresponding to the arity of the relation). A typical example of such a structure is $(\mathbb{N}, +, <)$, the natural numbers with addition and order. The natural numbers are encoded by words corresponding to their binary representation (or any other base). The order and the addition (as ternary relation) can then be accepted by synchronous automata with the corresponding number of input tapes.

Another class of examples are configuration graphs of pushdown automata with reachability relation. The vertices of a pushdown graph are naturally encoded as words (a control state followed by a stack content). The set of reachable configurations (from the initial configuration) forms a regular language [4], and more generally, the reachability relation is automatic, that is, there is a finite two-tape automaton that accepts those pairs of configurations such that the second one is reachable from the first one [7].[1]

Automatic structures are interesting in verification because their first-order theory (FO) is decidable: the atomic formulas are already given by automata, and the closure properties of finite automata can be used for an inductive translation of composed formulas.

In [13] we have introduced the notion of resource automatic structures. In this model of resource structures, a relation is not a set of tuples but a function that assigns to each tuple a natural number or ∞, where the value ∞ corresponds to the classical case of not being in the relation. A value n for a tuple can be seen as a cost for being in the relation.

As an illustration, we extend the above example of pushdown graphs with reachability relation: Assume that the transitions of the pushdown system are annotated with operations on resources. For each type of resource, a transition can either consume one unit of this resource, completely replenish the resource at once, or not use the resource at all. Then we can associate the cost of a finite path through the pushdown graph to be the maximal number of units consumed from a resource without being replenished in

[1] The result in [7] is for the more general case of ground term rewriting systems, which include pushdown automata as special case.

between. This corresponds to the size of the reservoir required for the resource to execute the path. We naturally obtain a resource relation that assigns to each pair of configurations the cost of a cheapest path between these two configurations (and ∞ if there is no path between the configurations).

In [13] it is shown that this resource reachability relation for pushdown automata can be defined by an automaton model called B-automata. The transitions of these automata are annotated by actions on counters that either increment the counter, reset the counter to value 0, or leave the counter unchanged. In this way, a cost is assigned to each input word that is accepted by the automaton as follows. The cost of a run is the maximal value that one of the counters assumes during the run. The cost of the input word is the minimal cost of an accepting run for this input word (and ∞ if the word is not accepted).

The class of resource automatic structures [13] is defined to be the class of resource structures that can be encoded by B-automata, thus pushdown graphs with the resource reachability relation are resource automatic structures.

As logic over these structures, we consider FO+RR, first-order logic with resource relations, which is standard FO logic without negation. Similar to the resource relations, a formula of FO+RR has a value (instead of being true or false). Intuitively, this value corresponds to the cost for making this formula true: the value of the atomic formulas is given by the resource relations, disjunction and conjunction are translated to min and max, and existential and universal quantifiers are translated to inf and sup.

The intention of this logic is to be able to formalize and solve boundedness properties for resource structures. Taking again the example of pushdown graphs with resource reachability, a typical question would be the bounded reachability problem: given two regular sets A, B of configurations, does there exist a bound K such that from each configuration in A there is a path to a configuration in B with cost at most K. The corresponding formula is

$$\forall x \in A \exists y \in B \ : \ x \rightarrow^* y$$

where $\rightarrow^*$ is the resource reachability relation. According to the semantics of FO+RR, the value of the formula is not ∞ if, and only if, the above bounded reachability property holds.

Similar to the translation of classical FO formulas over automatic structures into finite automata, FO+RR formulas can be translated into B-automata preserving boundedness [13] (using the closure of B-automata under the operations max, min, inf, sup). Thus, for deciding whether the value of a formula is finite, it suffices to check the boundedness property on B-automata.

Boundedness properties for finite automata have been studied in the context of the star-height problem for regular expressions [8, 11]. Given a regular language and a number h, the question of whether there exists a regular expression of star-height at most h for this language, can be reduced to a boundedness question of B-automata. The boundedness (or limitedness) problem for B-automata is the question whether there is a bound on the cost of the accepted words. It is shown to be decidable in [11], where the automata are called distance-desert automata. The name of B-automata originates from a model introduced in [3] for describing boundedness properties of infinite words. Based on the decidability results for B-automata, one obtains the decidability of the boundedness problem for FO+RR formulas over resource automatic structures.

In [14] the class of resource automatic structures is studied in more detail. It is shown that there is a complete resource automatic structure (each other resource automatic structure can be obtained from this complete structure by interpretations in $\mathsf{FO{+}RR}$ logic). Furthermore, connections between $\mathsf{FO{+}RR}$ over resource automatic structures and cost monadic second-order logic (cost MSO) [5] and cost FO [12] over words are established that generalize the standard setting over words without costs.

The model of B-automata and the corresponding decidability results can be extended to finite trees [6], which leads to the class of resource tree automatic structures. In recent work [9], it is shown that an extension of $\mathsf{FO{+}RR}$ with an operator for testing boundedness of formulas, can be used to capture weak cost MSO [5] and weak MSO+U [2], obtaining alternative proofs for the decidability of these logics.

References

1. Blumensath, A., Grädel, E.: Automatic structures. In: Proceedings of the 15th IEEE Symposium on Logic in Computer Science, LICS 2000, pp. 51–62. IEEE Computer Society Press (2000)
2. Bojańczyk, M.: Weak MSO with the unbounding quantifier. Theory Comput. Syst. **48**(3), 554–576 (2011). http://dx.doi.org/10.1007/s00224-010-9279-2
3. Bojańczyk, M., Colcombet, T.: Bounds in w-regularity. In: 21th IEEE Symposium on Logic in Computer Science (LICS 2006), Seattle, WA, USA, Proceedings, 12–15 August 2006, pp. 285–296. IEEE Computer Society (2006)
4. Büchi, J.R.: Regular canonical system. Archiv für Mathematische Grundlagenforschung **6**, 91–111 (1964)
5. Colcombet, T.: Regular cost functions, part I: logic and algebra over words. Log. Methods Comput. Sci. **9**(3) (2013). http://dx.doi.org/10.2168/LMCS-9(3:3)2013
6. Colcombet, T., Löding, C.: Regular cost functions over finite trees. In: Twenty-Fifth Annual IEEE Symposium on Logic in Computer Science, LICS 2010, pp. 70–79. IEEE Computer Society (2010)
7. Dauchet, M., Tison, S.: The theory of ground rewrite systems is decidable. In: Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science, LICS 1990, pp. 242–248. IEEE Computer Society Press (1990)
8. Hashiguchi, K.: Algorithms for determining relative star height and star height. Inf. Comput. **78**(2), 124–169 (1988)
9. Kaiser, Ł., Lang, M., Löding, C., Leßenich, S.: A unified approach to boundedness properties in MSO (2015, submitted)
10. Khoussainov, B., Nerode, A.: Automatic presentations of structures. In: Leivant, D. (ed.) LCC 1994. LNCS, vol. 960, pp. 367–392. Springer, Heidelberg (1995)
11. Kirsten, D.: Distance desert automata and the star height problem. RAIRO – Theor. Inf. Appl. **3**(39), 455–509 (2005)
12. Kuperberg, D.: Study of classes of regular cost functions. Ph.D. thesis, LIAFA Paris, December 2012
13. Lang, M., Löding, C.: Modeling and verification of infinite systems with resources. Log. Methods Comput. Sci. **9**(4) (2013)
14. Lang, M., Löding, C., Manuel, A.: Definability and transformations for cost logics and automatic structures. In: Csuhaj-Varjú, E., Dietzfelbinger, M., Ésik, Z. (eds.) MFCS 2014, Part I. LNCS, vol. 8634, pp. 390–401. Springer, Heidelberg (2014). http://dx.doi.org/10.1007/978-3-662-44522-8

Finite-State Technology in Natural Language Processing
Extended Abstract

Andreas Maletti

Institute for Natural Language Processing, Universität Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
maletti@ims.uni-stuttgart.de

Finite-state technology is at the core of many standard approaches in natural language processing [11, 15]. However, the terminology and the notations differ significantly between theoretical computer science (TCS) [8] and natural language processing (NLP) [13]. In this lecture, inspired by [11, 13], we plan to illustrate the close ties between formal language theory as discussed in TCS and its use in mainstream applications of NLP. In addition, we will try to match the different terminologies in three example tasks. Overall, this lecture shall serve as an introduction to (i) these tasks and (ii) the use of finite-state technology in NLP and shall encourage closer collaboration between TCS and NLP.

We will start with the task of part-of-speech tagging [11, Chapter 5], in which given a natural language sentence the task is to derive the word category (the part-of-speech, e.g. noun, verb, adjective, etc.) for each occurring word in the sentence. The part-of-speech information is essential for several downstream applications like co-reference resolution [11, Chapter 21] (i.e., detecting which entities in a text refer to the same entities), automatic keyword detection [11, Chapter 22] (i.e., finding relevant terms for a document), and sentiment analysis [18] (i.e., the process of determining whether a text speaks favorably or negatively about a subject). Along the historical development of systems for this task [9] we will discuss the main performance breakthrough (in the mid 80s) that led to the systems that are currently state-of-the-art for this task. This breakthrough was achieved with the help of statistical finite-state systems commonly called *hidden Markov models* [11, Chapter 6], which roughly equate to probabilistic finite-state transducers [17]. We will outline the connection and also demonstrate how various well-known algorithms like the forward and backward algorithms relate to TCS concepts.

Second, we will discuss the task of parsing [11, Chapter 13], in which a sentence is given and its syntactic structure is to be determined. The syntactic structure is beneficial in several applications including syntax-based machine translation [14] or natural language understanding [11, Chapter 18]. In parsing, a major performance breakthrough was obtained in 2005 by adding finite-state information to probabilistic context-free grammars [16]. The currently state-of-the-art models (for English) are

Supported by the German Research Foundation (DFG) grant MA/ 4959 / 1-1.

probabilistic context-free grammars with latent variables, which are known as probabilistic finite-state tree automata [10] in TCS. We will review the standard process [7] (expectation maximization), which determines the hidden finite-state information in the hope that similar processes might be helpful also in the TCS community. In addition, we will recall a spectral learning approach [6], which builds on the minimization of nondeterministic field-weighted tree automata [3]. Similarly, advanced evaluation mechanisms like coarse-to-fine parsing [19] that have been developed in NLP should be considered in TCS.

Finally, we will cover an end-user application in NLP. The goal of machine translation [14] is the provision of high-quality and automatic translations of input sentences from one language into another language. The main formalisms used in NLP in this area are *probabilistic synchronous grammars* [5], which originate from the seminal syntax-based translation schemes of [1]. These grammars correspond to certain subclasses of probabilistic finite-state transducers [17] or probabilistic tree transducers [10]. So far, only local versions (grammars without latent variables) are used in state-of-the-art systems, so the effective inclusion of finite-state information remains an open problem in this task. However, the requirements of syntax-based machine translation already spurred a lot of research in TCS because the models traditionally studied had significant shortcomings [12]. In the other direction, advanced models like multi bottom-up tree transducers [2] have made reasonable impact in syntax-based machine translation [4].

References

1. Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation, and Compiling. Prentice Hall (1972)
2. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theor. Comput. Sci. **20**, 33–93 (1982)
3. Bozapalidis, S.: Effective construction of the syntactic algebra of a recognizable series on trees. Acta Informatica **28**(4), 351–363 (1991)
4. Braune, F., Seemann, N., Quernheim, D., Maletti, A.: Shallow local multi-bottom-up tree transducers in statistical machine translation. In: Proceedings of 51st ACL, pp. 811–821. Association for Computer Linguistics (2013)
5. Chiang, D.: An introduction to synchronous grammars. In: Proceedings of 44th ACL. Association for Computational Linguistics (2006), part of a tutorial given with Kevin Knight
6. Cohen, S.B., Stratos, K., Collins, M., Foster, D.P., Ungar, L.: Spectral learning of latent-variable PCFGs. In: Proceedings of 50th ACL, pp. 223–231. Association for Computational Linguistics (2012)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. Series B (Methodol.) **39**(1), 1–38 (1977)
8. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. Springer (2009)
9. Francis, W.N., Kučera, H., Mackie, A.W.: Frequency Analysis of English Usage: Lexicon and Grammar. Houghton Mifflin (1982)
10. Fülöp, Z., Vogler, H.: Weighted tree automata and tree transducers. In: Droste et al. [8], chap. 9, pp. 313–403
11. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn. Prentice Hall (2008)

12. Knight, K.: Capturing practical natural language transformations. Mach. Transl. **21**(2), 121–133 (2007)
13. Knight, K., May, J.: Applications of weighted automata in natural language processing. In: Droste et al. [8], chap. 14, pp. 571–596
14. Koehn, P.: Statistical Machine Translation. Cambridge University Press (2010)
15. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
16. Matsuzaki, T., Miyao, Y., Tsujii, J.: Probabilistic CFG with latent annotations. In: Proceedings of 43rd ACL, pp. 75–82. Association for Computational Linguistics (2005)
17. Mohri, M.: Weighted automata algorithms. In: Droste et al. [8], chap. 6, pp. 213–254
18. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008)
19. Petrov, S.: Coarse-to-Fine Natural Language Processing. Ph.D. thesis, University of California at Bekeley, Berkeley, CA, USA (2009)

Hardware Implementations of Finite Automata and Regular Expressions
Extended Abstract

Bruce W. Watson

FASTAR Group, Department of Information Science
Stellenbosch University, South Africa
bruce@fastar.org

Abstract. This extended abstract sketches some of the most recent advances in hardware implementations (and surrounding issues) of finite automata and regular expressions. The traditional application areas for automata and regular expressions are compilers, text editors, text programming languages (for example Sed, AWK, but more recently Python, and Perl), and text processing in general purpose languages (such as Java, C++ and C#). In all these cases, while the regular expression implementation should be efficient, it rarely forms the performance bottleneck in resulting programs and applications. Even more exotic application areas such as computational biology are not particularly taxing on the regular expression implementation — provided some care is taken while crafting the regular expressions [5].

Contents

Tool Demonstration Papers