6th International Workshop
Weighted Automata: Theory and Applications
WATA 2012

Dresden, Germany, May 29 - June 2, 2012

edited by Manfred Droste and Heiko Vogler

Preface

This report contains the programme and the abstracts of lectures delivered at the workshop *Weighted Automata: Theory and Applications* which took place at Technische Universität Dresden, May 29 - June 2, 2012. This workshop covered all aspects of weighted automata, ranging from the theory of weighted automata and quantitative logics to applications for real-time systems and natural language processing. The workshop was attended by 67 participants from 15 countries.

Three tutorials were given by
- Javier Esparza and Michael Luttenberger (Munich, Germany)
- Orna Kupferman (Jerusalem, Israel)
- Anoop Sarkar (Burnaby, Canada).

In addition, seven survey lectures were presented by

Frank Drewes (Umeå, Sweden)	Zoltán Ésik (Szeged, Hungary)
Paul Gastin (Cachan, France)	Laura Kallmeyer (Düsseldorf, Germany)
Kevin Knight (Los Angeles, USA)	Kim Larsen (Aalborg, Denmark)
Karin Quaas (Leipzig, Germany).	

There was a special session honoring Werner Kuich on the occasion of his 70^{th} birthday. Furthermore 28 talks were selected as technical contributions.

The workshop was organized jointly by the Chair of Automata and Formal Languages of *Leipzig University* and the Chair of Foundations of Programming of *Technische Universität Dresden*. For financial support we would like to thank the *German Research Foundation (DFG)*, *Faculty of Computer Science (TU Dresden)*, *Gesellschaft von Freunden und Förderern der TU Dresden*, and the *International Center for Computational Logic*.

Call for Papers

The journal *Theoretical Computer Science TCS-A* has agreed to publish a special issue on the topic of this workshop. Submissions could be either survey articles or research papers, they will be refereed according to the usual high journal standards. Participation in the workshop is encouraged, but is not a prerequisite for submission.

Authors are asked to submit their contribution preferably in PDF to each of the three editors of the special issue. Please send your files to:

Manfred Droste	`droste@informatik.uni-leipzig.de`
Zoltán Ésik	`ze@inf.u-szeged.hu`
Heiko Vogler	`heiko.vogler@tu-dresden.de`

The deadline for submission is **July 30, 2012**. We intend to ensure a quick refereeing process.

Manfred Droste and Heiko Vogler

Contents

Preface . . . i

I. Scientific Programme 1

Tuesday, May 29 . . . 3
Wednesday, May 30 . . . 4
Thursday, May 31 . . . 5
Friday, June 1 . . . 6
Saturday, June 2 . . . 7

II. Abstracts 9

1. Tutorials 11

Esparza, Luttenberger: Derivation tree analysis of fixed-point equations over semirings . . . 12
Kupferman: Weighted Automata . . . 15
Sarkar: Strings to Trees to Strings: A Survey of Tree Adjoining Grammars . . . 16

2. Survey Lectures 17

Drewes: Weighted Automata, Coloured Pictures, and More . . . 18
Ésik: Axiomatizing weighted bisimulation . . . 19
Gastin, Monmege: Efficient computations with pebbles . . . 20
Kallmeyer, Maier: Weighted Deductive Parsing using Probabilistic Linear Context-Free Rewriting Systems . . . 21
Knight: Generative Processes for Human Language Translation: Strings, Trees, and Graphs . . . 23
Larsen: From Priced Timed Automata to Energy Games . . . 24
Quaas: The Fruitful interplay of Weighted Automata and Weighted Timed Automata . . . 27

3. Technical Contributions 29

Berlinkov: Markov Chains and Synchronizing Automata . . . 30
Bollig, Gastin, Monmege, Zeitoun: Series and Infinite Sums: Automata, Expressions, and Logic . . . 31
Braune, Seemann, Quernheim, Maletti: Machine Translation with Multi Bottom-up Tree Transducers . . . 32
Bubenzer, Watson: A Minimality-Preserving Regular Calculus of ADFAs . . . 34
Büchse, Dietze, Osterholzer, Fischer, Leuschner: Vanda - A Statistical Machine Translation Toolkit . . . 36

Büchse, Fischer: Deciding the Twins Property for Weighted Tree Automata over Extremal Semifields 39
Ćirić, Ignjatović: Fuzziness in Automata Theory: Why? How? 40
Crosswhite: Embracing divergence: an approach to generalizing weighted automata to infinite words with applications in quantum simulation 43
Damljanović, Ćirić, Ignjatović: Bisimulations for weighted automata 44
Droste, Götze, Märcker, Meinecke: Weighted Tree Automata over Valuation Monoids and Their Characterization by Weighted Logics 46
Fahrenberg, Legay, Quaas: Büchi Conditions for Generalized Energy Automata . 47
Felgenhauer, Schwarz, Waldmann: Size-Change Termination and Arctic Matrix Monoids 48
Fijalkow, Gimbert, Oualhadj: Deciding the Value 1 Problem of Probabilistic Leaktight Automata 50
Fülöp, Vogler: Weighted Tree Automata and Branching Transitive Closure Logics 52
Huber, Kölbl: Weighted Automata as Semantic Representation 53
Ignjatović, Ćirić, Simović: Subsystems of fuzzy transition systems 55
Jančić, Jančić, Ignjatović, Ćirić: Fuzzy automata: Determinization using simulations 57
Jančić, Jančić, Ignjatović, Ćirić: Fuzzy and weighted automata: Canonization methods 58
Jurish, Würzner: Multi-threaded composition of weighted finite-state transducers 59
Kabanova: Distance between formulas of the five-valued Lukasiewicz logic and the uncertainty measure of expert statements 62
Kirsten: Decidability, Undecidability, and PSPACE-Completeness of the Twins Property in the Tropical Semiring 64
Lehmann, Peñaloza: The Complexity of Computing the Behaviour of Weighted Büchi Automata over Lattices 65
Lombardy, Sakarovitch: On the weighted closure problem 66
Maletti, Quernheim: Pushing for weighted tree automata 67
Meinecke, Quaas: Parameterized Model Checking of Weighted Networks 69
Perevoshchikov: Büchi-Type Theorems for Unambiguous, Functional and Multi-Weighted Automata 72
Stüber: Consistency of Probabilistic Context-Free Grammars 73
Weidner: Probabilistic Automata and Probabilistic Logic 75

Part I.

Scientific Programme

Tuesday, May 29

08:30-09:00			REGISTRATION
09:00-10:30	Esparza, Luttenberger *Derivation tree analysis of fixed-point equations over semirings*		TUTORIAL
10:30-11:00			BREAK
11:00-12:30	Sarkar *Strings to Trees to Strings: A Survey of Tree Adjoining Grammars*		TUTORIAL
12:30-14:00			LUNCH
14:00-15:00	Knight *Generative Processes for Human Language Translation: Strings, Trees, and Graphs*		SURVEY LECTURE
15:00-15:20	Meinecke, Quaas *Parameterized Model Checking of Weighted Networks*		TECHNICAL CONTRIBUTION
15:20-15:40	Jurish, Würzner *Multi-threaded composition of weighted finite-state transducers*		TECHNICAL CONTRIBUTION
15:40-16:00	Crosswhite *Embracing divergence: an approach to generalizing weighted automata to infinite words with applications in quantum simulation*		TECHNICAL CONTRIBUTION
16:00-16:30			BREAK
16:30-16:50	Lombardy, Sakarovitch *On the weighted closure problem*		TECHNICAL CONTRIBUTION
16:50-17:10	Berlinkov *Markov Chains and Synchronizing Automata*		TECHNICAL CONTRIBUTION

Wednesday, May 30

08:30-09:00		REGISTRATION
09:00-10:30	Esparza, Luttenberger *Derivation tree analysis of fixed-point equations over semirings*	TUTORIAL
10:30-11:00		BREAK
11:00-12:30	Sarkar *Strings to Trees to Strings: A Survey of Tree Adjoining Grammars*	TUTORIAL
12:30-14:00		LUNCH
14:00-15:00	Larsen *From Priced Timed Automata to Energy Games*	SURVEY LECTURE
15:00-15:20	Braune, Seemann, Quernheim, Maletti *Machine Translation with Multi Bottom-up Tree Transducers*	TECHNICAL CONTRIBUTION
15:20-15:40	Huber, Kölbl *Weighted Automata as Semantic Representation*	TECHNICAL CONTRIBUTION
15:40-16:00	Büchse, Dietze, Osterholzer, Fischer, Leuschner *Vanda - A Statistical Machine Translation Toolkit*	TECHNICAL CONTRIBUTION
16:00-16:30		BREAK
16:30-16:50	Fahrenberg, Legay, Quaas *Büchi Conditions for Generalized Energy Automata*	TECHNICAL CONTRIBUTION
16:50-17:10	Bubenzer, Watson *A Minimality-Preserving Regular Calculus of ADFAs*	TECHNICAL CONTRIBUTION

Thursday, May 31

08:30-09:00		REGISTRATION
09:00-10:30	Kupferman *Weighted Automata*	TUTORIAL
10:30-11:00		BREAK
11:00-12:00	Gastin, Monmege *Efficient computations with pebbles*	SURVEY LECTURE
12:00-12:20	Bollig, Gastin, Monmege, Zeitoun *Series and Infinite Sums: Automata, Expressions, and Logic*	TECHNICAL CONTRIBUTION
12:20-14:00		LUNCH
14:00-15:00	Quaas *The Fruitful interplay of Weighted Automata and Weighted Timed Automata*	SURVEY LECTURE
15:00-16:00	Special session for Prof. Kuich	
16:00-16:30		BREAK
16:30-16:50	Ćirić, Ignjatović *Fuzziness in Automata Theory: Why? How?*	TECHNICAL CONTRIBUTION
16:50-17:10	Ignjatović, Ćirić, Simović *Subsystems of fuzzy transition systems*	TECHNICAL CONTRIBUTION
17:10-17:30	Damljanović, Ćirić, Ignjatović *Bisimulations for weighted automata*	TECHNICAL CONTRIBUTION
17:30-17:50	Fijalkow, Gimbert, Oualhadj *Deciding the Value* 1 *Problem of Probabilistic Leaktight Automata*	TECHNICAL CONTRIBUTION
20:00-open		CONFERENCE DINNER

Friday, June 1

08:30-09:00		REGISTRATION
09:00-10:30	Kupferman *Weighted Automata*	TUTORIAL
10:30-11:00		BREAK
11:00-12:00	Kallmeyer, Maier *Weighted Deductive Parsing using Probabilistic Linear Context-Free Rewriting Systems*	SURVEY LECTURE
12:00-12:20	Stüber *Consistency of Probabilistic Context-Free Grammars*	TECHNICAL CONTRIBUTION
12:20-14:00		LUNCH
14:00-15:00	Ésik *Axiomatizing weighted bisimulation*	SURVEY LECTURE
15:00-15:20	Kirsten *Decidability, Undecidability, and PSPACE-Completeness of the Twins Property in the Tropical Semiring*	TECHNICAL CONTRIBUTION
15:20-15:40	Büchse, Fischer *Deciding the Twins Property for Weighted Tree Automata over Extremal Semifields*	TECHNICAL CONTRIBUTION
15:40-16:00	Maletti, Quernheim *Pushing for weighted tree automata*	TECHNICAL CONTRIBUTION
16:00-16:30		BREAK
16:30-16:50	Jančić, Jančić, Ignjatović, Ćirić *Fuzzy automata : Determinization using simulations*	TECHNICAL CONTRIBUTION
16:50-17:10	Jančić, Jančić, Ignjatović, Ćirić *Fuzzy and weighted automata: Canonization methods*	TECHNICAL CONTRIBUTION
17:10-17:30	Felgenhauer, Schwarz, Waldmann *Size-Change Termination and Arctic Matrix Monoids*	TECHNICAL CONTRIBUTION
17:30-17:50	Lehmann, Peñaloza *The Complexity of Computing the Behaviour of Weighted Büchi Automata over Lattices*	TECHNICAL CONTRIBUTION

Saturday, June 2

08:30-09:00		REGISTRATION
09:00-10:00	Drewes *Weighted Automata, Coloured Pictures, and More*	SURVEY LECTURE
10:00-10:20	Droste, Götze, Märcker, Meinecke *Weighted Tree Automata over Valuation Monoids and Their Characterization by Weighted Logics*	TECHNICAL CONTRIBUTION
10:20-11:00		BREAK
11:00-11:20	Weidner *Probabilistic Automata and Probabilistic Logic*	TECHNICAL CONTRIBUTION
11:20-11:40	Fülöp, Vogler *Weighted Tree Automata and Branching Transitive Closure Logics*	TECHNICAL CONTRIBUTION
11:40-12:00	Kabanova *Distance between formulas of the five-valued Lukasiewicz logic and the uncertainty measure of expert statements*	TECHNICAL CONTRIBUTION
12:00-12:20	Perevoshchikov *Büchi-Type Theorems for Unambiguous, Functional and Multi-Weighted Automata*	TECHNICAL CONTRIBUTION

Part II.

Abstracts

1. Tutorials

Derivation tree analysis of fixed-point equations over semirings*

Javier Esparza and Michael Luttenberger

Institut für Informatik
Technische Universität München
85748 Garching, Germany
{esparza, luttenb}@model.in.tum.de

Fixed-point equations over semirings are the mathematical foundation of program analysis, and have many other applications, e.g. in probability theory and natural language processing. These equations can be formally assigned context-free grammars in a natural way: for instance, the equation $X = aXX + bX + c$, where a, b, c are semiring elements, is assigned the grammar $X \to aXX \mid bX \mid c$ with a, b, c as terminals. We show in the tutorial how an analysis of the derivation trees of this grammar leads to interesting algorithms for computing or at least approximating the least solution of the equation.

Introduction

We are interested in computing (or approximating) solutions of *systems of fixed-point equations* of the form

$$\begin{aligned} X_1 &= f_1(X_1, X_2, \ldots, X_n) \\ X_2 &= f_2(X_1, X_2, \ldots, X_n) \\ &\vdots \\ X_n &= f_n(X_1, X_2, \ldots, X_n) \end{aligned}$$

where $X_1, X_2, \ldots, X_n$ are variables and $f_1, f_2, \ldots, f_n$ are n-ary functions over some common domain S. Fixed-point equations are a natural way of describing the equilibrium states of systems with n interacting components (particles, populations, program points, etc.). Loosely speaking, the function f_i describes the next state of the i-th component as a function of the current states of all components, and so the solutions of the system describe the equilibrium states.

In formal language theory, such systems arise naturally in the context of context-free languages – a perspective pioneered by Schützenberger and Chomsky in the 1960s, see e.g. [4]. Context-free grammars can readily be interpreted as *algebraic systems*, i.e. the functions f_i are multivariate polynomials, over the semiring of languages over the given alphabet Σ. The language represented by the given grammar then coincides with the least solution of the algebraic system.

We are in particular interested in algebraic systems over ω-continuous semirings. As is well-known, ω-continuity guarantees that an algebraic system always has a least solution [16]. These systems arise naturally in the setting of interprocedural program analysis, or stochastic recursive programs.

In this tutorial, we show how efficient algorithms for either computing or at least approximating the least solution of algebraic systems over ω-continuous semirings can be obtained by essentially moving from the given system to a context-free grammar and from

*This work was partially supported by the project "Polynomial Systems on Semirings: Foundations, Algorithms, Applications" of the Deutsche Forschungsgemeinschaft.

there to the associated language of derivation trees – a shift of perspective well-known to be fruitful, see e.g. [20, 1, 2]:

The first part of the tutorial starts with a known result: the least solution of a system is equal to the *value* (or: yield or flattening) of its associated grammar, where the value of a grammar is defined as the sum of the values of its derivation trees, and the value of a derivation tree is defined as the (ordered) product of its leaves. This connection allows us to approximate the least solution of a system by computing the values of "approximations" to the grammar. Loosely speaking, a grammar G_1 approximates G_2 if every derivation tree of G_1 is a derivation tree of G_2 up to irrelevant details. We show that Kleene's theorem, which not only proves the existence of the least solution, but also provides an algorithm for approximating it, corresponds to approximating G by grammars $G^{[1]}, G^{[1]}, \ldots$ where $G^{[h]}$ generates the derivation trees of G of height at most h. We then introduce a faster approximation by grammars $H^{[1]}, H^{[1]}, \ldots$ where $H^{[h]}$ generates the derivation trees of G of *dimension* at most h [8, 9]. We show that this approximation is a generalization of Newton's method for approximating the zero of a differentiable function. As a first application, we prove that for semirings where addition is idempotent and multiplication is commutative the approximation reaches the least solution after a finite number of steps bounded by the number of equations of the system [14, 9].

In the second part of the tutorial, we discuss applications and extensions of the results presented in the first part. We give a lower bound for the speed at which Newton's method converges over ω-continuous commutative semirings [18]. We then present several special cases of idempotent semirings, and show how to use their additional axioms to compute the least solution more efficiently [8].

If time permits, we will also give a brief overview of related results [3, 11, 6, 7, 10, 5, 15, 19, 21, 12, 17, 13].

References

[1] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theor. Comput. Sci.*, 18:115–148, 1982.

[2] S. Bozapalidis. Equational elements in additive algebras. *Theory Comput. Syst.*, 32(1):1–33, 1999.

[3] T. Brázdil, J. Esparza, S. Kiefer, and M. Luttenberger. Space-efficient scheduling of stochastically generated tasks. In *ICALP (2)*, pages 539–550, 2010.

[4] N. Chomsky and M.P. Schützenberger. *Computer Programming and Formal Systems*, chapter The Algebraic Theory of Context-Free Languages, pages 118 – 161. North Holland, 1963.

[5] A. P. Ershov. On programming of arithmetic operations. *Commun. ACM*, 1(8):3–9, 1958.

[6] J. Esparza and P. Ganty. Complexity of pattern-based verification for multithreaded programs. In *POPL*, pages 499–510, 2011.

[7] J. Esparza, P. Ganty, S. Kiefer, and M. Luttenberger. Parikhs theorem: A simple and direct automaton construction. *Inf. Process. Lett.*, 111(12):614–619, 2011.

[8] J. Esparza, S. Kiefer, and M. Luttenberger. Derivation tree analysis for accelerated fixed-point computation. In *DLT*, pages 301–313, 2008.

[9] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian program analysis. *J. ACM*, 57(6):33, 2010.

[10] P. Flajolet, J. Françon, and J. Vuillemin. Towards analysing sequences of operations for dynamic data structures (preliminary version). In *FOCS*, pages 183–195, 1979.

[11] P. Ganty, R. Majumdar, and B. Monmege. Bounded underapproximations. In *CAV*, pages 600–614, 2010.

[12] S. Ginsburg and E. Spanier. Derivation-bounded languages. *Journal of Computer and System Sciences*, 2:228–250, 1968.

[13] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS*, pages 31–40, 2007.

[14] M. W. Hopkins and D. Kozen. Parikh's theorem in commutative Kleene algebra. In *Logic in Computer Science*, pages 394–401, 1999.

[15] R. E. Horton. Erosional development of streams and their drainage basins: hydrophysical approach to quantitative morphology. *Geological Society of America Bulletin*, 56(3):275–370, 1945.

[16] W. Kuich. *Handbook of Formal Languages*, volume 1, chapter 9: Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata, pages 609 – 677. Springer, 1997.

[17] Mark Luker. A family of languages having only finite-index grammars. *Information and Control*, 39(1):14–18, 1978.

[18] Michael Luttenberger. An extension of parikh's theorem beyond idempotence. *CoRR*, abs/1112.2864, 2011.

[19] A. N. Strahler. Hypsometric (area-altitude) analysis of erosional topology. *Geological Society of America Bulletin*, 63(11):1117–1142, 1952.

[20] J. W. Thatcher. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *J. Comput. Syst. Sci.*, 1(4):317–322, 1967.

[21] M.K. Yntema. Inclusion relations among families of context-free languages. *Information and Control*, 10:572–597, 1967.

Weighted Automata

Orna Kupferman

School of Computer Science and Engineering
Hebrew University
Jerusalem, Israel

A nondeterministic weighted finite automaton maps an input word to a numerical value. Applications of weighted automata include formal verification of quantitative properties, as well as text, speech, and image processing.

Technically, each weighted automaton is defined with respect to an algebraic semiring. For example, when the automaton is defined with respect to the tropical semiring, each transition is labeled by a real number – the cost of traversing it. The value of a run is the sum of the costs of the transitions taken along the run, and the value of a word is the value of a minimal run on it.

The rich structure of weighted automata makes them intriguing mathematical objects. Fundamental problems that have been solved decades ago for Boolean automata are still open or known to be undecidable in the weighted setting. For example, while in the Boolean setting, nondeterminism does not add to the expressive power of the automata, not all weighted automata can be determinized. Also, the problem of deciding whether a given nondeterministic weighted automaton can be determinized is still open, in the sense we do not even know whether it is decidable.

The tutorial introduces weighted automata, their theoretical properties, decision problems for them, and some of their applications.

Strings to Trees to Strings: A Survey of Tree Adjoining Grammars

Anoop Sarkar

School of Computing Science
Simon Fraser University
Burnaby, BC, Canadai
anoop@sfu.ca

Parsing algorithms commonly used in computational linguistics are built upon formal grammars that generate strings. This tutorial will motivate an approach using tree adjoining grammars (TAG) that goes from strings to trees. Formal properties of TAG that make it attractive for computational linguistics will be introduced such as closure properties, enabling context-sensitivity with trees, handling of crossing dependencies and lexicalization of grammars. We will briefly describe the formal relationship between TAG and various classes of dependency tree structures.

There is a hierarchy of generative devices that generate trees: starting with regular tree languages (RTLs), which are contained within context-free tree languages (CFTLs), and so on. The string yield of the RTLs is exactly the set of context-free languages, while the yield of the CFTLs is exactly the set of indexed languages. We compare the generative capacity of tree adjoining grammars in terms of strings and trees with previously proposed simplifications of CFTGs, called monadic simple CFTGs.

Synchronous grammars are quite useful extensions with many applications in computational linguistics such as machine translation and natural language semantics. We contrast the generative capacity of synchronous tree adjoining grammars and their relationship with string to tree transducers.

We will cover various probabilistic and weighted variants of tree adjoining grammars that have been shown to be useful for statistical parsing and machine translation applications. We will see how the complex structural descriptions in TAG can be exploited in a corpus-based learning approach for parsing language. We describe a robust shallow parsing approach called SuperTagging and see how TAGs are used to define a simple model for statistical parsers that can obtain state-of-the-art accuracy.

For parsing English in particular, the emphasis has been on reducing the generative capacity of tree adjoining grammars in order to parse in cubic time. We compare the various results in the literature which use variants of tree adjoining grammar for phrase structure parsing and dependency parsing.

2. Survey Lectures

Weighted Automata, Coloured Pictures, and More

F. Drewes

Dept. of Computer Science
Umeå University, Sweden

Pictures that consist of geometrical objects like squares and circles can be generated by tree grammars whose output tree are regarded as expressions over operations that transform such objects. This is the tree-based notion of picture generation studied in detail in [2]. Coloured pictures can be obtained in this setting by

- providing each object with attributes representing colour (i.e., *red*, *green*, and *blue* values in the interval $[0, 1]$), and
- adding operations that manipulate these attribute values.

A simple type of such colour operation has been proposed in [2, Chapter 7]. In the talk, these colour operations will be explained, and a few results that have been obtained for the resulting picture languages will be discussed.

As it turns out, the effect of the colour operations considered can be described by weighted finite-state automata in a rather straightforward way. Instead of using weighted finite-state *tree* automata that work on the generated trees, and thus compute a single value for the input tree as a whole, we use ordinary weighted finite-state *string* automata (wfsa) that process the paths of a given tree. In this way, every geometrical object in the generated picture (corresponding to a leaf of the tree) is assigned its own individual attribute values that represent the colour of the object in question.

This way of using wfsa for the generation of coloured pictures is closely related to the well-known use of wfsa for picture compression [1]. However, it can be employed for other purposes as well, since the objects being generated do not need to be pictures, and the attributes do not need to represent colour. One example is the generation of (uncoloured) pictures, where the transformation of objects can be described in terms of weights. Another example is given by string languages (e.g., context-free ones), where weights can be used to provide the individual letters with attributes such as font size and (typographical) weight.

References

[1] Jürgen Albert and Jarkko Kari. Digital image compression. In Werner Kuich, Manfred Droste, and Heiko Vogler, editors, *Handbook of Weighted Automata*, chapter 11, pages 453–479. Springer, 2009.

[2] Frank Drewes. *Grammatical Picture Generation – A Tree-Based Approach.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

Axiomatizing weighted bisimulation

Z. Ésik

Dept. of Computer Science
University of Szeged, Hungary

We provide various axiomatizations of weighted tree equivalence and weigthed bisimulation equivalence of finite state processes. We derive our completeness results from the characterization of the behavior of finite state processes with respect to the tree semantics or the weighted bisimulation semantics as the free theories in certain classes of enriched Lawvere theories. We also compare weighted bisimulation semantics and weighted trace semantics.

References

[1] L. Aceto, Z. Ésik, A. Ingólfsdóttir, Equational axioms for probabilistic bisimilarity, in: Proceedings of *Algebraic Methodology and Software Technology, AMAST 2002*, Reunion, LNCS 2422, Springer, 2002, 239–253.

[2] S.L. Bloom, Z. Ésik, *Iteration Theories*, EATCS Monograph Series on Theoretical Computer Science, Springer, 1993.

[3] S.L. Bloom, Z. Ésik, D. Taubner, Iteration theories of synchronization trees, *Information and Computation*, 102(1993), 1-55.

[4] Z. Ésik, Multi-linear iterative K-semialgebras, in: Proc. *27th Int. Conference on Mathematical Foundations of Programming Semantics*, Carnegie Mellon University, May 2011, ENTCS, Vol. 276, 2011, 159–170.

[5] Z. Ésik, Axiomatizing weighted bisimulation, to appear.

[6] Z. Ésik, W. Kuich, A generalization of Kozen's axiomatization of the equational theory of the regular sets, in: *Words, Semigroups, and Transductions*, M. Ito, Gh. Paun, Sheng Yu, eds., World Scientific, 2001, 99–114.

[7] Z. Ésik, W. Kuich, Axiomatizing rational series, in: proc. *8th Panhellenic Logic Symposium*, Ioannina, 2011, 30–34.

[8] Z. Ésik, W. Kuich, Free iterative and iteration K-semialgebras, *Algebra Universalis*, 67(2012), 141–162.

[9] K.G. Larsen, A. Skou, Bisimulation through probabilistic testing, *Information and Computation*, 94(1991), 1–28.

[10] R. Milner, A complete inference system for a class of regular behaviours, *J. Comput. System Sci.*, 28 (1984), 439–466.

[11] E.W. Stark, S.A. Smolka, A complete axiom system for finite-state probabilistic processes, in: *Proof, Language, and Interaction: Essays in Honour of Robin Milner*, MIT Press, Cambridge, MA, 2000, 571–595.

Efficient computations with pebbles

Paul Gastin and Benjamin Monmege

LSV, ENS Cachan, CNRS & INRIA
France

We extend weighted automata and weighted rational expressions with 2-way moves and (reusable) pebbles. We show with examples from natural language modeling and quantitative model-checking that weighted expressions and automata with pebbles are more expressive and allow much more natural and intuitive specifications than classical ones. We extend Kleene-Schützenberger theorem showing that weighted expressions and automata with pebbles have the same expressive power. We focus on an efficient translation from expressions to automata. We also prove that the evaluation problem for weighted automata can be done very efficiently if the number of (reusable) pebbles is low.

Weighted Deductive Parsing using Probabilistic Linear Context-Free Rewriting Systems

Laura Kallmeyer and Wolfgang Maier

Heinrich-Heine-Universität Düsseldorf
Düsseldorf, Germany

The work reported here presents the first efficient implementation of a weighted deductive CYK parser for Probabilistic Linear Context-Free Rewriting Systems (PLCFRS). The parser is evaluated on the German treebank NeGra.

In the context of data-driven natural language parsing, challenges always arise from discontinous constituents since they call for formalisms which have a larger domain of locality than context-free grammars. Discontinuities are particularly frequent in so-called free word order languages such as German. An example is given in Fig. 2.1.

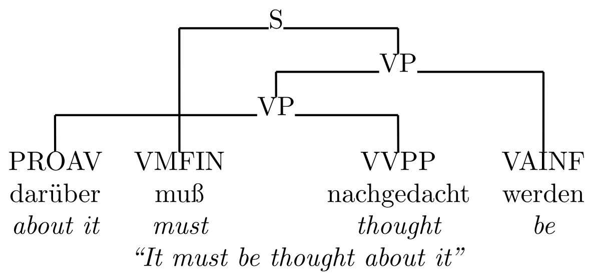


Figure 2.1.: A sample tree from NeGra

Linear Context-Free Rewriting Systems (LCFRSs) are a natural extension of CFGs where the non-terminals can span tuples of possibly non-adjacent strings (see Fig. 2.2). Due to this, they can describe discontinuities in a straightforward way and are therefore a natural candidate to be used for data-driven parsing.

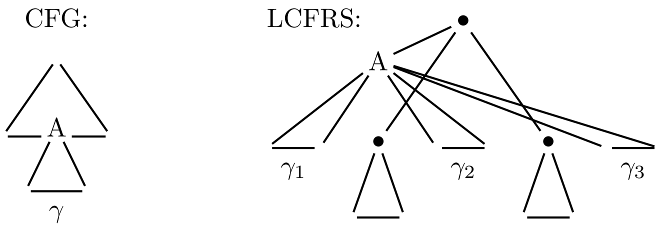


Figure 2.2.: Different domains of locality

Based on this motivation, we extract probabilistic LCFRSs (PLCFRSs) from German treebanks and use them for parsing. Our parser performs, after binarization of the grammar, a weighted deductive CYK parsing. The weights incorporate the probability of the best parse tree for a given category and a given span. In addition to this, in order to speed up parsing, we also add an estimate or the outside probability of such a category-span pair to its weight. We experimented with several outside estimates where some of them are monotonic and therefore allow for true A* parsing.

We evaluate our parser with grammars extracted from the German NeGra treebank. Our experiments show that data-driven LCFRS parsing is feasible and yields output of competitive quality.

Generative Processes for Human Language Translation: Strings, Trees, and Graphs

Kevin Knight

USC / Information Sciences Institute
Marina del Rey, California, USA

Human language translation (e.g., Chinese to English) is profitably viewed as an infinite weighted relation over strings. A Chinese sentence will have many translations into English, with some more likely than others. When we translate by machine, we take a novel input text and search for a likely translation.

Some important research topics for human language translation are: (1) how to represent this relation formally, (2) how to efficiently populate it, and (3) how to efficiently translate with it.

On the first topic – representation – automata theory provides a range of possible answers, from finite-state string transducers on up. Of course, automata are not strictly necessary. For example, we could simply build a large set of weighted binary features and let them vote on candidate translations, summing all weights. But automata turn out to be useful – they not only assign scores to translation pairs, but their generative processes also help us organize efficient search for high-scoring translations.

This talk will first cover existing models of translation, most of which deal with transforming strings or syntactic trees. Translation system designers have recently begun to explore moving from syntactic representations to semantic ones. This talk will also cover new automata questions raised by this trend.

From Priced Timed Automata to Energy Games

Kim G. Larsen

CS Department, Aalborg University, Denmark

This tutorial will offer a comprehensive review of priced timed automata and energy games highlightening results obtained and problems that remain open.

Priced (or weighted) *timed automata* [2, 1] have emerged as a useful formalism for formulating a wide range of resource-allocation and optimization problems [3] with applications in areas such as embedded systems.

The intial optimization problem considered was that of cost-optimal reachability, shown to be decidable [2] and PSPACE-complete [5]. Later decidability and PSPACE-completeness of the problem of optimal infinite runs, in terms of minimal (maximal) cost per time ratio or accumulated discounted cost has been established [4, 11]. For priced timed games cost-optimal winning strategies for reachability games has been shown decidable [9] when restricting to priced timed games with only one clock. The recent result of [12] narrows the gap in lower and upper bound complexity to PTIME versus EXPTIME, still leaving the exact complexity open.

In [7], a new class of resource-allocation problems was introduced, namely that of constructing infinite schedules subject to boundary constraints on the accumulation of resources, so-called *energy games*.

For weighted timed automata with a single clock and a single weight variable, the existence of a lower-bound constrained infinite run has been shown decidable in polynomial time [7] with the restriction that no discrete updates of the accumulated weight occur on transitions. In [6], it is shown that the problem remains decidable if this restriction is lifted and even if the accumulated weight grows not only linearly but also exponentially. In contrast, the existence of interval-constrained infinite runs—where a simple energy-maximizing strategy does not suffice—have recently been proven undecidable for weighted timed automata with varying numbers of clocks and weight variables: e.g. two clocks and two weight variables [14], one clock and two weight variables [10], and two clocks and one weight variable [13]. Also, the interval-constrained problem is undecidable for weighted timed automata with one clock and one weight variable in the game setting [7].

Still, the general problem of existence of infinite lowerbound runs for weighted timed automata has remained unsettled since [7]. In the recent paper [8] we close this open problem showing that it is undecidable for weighted timed automata with four or more clocks. Given that this problem looks rather simple (it suffices to consider energy-maximizing runs), we find this result quite surprising and somewhat disappointing. Thus, we consider a number of related problems for which we show decidability and settle complexity. In particular, the undecidability result assumes a fixed and known initial energy-level. We show that the related problem of existence of an initial energy-level allowing an infinite lower-bound constrained run is decidable in PSPACE. We also investigate the variant of these problems, where only the existence of time-bounded runs are required. We show that this restriction makes our original problem decidable and NEXPTIME-complete.

References

[1] R. Alur, S. La Torre, and G. J. Pappas. Optimal Paths in Weighted Timed Automata.

In *Proc. of 4th Int. Work. Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 49–62. Springer-Verlag, 2001.

[2] G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In *Proc. 4th Int. Work. on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 147–161. Springer-Verlag, 2001.

[3] Gerd Behrmann, Ed Brinksma, Martijn Hendriks, and Angelika Mader. Production scheduling by reachability analysis - a case study. In *IPDPS*. IEEE Computer Society, 2005.

[4] P. Bouyer, E. Brinksma, and K. G. Larsen. Staying alive as cheaply as possible. In *Proc. 7th Int. Work. on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *LNCS*, pages 203–218. Springer-Verlag, 2004.

[5] Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Improved undecidability results on weighted timed automata. *Inf. Process. Lett.*, 98(5):188–194, 2006.

[6] Patricia Bouyer, Uli Fahrenberg, Kim G. Larsen, and Nicolas Markey. Timed automata with observers under energy constraints. In Karl Henrik Johansson and Wang Yi, editors, *HSCC*, pages 61–70. ACM ACM, 2010.

[7] Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *FORMATS*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.

[8] Patricia Bouyer, Kim G. Larsen, and Nicolas Markey. Lower-bound constrained runs in weighted timed automata. In *Proceedings of the 9th International Conference on Quantitative Evaluation of Systems (QEST'12)*, London, UK, September 2012. IEEE Computer Society Press. To appear.

[9] Patricia Bouyer, Kim Guldstrand Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost optimal strategies in one clock priced timed games. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS*, volume 4337 of *LNCS*, pages 345–356. Springer-Verlag, 2006.

[10] Uli Fahrenberg, Line Juhl, Kim G. Larsen, and Jirí Srba. Energy games in multiweighted automata. In Antonio Cerone and Pekka Pihlajasaari, editors, *ICTAC*, volume 6916 of *Lecture Notes in Computer Science*, pages 95–115. Springer, 2011.

[11] Ulrich Fahrenberg and Kim Guldstrand Larsen. Discount-optimal infinite runs in priced timed automata. *Electr. Notes Theor. Comput. Sci.*, 239:179–191, 2009.

[12] Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A faster algorithm for solving one-clock priced timed games. *CoRR*, abs/1201.3498, 2012.

[13] Nicolas Markey. *Verification of Embedded Systems – Algorithms and Complexity.* Mémoire d'habilitation, École Normale Supérieure de Cachan, France, April 2011.

[14] Karin Quaas. On the interval-bound problem for weighted timed automata. In Adrian Horia Dediu, Shunsuke Inenaga, and Carlos Martín-Vide, editors, *LATA*, volume 6638 of *Lecture Notes in Computer Science*, pages 452–464. Springer, 2011.

The Fruitful interplay of Weighted Automata and Weighted Timed Automata

Karin Quaas

Universität Leipzig, Germany

Weighted timed automata [2] form an extension of timed automata [1], where both locations and edges are assigned weights over the positive reals. In this way, they allow for the modelling of *continuous* consumption of resources. Many interesting problems, e.g. optimal reachability [3] and model checking [4], have been studied. These works consider the *behaviours* of weighted timed automata rather than their means as an acceptor of languages and/or formal power series.

In [7], we propose a uniform definition of weighted timed automata over semirings, adapting an approach that has been pursued for many decades in the theory of weighted automata [6]. We present two generalizations of fundamental classical theorems, namely the Kleene-Schützenberger theorem and the Büchi theorem [7, 8]. In these works, we can profit from the elaborated work that has been done in the field of weighted automata.

On the other hand side, research in the weighted automata community may be stimulated by the work done in the real-time community. As an example, we present some results concerning energy games [5] and model-checking of automata and weighted/timed extensions of linear temporal logic (LTL).

References

[1] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[2] Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. *Theor. Comput. Sci.*, 318(3):297–322, 2004.

[3] Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, 2007.

[4] Patricia Bouyer, Ed Brinksma, and Kim Guldstrand Larsen. Optimal infinite scheduling for multi-priced timed automata. *Formal Methods in System Design*, 32(1):3–23, 2008.

[5] Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *FORMATS*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.

[6] Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata.* EATCS Monographs in Theoretical Computer Science. Springer, 2009.

[7] Manfred Droste and Karin Quaas. A Kleene-Schützenberger theorem for weighted timed automata. *Theor. Comput. Sci.*, 412(12-14):1140–1153, 2011.

[8] Karin Quaas. MSO logics for weighted timed automata. *Formal Methods in System Design*, 38(3):193–222, 2011.

3. Technical Contributions

Markov Chains and Synchronizing Automata

Mikhail Berlinkov

Laboratory of combinatorics, Institute of Mathematics and Computer Science
Ural Federal University, Ekaterinburg, Russia
berlm@mail.ru

Synchronizing automata serve as transparent and natural models of error-resistant systems in many applications (coding theory, robotics, testing of reactive systems) and also reveal interesting connections with symbolic dynamics and other parts of mathematics.

In 1964 Černý conjectured that each n-state synchronizing automaton possesses a reset word of length at most $(n-1)^2$. The conjecture still remains open, moreover, the best upper bound on the reset length (that is, the minimum length of reset words) for n-state synchronizing automata known so far is cubic in n. Thus, a major problem in the area is to prove quadratic (in n) upper bounds. Since 1964, this problem has been solved for a few special classes of synchronizing automata. In [http://arxiv.org/abs/1203.3402] we have presented a new idea for bounding reset lengths by assigning some positive weights to the states of automata under consideration. These weights are determined by the weighted adjacency matrix S of a given automaton A. In this paper we further explore this idea and show that the reset length of A can be bounded by a function of eigenvalues of S. This bound holds in the general case (that is, with no apriori restrictions imposed on A) and its specializations to some specific classes of synchronizing automata yield a few previously known results in a uniform way.

Series and Infinite Sums: Automata, Expressions, and Logic

Benedikt Bollig[1], Paul Gastin[1], Benjamin Monmege[1], and Marc Zeitoun[2]

[1]LSV, ENS Cachan, CNRS & INRIA, France
`lastname@lsv.ens-cachan.fr`

[2]LaBRI, Univ. Bordeaux & CNRS, France
`mz@labri.fr`

We introduce *unrestricted* automata, logic, and rational expressions that compute quantitative properties of words in *arbitrary* semirings. We consider two-way automata (similar to two-way word automata and tree-walking automata) and navigating logics and expressions (similar to transitive closure logic [2] and XPath [3]). Moreover, enriched with pebbles, they capture first-order logic with positive transitive closure [2]. Usually, the value that a weighted automaton associates with a word, is the sum of all run weights (each of which is the product of transition weights). In the presence of two-way devices, however, there might be an infinite number of runs, which, for a semantics to be well defined, requires additional assumptions on the semiring or must impose syntactic restrictions on the specification. In [1], a logical characterization of weighted two-way pebble automata is given in terms of "one-way" transitive closure logic. The automata have to restrict to non-looping runs, though, which is a semantic and sometimes (e.g., in a probabilistic setting) inappropriate or unnecessary restriction.

We provide a new general framework that does not rely on such assumptions. Instead of computing infinite sums directly, we consider a "family semantics", which delays summation and gathers all possible run weights in a multiset. We actually obtain a Kleene-Schützenberger-Büchi correspondence showing that automata, logic, and expressions are expressively equivalent wrt. the family semantics in the sense that it preserves a bijection between two multisets. We apply the general framework to specific settings such as complete semirings and the probabilistic semiring where (infinite or finite) sums are well defined. In particular, we obtain a class of probabilistic expressions that are equivalent to probabilistic two-way pebble automata. To the best of our knowledge, this is the first Kleene-Schützenberger correspondence in a probabilistic setting and, therefore, constitutes a first step towards probabilistic database query languages.

This work has been submitted to ICALP'12.

References

[1] B. Bollig, P. Gastin, B. Monmege, and M. Zeitoun. Pebble weighted automata and transitive closure logics. In *ICALP'10*, vol. 6199 *LNCS*, p. 587–598. Springer, 2010.

[2] J. Engelfriet and H. J. Hoogeboom. Automata with nested pebbles capture first-order logic with transitive closure. *Logical Methods in Computer Science*, 3, 2007.

[3] B. ten Cate and L. Segoufin. XPath, transitive closure logic, and nested tree walking automata. In *PODS'08*, p. 251–260. ACM, 2008.

Machine Translation with Multi Bottom-up Tree Transducers*

Fabienne Braune, Nina Seemann, Daniel Quernheim, and Andreas Maletti

Institute for Natural Language Processing, Universität Stuttgart
Azenbergstraße 12, 70174 Stuttgart, Germany
{braunefe, seemanna, daniel, maletti}@ims.uni-stuttgart.de

We have implemented a decoder for weighted multi bottom-up tree transducers [4, 7] using standard features inside the Moses framework [6]. Our weights currently represent probabilities. We can import weighted multi bottom-up tree transducers (mbots) into Moses and run the standard machine translation pipeline using an mbot as translation model. We will demonstrate the implementation and present a comparison and evaluation of our translation system using the typical synchronous context-free grammars [2] used in [5] as a baseline. We expect that mbots allow us to model target side discontinuities much better than Hiero, but we are currently fighting to increase coverage.

The mbot rules that we use are obtained with the help of external tools. First, the rules are extracted from an aligned bi-parsed corpus of sentence pairs using the method of [8]. Second, a standard input and output restriction generates the regular derivation tree language for each sentence pair, which is used to determine the inside weights used in em training [1]. The training step delivers the rule weights that best explain the training data. The obtained weighted rules are output in a format that is readable by Moses. Inside Moses all standard features (except backward translation) are used and feature weights are tuned using mert [3] as usual.

We will recall mbots as the main theoretical model and then present a detailed overview of all steps (both theoretical as well as practical) that are used to obtain our final translation system. Whenever feasible this will be illustrated on examples.

References

[1] D. B. Rubin A. P. Dempster, N. M. Laird. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[2] David Chiang. An introduction to synchronous grammars. In *Proc. 44th Annual Meeting of the Association for Computational Linguistics*. ACL, 2006. Part of a tutorial given with Kevin Knight.

[3] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley, New York, 1973.

[4] Joost Engelfriet, Eric Lilin, and Andreas Maletti. Composition and decomposition of extended multi bottom-up tree transducers. *Acta Inform.*, 46(8):561–590, 2009.

[5] Hieu Hoang and Philipp Koehn. Improved translation with source syntax labels. In *Proc. 5th Workshop Statistical Machine Translation and MetricsMATR*, pages 409–417. ACL, 2010.

*All authors were financially supported by the German Research Foundation (DFG) grant MA 4959/1-1.

[6] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. MOSES: Open source toolkit for statistical machine translation. In *Proc. 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180. ACL, 2007.

[7] Andreas Maletti. An alternative to synchronous tree substitution grammars. *Journal of Natural Language Engineering*, 17(2):221–242, 2011.

[8] Andreas Maletti. How to train your multi bottom-up tree transducer. In *Proc. 49th Annual Meeting Association for Computational Linguistics*, pages 825–834. ACL, 2011.

A Minimality-Preserving Regular Calculus of ADFAs

Johannes Bubenzer[1] and Bruce W.Watson[2]

[1]University of Potsdam [2]University of Stellenbosch

We present a unified algorithmic framework for the computation of mimimal acyclic deterministic finite state automata (MADFAs) using the regular operations union, intersection and difference operators. The framework is extensible to a lot more regular operators, but we will use these operations to exemplify its usage. The actual algorithms (for each regular operator given here) shows linear running time and memory complexity. For simplicity, we require acyclic finite-state automata (ADFA) as operands to these operations even if the framework could be extended such that the only requirement is that the *resulting* automaton is acyclic.

The framework is based upon a parallel depth-first traversal over pairs of states of the input automata. The actual algorithms for each operand differ only in two ways:

1. in the way the transitions for the traversal are selected, and
2. how final states are constructed in the output automaton.

The output automaton is computed during the traversal. Output states and transition are created if they are on a path to a final state, if they are not already present and if there is no equivalent state present. This is ensured using a *state map* and a *right-language register* known from acyclic minimization [1] and the compilation of ADFAs [2].

The following table shows the restrictions made on finality and transitons for union $\cup$, intersection $\cap$ and difference $-$. Operands are the two ADFAs $\mathcal{A}_1$ and $\mathcal{A}_2$. Q_i are the states, F_i are the final states and δ_i the transitions of automaton $\mathcal{A}_i$. We consider a state pair $\langle q_1, q_2\rangle \mid q_i \in Q_i$.

	finality	transitions
$\mathcal{A}_1 \cup \mathcal{A}_2$	$q_1 \in F_1 \vee q_2 \in F_2$	$\delta_1(q_1, a) \in Q_1 \vee \delta_2(q_2, a) \in Q_2$
$\mathcal{A}_1 \cap \mathcal{A}_2$	$q_1 \in F_1 \wedge q_2 \in F_2$	$\delta_1(q_1, a) \in Q_1 \wedge \delta_2(q_2, a) \in Q_2$
$\mathcal{A}_1 - \mathcal{A}_2$	$q_1 \in F_1 \wedge q_2 \notin F_2$	$\delta_1(q_1, a) \in Q_1$

(Note that the last line *does not* contain the clause $\delta_2(q_2, a) \notin Q_2$.)

Using a slightly modified algorithm, one can also compute the minimal result of a regular expression of $n > 2$ automata using only the given operators in one pass and in linear time. In this case the traversal runs over n automata and finality and traversed transitions of the resulting automaton are calculated in a similar way as before. The boolean expression of the table above just have to be combinted in the way indicated by the regular expression. Consider the regular expression $\mathcal{A}_1 \cup (\mathcal{A}_2 \cap \mathcal{A}_3)$. The finality of a triple state $\langle q_1, q_2, q_3\rangle \mid q_i \in Q_i$ can be determined by the expression $q_1 \in F_1 \vee (q_2 \in F_2 \wedge q_3 \in F_3)$. The transitions to consider are determined by $\delta_1(q_1, a) \in Q_1 \vee (\delta_2(q_2, a) \in Q_2 \wedge \delta_3(q_3, a) \in Q_3)$. This means that we can compute a significant subset of the regular expressions over ADFAs in one pass and linear time. We believe that the new algorithms pose the potential to speed-up automata operations in many real-word scenarios.

Further, we will briefly discuss the application of the algorithm to the case of weighted automata and the difficulties that arise if one tries to extend the framework to compute all regular expressions over ADFAs in one pass.

References

[1] Johannes Bubenzer. Minimization of Acyclic DFAs. In Jan Holub and Jan Žďárek, editors, *Proceedings of the Prague Stringology Conference*, pages 132–146, Czech Technical University in Prague, Czech Republic, 2011. Springer.

[2] Jan Daciuk, Bruce W. Watson, Stoyan Mihov, and Richard E. Watson. Incremental Construction of Minimal Acyclic Finite-State Automata. *Computational Linguistics*, 26(1):3–16, 2000.

Vanda
A Statistical Machine Translation Toolkit

Matthias Büchse*, Toni Dietze, Johannes Osterholzer,
Anja Fischer*, and Linda Leuschner*

Technische Universität Dresden

The aim of Machine Translation is to teach computers to translate sentences (and ultimately, texts) from one language into another. The subarea of Statistical Machine Translation (SMT) applies methods from Statistics and Machine Learning to automatically select a translation function that performs well on a sample of existing translations.

More precisely, an engineer devises a class of translation functions (called hypothesis space) and a loss function, and then she executes an algorithm that selects an element of the hypothesis space that has the least loss on the sample. Nowadays, the hypothesis space is often specified via a class of weighted synchronous grammars; i.e., each translation function corresponds to a grammar, and it maps every input sentence to a translation that has a derivation with highest weight.

Current SMT systems such as Hiero [1, 2], Moses [6], Joshua [7], or cdec [5] implement their respective specification directly in thousands of lines of code in languages such as Python, Java or C++. Sometimes the code adds substantial refinement to the specification, in form of poorly documented short-cuts and heuristics, making it hard to grasp the actual workings of the systems.

Moreover, the tasks of (i) preparing the data, (ii) selecting the translation function, and (iii) performing a translation require complex workflows, consisting of several programs to be run, each with its own command-line syntax. Because of the rapid development in the area of SMT and the prototypic nature of academic research, documentation of the workflows is often sub-optimal. In the absence of a workflow management system such as LoonyBin [3, 4], carrying out these workflows is error-prone, regardless of whether it is done manually or by means of a script.

We introduce our new SMT toolkit, Vanda, and the accompanying workflow management system, Vanda Studio. We hope to accomplish two objectives:

1. facilitate specifying core SMT algorithms in an algebraic manner, with a straightforward implementation; to this end, our solution is twofold:
   - we make full use of the theory of weighted automata and grammars as well as accompanying constructions and algorithms; for example, product constructions (weighted intersection), reduction constructions (removing useless rules), algorithms finding best derivations;
   - we use the functional programming language Haskell, which directly supports an algebraic way of thinking.
2. facilitate the cycle of "specify – take measurements – evaluate"; to this end, we adopt the concept of hyper-workflows from LoonyBin, i.e.,
   - a workflow is a formal object that can be edited graphically via Vanda Studio;

*financially supported by DFG VO 1011/6-1.

- a workflow element (e.g., a program) is a formal object as well, and the user interface provides a specification of every workflow element;
- a workflow can be transformed into an executable artifact, e.g., a shell script, that incorporates sanity checks to ensure correct interplay between workflow elements;
- a workflow can be transformed into different executable artifacts depending on the intended platform, e.g., a local workstation vs. a grid of computers;
- hyper-workflows introduce nondeterminism into workflows, allowing the engineer to specify alternatives; measurements are taken for each of these alternatives, and tables and diagrams for comparison can be generated automatically.

We note that Vanda Studio is tailored to the Vanda toolkit, but it is not limited to it.

In our presentation, we will show that Vanda and Vanda Studio attain our two abovementioned objectives by means of simple SMT tasks such as parsing and translation.

References

[1] David Chiang. Hierarchical phrase-based translation. *Comp. Ling.*, 33(2):201–228, 2007.

[2] David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. The hiero machine translation system: extensions, evaluation, and analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 779–786, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[3] Jonathan H. Clark and Alon Lavie. Loonybin: Keeping language technologists sane through automated management of experimental (hyper)workflows. In *Proceedings LREC 2010*, 2010.

[4] Jonathan H. Clark, Jonathan Weese, Byung Gyu Ahn, Andreas Zollmann, Qin Gao, Kenneth Heafield, and Alon Lavie. The machine translation toolpack for loonybin: Automated management of experimental machine translation hyperworkflows. *The Prague Bulletin of Mathematical Linguistics*, 2010.

[5] Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[6] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[7] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Jonathan Weese, and Omar F. Zaidan. Joshua: an open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 135–139, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

Deciding the Twins Property for Weighted Tree Automata over Extremal Semifields

Matthias Büchse* and Anja Fischer

Technische Universität Dresden

It is known that deterministic weighted tree automata are strictly less powerful than their general (nondeterministic) counterparts. The paper [1] contains a review of known sufficient conditions under which determinization is possible. One of these conditions requires that (i) the weights are calculated in an extremal semiring [4], (ii) there is a maximal factorization [3], and (iii) the weighted tree automaton has the twins property.

It has remained open whether the twins property is decidable, until Kirsten [2] gave an affirmative answer for a particular case: weighted string automata over the tropical semiring. He also showed that the decision problem is PSPACE-complete. In our presentation, we adapt and generalize Kirsten's proof: we show that the twins property is decidable for weighted tree automata over extremal semifields.

References

[1] Matthias Büchse, Jonathan May, and Heiko Vogler. 2010. Determinization of Weighted Tree Automata using Factorizations. *J. Autom. Lang. Comb.*, 15(3/4).

[2] Daniel Kirsten. 2012. Decidability, Undecidability, and PSPACE-Completeness of the Twins Property in the Tropical Semiring. *Theoretical Computer Science*, 420:56–63.

[3] Daniel Kirsten and Ina Mäurer. 2005. On the determinization of weighted automata. *J. Autom. Lang. Comb.*, 10:287–312.

[4] Bernd Mahr. 1984. Iteration and summability in semirings. *Annals of Discrete Mathematics*, 19:229–256.

*financially supported by DFG VO 1011/6-1.

Fuzziness in Automata Theory: Why? How?

Miroslav Ćirić and Jelena Ignjatović

University of Niš, Faculty of Sciences and Mathematics, Niš, Serbia
miroslav.ciric@pmf.edu.rs, jelena.ignjatovic@pmf.edu.rs

The aim of this talk is to explain why we study fuzzy automata and how we do it, i.e., to highlight the powerfull tools of the theory of fuzzy sets that we use in our research. In addition, we want to show how research in the theory of fuzzy automata affected our research in other areas of the theory of fuzzy sets.

We entered in the world of fuzziness when we crossed from the classical algebra and automata theory to the theory of fuzzy automata. Besides being considered as a natural generalization of ordinary automata and languages, fuzzy automata and related languages have also been studied as a means for bridging the gap between the precision of computer languages and vagueness and imprecision, which are frequently encountered in the study of natural languages. During the decades, they have got a wide field of applications. However, many authors thought mainly about the properties of ordinary automata which can be transferred to fuzzy automata. We found that the theory of fuzzy automata is not only simple translation of the results from the classical automata theory to the language of fuzzy sets, but it is possible to use powerful tools of the theory of fuzzy sets in the study of fuzzy automata.

The key point is that a fuzzy automaton can be regarded as a fuzzy relational system. It can be specified by a family $\{\delta_x\}_{x\in X}$ of fuzzy transition relations on the set of states A, indexed by the input alphabet X, and fuzzy subsets σ and τ of A, the fuzzy subsets of initial and terminal states. Inductively we define the composite fuzzy transition relations $\{\delta_u\}_{u\in X^*}$ by putting that δ_ε is the crisp equality, and $\delta_{ux} = \delta_u \circ \delta_x$, for $u \in X^*$, $x \in X$. Now, the fuzzy language recognized by the fuzzy automaton $\mathcal{A}$ is defined as a fuzzy subset $L_{\mathcal{A}}$ of X^* given by $L_{\mathcal{A}}(u) = \sigma \circ \delta_u \circ \tau$, for $u \in X^*$.* This way of representing fuzzy automata, and fuzzy languages that they recognize, enables to study fuzzy automata using fuzzy relational calculus, and to express many problems through fuzzy relation equations and inequalities. Fuzzy relational calculus and fuzzy relation equations and inequalities have been widely used in our research.

Previously, fuzzy relational calculus and fuzzy relation equations and inequalities were used in the theory of fuzzy automata only by few authors – by Peeva [18, 19, 20], Bělohlávek [1], and Li and Pedrycz [17]. Surprisingly, such approach has not been used for ordinary nondeterministic automata, although their behavior can be expressed in terms of the calculus of two-valued relations. Probably, the reason for this is the fact that nondeterministic automata are predominantly considered from the perspective of the graph theory, and not from the perspective of the algebra of relations. A little bit similar approach has been used for weighted automata over a semiring, whose behavior is defined through the calculus of matrices with entries in the underlying semiring (cf. [8]). However, matrices over a semiring do not possess some very important properties of ordinary and fuzzy relations, and their use in the study of weighted automata is not as fruitful as the use of

*Here X^* denotes the monoid of all words over X, $\varepsilon \in X^*$ is the empty word, and $\circ$ denotes the compositions of two fuzzy relations, of a fuzzy set and a fuzzy relation and two fuzzy sets, defined in the usual way over a residuated lattice or a lattice-ordered monoid.

fuzzy relations in the study of fuzzy automata.

We will briefly explain how we used fuzzy relational calculus and fuzzy relation equations and inequalities in solving the fundamental problems of the theory of fuzzy automata: *determinization* [2, 10, 13, 16], *equivalence* [5, 6], and *state reduction* [7, 21]. Moreover, we will explain how the study of fuzzy automata affected our study of fuzzy relations [3, 4, 11] and fuzzy relation equations and inequalities [9, 12, 14, 15].

References

[1] R. Bělohlávek, Determinism and fuzzy automata, Inform Sci. 143 (2002) 205–209.

[2] M. Ćirić, M. Droste, J. Ignjatović, H. Vogler, Determinization of weighted finite automata over strong bimonoids, Inform. Sci. 180 (2010) 3497–3520.

[3] M. Ćirić, J. Ignjatović, S. Bogdanović, Fuzzy equivalence relations and their equivalence classes, Fuzzy Sets Syst. 158 (2007) 1295–1313.

[4] M. Ćirić, J. Ignjatović, S. Bogdanović, Uniform fuzzy relations and fuzzy functions, Fuzzy Sets and Systems 160 (2009) 1054–1081.

[5] M. Ćirić, J. Ignjatović, N. Damljanović, M. Bašić, Bisimulations for fuzzy automata, Fuzzy Sets Syst. 186 (2012) 100139.

[6] M. Ćirić, J. Ignjatović, I. Jančić, N. Damljanović, Computation of the greatest simulations and bisimulations between fuzzy automata, Fuzzy Sets Syst., submitted.

[7] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, Fuzzy relation equations and reduction of fuzzy automata, J. Comput. Syst. Sci. 76 (2010) 609–633.

[8] M. Droste, W. Kuich, H. Vogler (eds.), Handbook of Weighted Automata, Monographs in Theoretical Computer Science, EATCS Series, Springer-Verlag, 2009.

[9] J. Ignjatović, M. Ćirić, Weakly linear systems of fuzzy relation inequalities and their applications: A brief survey, Filomat 26:2 (2012) 1–35.

[10] J. Ignjatović, M. Ćirić, S. Bogdanović, Determinization of fuzzy automata with membership values in complete residuated lattices, Inform. Sci. 178 (2008) 164–180.

[11] J. Ignjatović, M. Ćirić, S. Bogdanović, Fuzzy homomorphisms of algebras, Fuzzy Sets Syst. 160 (2009), 2345–2365.

[12] J. Ignjatović, M. Ćirić, S. Bogdanović, On the greatest solutions to weakly linear systems of fuzzy relation inequalities and equations, Fuzzy Sets Syst. 161 (2010) 3081–3113.

[13] J. Ignjatović, M. Ćirić, S. Bogdanović, T. Petković, Myhill-Nerode type theory for fuzzy languages and automata, Fuzzy Sets Syst. 161 (2010) 1288-1324.

[14] J. Ignjatović, M. Ćirić, N. Damljanović, I. Jančić, Weakly linear systems of fuzzy relation inequalities: The heterogeneous case, Fuzzy Sets Syst. (2011), doi: 10.1016/j.fss.2011.11.011.

[15] J. Ignjatović, M. Ćirić, B. De Baets: General method for solving equations and inequalities defined by residuated functions, to appear.

[16] Z. Jančić, J. Ignjatović, M. Ćirić, An improved algorithm for determinization of weighted and fuzzy automata, Inform. Sci. 181 (2011) 1358–1368.

[17] Y. M. Li, W. Pedrycz, Fuzzy finite automata and fuzzy regular expressions with membership values in lattice ordered monoids, Fuzzy Sets Syst. 156 (2005) 68–92.

[18] K. Peeva, Behaviour, reduction and minimization of finite L-automata, Fuzzy Sets Syst. 28 (1988) 171–181.

[19] K. Peeva, Finite L-fuzzy machines, Fuzzy Sets Syst. 141 (2004) 415–437.

[20] K. Peeva, Y. Kyosev, Fuzzy Relational Calculus: Theory, Applications, and Software, World Scientific, 2004.

[21] A. Stamenković, M. Ćirić, J. Ignjatović, Reduction of fuzzy automata by means of fuzzy quasi-orders, Inform. Sci., submitted.

Embracing divergence: an approach to generalizing weighted automata to infinite words with applications in quantum simulation

Gregory Crosswhite

University of Queensland
g.crosswhite@uq.edu.au

There is a fundamental difficulty in generalizing weighted automata to the case of infinite words: in general the infinite sum-of-products from which the weight of a given word is derived will diverge. Of course, one approach in dealing with this is to simply restrict oneself to the class of semirings for which an infinite sum-of-products is always guaranteed to converge, thus excluding the possibility of divergences altogether. In our presentation we shall describe an alternative approach: rather than avoiding divergences, we *embrace* them in order to learn more about them. Specifically, we start with an arbitrary semiring S, but rather than defining a language to be a formal power series over words with coefficients in S we instead define it to be a formal power series with coefficients in $S^{\mathbb{N}}$, the map from natural numbers to S. Put another way, to each infinite word is now associated a sequence of partial sums that gives us information about exactly how its weight does or does not diverge — e.g., assuming it diverges, whether it does so polynomially, exponentially, etc. In this setting we can immediately generalize Büchi automata in the same fashion that we do automata for finite languages — namely, by labeling the transitions with weights from an arbitrary semiring — without having to worry about undefined values coming out of the (no-longer) infinite sum.

Having described this approach, we shall then show how it has turned out to be incredibly useful in the field of quantum simulation (that is, the computational simulation of quantum systems by classical computers). When studying an infinite quantum system the goal is frequently to get information about the scaling properties of finite systems, and so a function from a size (which is a natural number when dealing with quantized particles) to a given quantity of interest is exactly what is needed. Furthermore, we have learned empirically that what we call "infinite matrix product states"/"infinite matrix product operators" — which are essentially weighted bi-directional Buchi automata/transducers by another name — provide a means of representing quantum states/observables that is powerful yet relatively concise. Most importantly, there exist efficient algorithms both to derive the values of observable quantities from matrix product states and also to solve for the matrix product state that has the minimal energy for a particular system. For these reasons, "infinite matrix product states" have become an important tool in the last decade for modeling quantum systems.

In sum, in our presentation we shall show that not only does there exist a generalization of weighted automata that works for arbitrary semirings, but that this construction has incredibly important applications within the field of quantum simulation.

Bisimulations for weighted automata

Nada Damljanović[1], Miroslav Ćirić[2], and Jelena Ignjatović[2]

[1]University of Kragujevac
Technical faculty in Čačak, Čačak,
Serbia
nada@tfc.kg.ac.rs

[2] University of Niš
Faculty of Sciences and Mathematics,
Niš, Serbia
jelena.ignjatovic@pmf.edu.rs,
miroslav.ciric@pmf.edu.rs

An important problem of automata theory is to determine whether two given automata are equivalent. For deterministic automata this problem is solvable in polynomial time, but for nondeterministic, fuzzy and weighted automata it is computationally hard. It is also desirable to express the equivalence of automata as a relation between their states, if possible, or find some relation between states which implies the equivalence. The equivalence of two deterministic automata can be expressed in terms of relationships between their states, but in the case of nondeterministic, fuzzy and weighted automata the problem is more complicated, and we can only examine various relations which imply the equivalence.

It is generally agreed that the best way to model the equivalence of automata is the concept of bisimulation. They give a close enough approximation of the equivalence and are efficiently computable. Bisimulations were introduced in concurrency theory, and independently, in set theory and modal logic, and nowadays, they are successfully employed in many areas of computer science and mathematics. They are used both to model equivalence between automata and to reduce the number of states of automata.

Simulations and bisimulations for fuzzy automata over complete residuated lattices have been recently studied in [6, 7]. They have been defined by systems of fuzzy relation inequalities and equations which have been studied in a more general context in [9, 11, 12]. The methods developed within the theory of fuzzy automata are applied here to weighted automata over additively idempotent semirings.

For given weighted automata $\mathcal{A}$ and $\mathcal{B}$ over an additively idempotent semiring we introduce two types of simulation relations (forward and backward simulations) and four types of bisimulation relations (forward, backward, forward-backward, and backward-forward bisimulations) between them. We provide efficient algorithms for computing the greatest simulations and bisimulations of any of the above-mentioned types between $\mathcal{A}$ and $\mathcal{B}$. The algorithms are based on the method developed in [11], which boils down to the computing of the greatest post-fixed point, contained in a given relation, of an isotone function on the lattice of relations.

Especially, we study bisimulations that are uniform relations. The concept of a uniform relation has been introduced in [5] as a model of equivalence between elements of two different sets. Here we prove that a uniform relation between weighted automata $\mathcal{A}$ and $\mathcal{B}$ is a forward bisimulation if and only if its kernel and co-kernel are forward bisimulation equivalence relations on $\mathcal{A}$ and $\mathcal{B}$ and there is an isomorphism between factor fuzzy automata with respect to these equivalence relations. As a consequence we get that weighted automata $\mathcal{A}$ and $\mathcal{B}$ are UFB-equivalent, i.e., there is a uniform forward bisimulation between them, if and only if there is a special isomorphism between the factor weighted automata of $\mathcal{A}$ and $\mathcal{B}$ with respect to their greatest forward bisimulation equivalence relations.

Note that bisimulations between weighted automata have already been studied by several authors (cf. [2, 3, 4]), but our approach is completely different.

References

[1] M. Ćirić, J. Ignjatovic, M. Bašić, I. Jančić, Nondeterministic automata: equivalence, bisimulations, and uniform relations, submitted to Information Sciences.

[2] S. L. Bloom, Z. Ésik, Iteration Theories: The Equational Logic of Iterative Processes, EATCS Monographs on Theoretical Computer Science, Springer, Berlin-Heilderberg, 1993.

[3] P. Buchholz, Bisimulation relations for weighted automata, Theoretical Computer Science 393 (2008) 109–123.

[4] Z. Ésik, A. Maletti, Simulation vs. Equivalence, CoRR abs/1004.2426 (2010).

[5] M. Ćirić, J. Ignjatović, S. Bogdanović, Uniform fuzzy relations and fuzzy functions, Fuzzy Sets and Systems 160 (2009) 1054–1081.

[6] M. Ćirić, J. Ignjatović, N. Damljanović, M. Bašić, Bisimulations for fuzzy automata, Fuzzy Sets and Systems 186 (2012) 100-139.

[7] M. Ćirić, J. Ignjatović, I. Jančić, N. Damljanović, Computation of the greatest simulations and bisimulations between fuzzy automata, submitted to Fuzzy Sets and Systems.

[8] M. Droste, W. Kuich, H. Vogler (eds.), Handbook of Weighted Automata, Springer, Berlin, 2009.

[9] J. Ignjatović, M. Ćirić: Weakly linear systems of fuzzy relation inequalities and their applications: A brief survey, Filomat 26(2) (2012) 1–35.

[10] J. Ignjatović, M. Ćirić, S. Bogdanović, Fuzzy homomorphisms of algebras, Fuzzy Sets and Systems 160 (2009) 2345–2365.

[11] J. Ignjatović, M. Ćirić, S. Bogdanović, On the greatest solutions to certain systems of fuzzy relation inequalities and equations, Fuzzy Sets and Systems 161 (2010) 3081–3113.

[12] J. Ignjatović, M. Ćirić, N. Damljanović, I. Jančić, Weakly linear systems of fuzzy relation inequalities: The heterogeneous case, Fuzzy Sets and Systems (2011), doi:10.1016/j.fss.2011.11.011.

Weighted Tree Automata over Valuation Monoids and Their Characterization by Weighted Logics

Manfred Droste[1], Doreen Götze[1], Steffen Märcker[2], and Ingmar Meinecke[1]

[1]Institut für Informatik,
Universität Leipzig
D-04109 Leipzig, Germany
{droste,goetze,meinecke}
@informatik.uni-leipzig.de

[2]Institut für Theoretische Informatik,
Technische Universität Dresden
01187 Dresden, Germany
steffen.maercker@tu-dresden.de

Trees or terms are one of the most fundamental concepts both in mathematics and in computer science. Thus, weighted tree automata gained a lot attention during the last decades. Usually, the weight of a run is computed locally by a binary operation. Recently, Chatterjee, Doyen, and Henzinger presented a new approach for words with real numbers as weights. The weight of a run is now determined by a global valuation function. An example of such a valuation function is the average of the weights. This idea was generalized by Droste and Meinecke to a more general weight structure, called valuation monoid, and the class of behaviors of these weighted automata was characterized by suitable weighted MSO logics.

We adapt this concept to weighted tree automata. A tree valuation monoid is an additive monoid equipped with a tree valuation function which maps a tree of weights to a single weight. Then the automaton uses the tree valuation function to calculate the weight of every run and sums up these weights.

In our main result we establish a characterization of the behaviors of the weighted tree automata over valuation monoids by three fragments of the weight- ed MSO logic. Which fragment can be used depends on the properties of the underlying product tree valuation monoid. The restrictions on the fragments are purely syntactic and refer only to the use of first-order universal quantification and conjunction.

Our main result generalizes the respective result about the connection between semiring-weighted tree automata and weighted MSO logic from Droste and Vogler, and the result about the connection between weighted automata over finite words and valuation monoids and weighted MSO logic from Droste and Meinecke.

Büchi Conditions for Generalized Energy Automata

Uli Fahrenberg[1], Axel Legay[1], and Karin Quaas[2]

[1]Irisa/INRIA Rennes, France [2]Universität Leipzig, Germany

Recently, formal methods for describing and analyzing *energy behavior*, i.e. consumption and production of energy, has attracted a lot of interest, see, e.g., [3, 1, 5, 6, 2]. In this paper we make the important observation that models for analyzing energy behavior should describe energy *transformations*, that is, use a functional description to model how energy levels change from one system state to another. We introduce *generalized energy automata*, i.e., finite automata whose edges are labeled by energy functions that define how energy levels evolve during transitions. This functional model generalizes all approaches mentioned above and, as an application of semiring-weighted automata, opens interesting new connections to the theory of weighted automata [4].

We show that for our functional model, the Büchi acceptance problem (of deciding whether there exists a run which infinitely often passes through a pre-described set of accepting states) is decidable. Our approach relies on the *transitive closure* of energy automata, a quite natural construction in our functional framework, which computes energy functions between any two system states. We also define a notion of *energy bisimilarity*, which characterizes pairs of states with similar energy behaviors, and which we show to be decidable.

References

[1] Patricia Bouyer, Uli Fahrenberg, Kim G. Larsen, and Nicolas Markey. Timed automata with observers under energy constraints. In Karl Henrik Johansson and Wang Yi, editors, *HSCC*, pages 61–70. ACM, 2010.

[2] Krishnendu Chatterjee and Laurent Doyen. Energy and mean-payoff parity Markov decision processes. In Filip Murlak and Piotr Sankowski, editors, *MFCS*, volume 6907 of *LNCS*, pages 206–218. Springer, 2011.

[3] Aldric Degorre, Laurent Doyen, Raffaella Gentilini, Jean-François Raskin, and Szymon Torunczyk. Energy and mean-payoff games with imperfect information. In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *LNCS*, pages 260–274. Springer, 2010.

[4] Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.

[5] Uli Fahrenberg, Line Juhl, Kim G. Larsen, and Jiří Srba. Energy games in multiweighted automata. In *ICTAC*, volume 6916 of *LNCS*, pages 95–115. Springer, 2011.

[6] Karin Quaas. On the interval-bound problem for weighted timed automata. In Adrian Horia Dediu, Shunsuke Inenaga, and Carlos Martín-Vide, editors, *LATA*, volume 6638 of *LNCS*, pages 452–464. Springer, 2011.

Size-Change Termination and Arctic Matrix Monoids

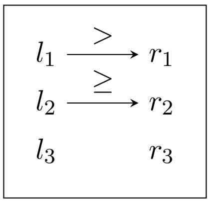
Bertram Felgenhauer[1], Sibylle Schwarz[2], and Johannes Waldmann[3]

[1]Universität Innsbruck
[2]WH Zwickau
[3]HTWK Leipzig

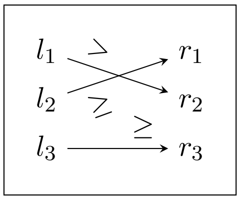
In this note, we show a connection between a boundedness problem for matrix products over the arctic semi-ring $(\{-\infty\} \cup \mathbb{N}, \max, +)$ and the "size-change" method for proving termination of programs [3].

This method translates an input program R into a set of size-change graphs that represent relations between argument sizes in function calls. For each rule $f(l_1, \dots, l_m) \to t$ in R, and for each subterm $g(r_1, \dots, r_n)$ of t, a directed, labelled, bipartite graph with m input vertices and n output vertices is constructed that contains edges $l_i \to r_j$ with label $>$ or $\geq$, expressing strong or weak decrease of the argument size from l to r in the corresponding positions.

For instance, the program rule $F(x+1, y+1, z) \to F(x, y+1, F(y+1, x, z))$ is translated to two size-change-graphs (one for every occurrence of F in the right-hand-side of the rule) given in the boxes:



$$\begin{pmatrix} 1 & -\infty & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & -\infty \end{pmatrix}$$



$$\begin{pmatrix} -\infty & 1 & -\infty \\ 0 & -\infty & -\infty \\ -\infty & -\infty & 0 \end{pmatrix}$$

We observe that every size-change-graph can be represented as matrix with entries in $\{-\infty, 0, 1\}$ where 0 represents $\geq$-labelled edges, 1 represents $>$-labelled edges, and $-\infty$ represents non-existent edges in G.

A sequence of nested function calls corresponds to a composition of size-change graphs, and multiplication of matrices in the arctic semi-ring, respectively. If each infinite sequence of size-change graphs contains a path with infinitely many $>$-edges, then the program terminates by contradiction to the well-foundedness of the parameter domain (in the example, $\mathbb{N}$).

The corresponding property for products of arctic matrices from a set S is that each infinite sequence has a suffix for which norms of products of prefixes are unbounded:

Definition 1. A sequence $(G_i)_{i<\omega} \in S^\omega$ is *universally tail-unbounded* if there is some k such that
$\sup\{\| \prod_{i=k}^{k'} G_i \| \mid k' \geq k\}$ is infinite, where $\|M\|$ is the maximal entry of M.

In general, not all sequences of matrices are meaningful for the program. Consequently, S^ω needs to be refined as the set of infinite paths through a finite automaton that has subprogram names as states, and transitions are labelled by matrices. This is similar to meta-transitions in [2].

For the problem of deciding universal tail-unboundedness, result from the size-change literature can be translated. The problem is PSPACE-complete [3], and a generalization to arctic integers $(\{-\infty\} \cup \mathbb{Z})$ is undecidable, but becomes again decidable for "fan-in one" (each matrix column contains at most one finite entry) [1]. For proving program termination automatically, it is of practical interest to find more (and efficiently) decidable restrictions of the problem.

References

[1] Amir M. Ben-Amram. Size-change termination with difference constraints. *ACM Trans. Program. Lang. Syst.*, 30(3), 2008.

[2] Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In Susanne Albers and Jean-Yves Marion, editors, *STACS*, volume 3 of *LIPIcs*, pages 589–600. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[3] Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. The size-change principle for program termination. In *POPL*, pages 81–92, 2001.

Deciding the Value 1 Problem of Probabilistic Leaktight Automata *

Nathanaël Fijalkow[1], Hugo Gimbert[2], and Youssouf Oualhadj[3]

[1]LIAFA, Université Denis Diderot - Paris 7, France
nath@liafa.jussieu.fr

[3]LaBRI, Université Bordeaux 1, France.
youssouf.oualhadj@labri.fr

[2]LaBRI, CNRS, Bordeaux, France
hugo.gimbert@labri.fr

The value 1 problem is a decision problem for probabilistic automata over finite words: given a probabilistic automaton $\mathcal{A}$, are there words accepted by $\mathcal{A}$ with probability arbitrarily close to 1?

This problem was proved undecidable recently [2]. However, it shows some similarities with another problem, the boundedness problem for distance automata. Both computational models are weighted automata, but the first compute in the real semiring $(\mathbb{R}, +, \cdot)$, while the second compute in the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +)$.

The boundedness problem for distance automata was shown decidable by Hashiguchi [3, 4, 5, 6]; an algebraic algorithm was proposed by Leung [8, 9], whose correctness proof relies on algebraic techniques developed by Simon [10, 11]. The decidability result was later extended to nested desert distance automata and better understood by Kirsten [7].

We have transferred the underlying ideas to probabilistic automata. Our main contribution is to introduce a new class of probabilistic automata, called *leaktight automata*, for which the value 1 problem is shown decidable (and PSPACE-complete). We construct an algorithm based on the computation of a monoid abstracting the behaviors of the automaton, following Leung's algorithm. As in the case of distance automata, the correctness proof relies on the algebraic techniques developed by Simon, namely factorization and decomposition trees.

A full version can be found as technical report [1].

References

[1] Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. A class of probabilistic automata with a decidable value 1 problem. *CoRR*, abs/1104.3055, 2011.

[2] Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *International Colloquium on Automata, Languages and Programming*, pages 527–538, 2010.

[3] Kosaburo Hashiguchi. Limitedness theorem on finite automata with distance functions. *Journal of Computuer and System Sciences*, 24(2):233–244, 1982.

[4] Kosaburo Hashiguchi. Regular languages of star height one. *Information and Control*, 53(3):199–210, 1982.

*Supported by the ANR projet FREC, 2010 BLAN 0202 02 FREC

[5] Kosaburo Hashiguchi. Representation theorems on regular languages. *Journal of Computuer and System Sciences*, 27(1):101–115, 1983.

[6] Kosaburo Hashiguchi. Relative star height, star height and finite automata with distance functions. In *Formal Properties of Finite Automata and Applications*, pages 74–88, 1988.

[7] Daniel Kirsten. Distance desert automata and the star height problem. *Informatique Théorique et Applications*, 39(3):455–509, 2005.

[8] Hing Leung. An algebraic method for solving decision problems in finite automata theory, 1987.

[9] Hing Leung. Limitedness theorem on finite automata with distance functions: An algebraic proof. *Theoretical Computer Science*, 81(1):137–145, 1991.

[10] Imre Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94, 1990.

[11] Imre Simon. On semigroups of matrices over the tropical semiring. *Informatique Théorique et Applications*, 28(3-4):277–294, 1994.

Weighted Tree Automata and Branching Transitive Closure Logics *

Zoltán Fülöp[1] and Heiko Vogler[2]

[1]Department of Foundations of Computer Science, University of Szeged
fulop@inf.u-szeged.hu

[2]Faculty of Computer Science, Technische Universität Dresden
Heiko.Vogler@tu-dresden.de

We report on the main result of [FV11], which is a generalization of [BGMZ10, Theorem 10] to weighted tree languages, but only for commutative semirings. We introduce the *branching transitive closure operator* on monadic second-order logic formulas for trees, where the branching corresponds in a natural way to the branching inherent in trees. Then for weighted tree languages over an arbitrary commutative semiring, we prove that recognizability is equivalent to definability by formulas of any of the following forms: (i) a branching transitive closure operator is applied to some body formula or (ii) a second-order $\exists$ quantifier followed by a first-order $\forall$ quantifier is applied to some body formula. In both cases the body formula is either (1) a Boolean first-order step-formulas enriched by modulo counting or (2) a Boolean monadic-second order step-formula.

References

[BGMZ10] B. Bollig, P. Gastin, B. Monmenge, and M. Zeitoun. Pebble weighted automata and transitive closure logics. In *Proc. of ICALP 2010*, LNCS 6199, pp. 587–598. Springer-Verlag, 2010.

[FV11] Z. Fülöp and H. Vogler. Weighted tree automata and branching transitive closure logics. submitted for publication, also cf. http://www.inf.tu-dresden.de/index.php?node_id=1571&ln=de, 2011.

*This research was financially supported by the programs TÁMOP-4.2.2/08/1/2008-0008 and TÁMOP-4.2.1/B-09/1/KONV-2010-0005 of the Hungarian National Development Agency

Weighted Automata as Semantic Representation

Markus Huber and Christian Kölbl

Universität Augsburg, Institut für Informatik
{markus.huber, christian.koelbl}@informatik.uni-augsburg.de

In [1] and [2] a mathematical formalism called weighted feature-values-relation (*wFVR*) which can be used to represent semantic information within a spoken dialogue system is described. Every output of a speech recogniser can be transformed into wFVRs [7] and incorporated into an information state which is a wFVR by itself. Therefor one of the authors introduced an update operation [1] which uses confidence numbers and probability vectors to combine old and new information in a clever way so that no information gets lost. While this operation processes only the weights, the structural update is described in [4]. For practical application we showed in [3] how wFVRs can be implemented through weighted finite-state-transducers (*FSTs*). We realised the update and unification by standard operations on FSTs [5] over the confidence-probability semiring [6]. However we did not use the language of the FST and neither the encoded information from the FVR in the structure of the resulting FST. So neither the notion of paths nor the weight of a path could be reasonable interpreted.

In this paper we exploit the use of weighted automata for representing the semantic information contained in wFVRs. We use the language of an automata to describe the structure of an FVR and we use the weights of paths to represent the weights of a wFVR. One advantage of this approach is that we no longer rely on the very structure of the automata which means we can use any operation which respects the automata's language. One disadvantage is that the update operation is more difficult to implement.

References

[1] Markus Huber. *Semantische Informationsbearbeitung unter Berücksichtigung von Konfidenz-Zahlen.* Diplomarbeit in Mathematik. Katholische Universität Eichstätt-Ingolstadt, 2009.

[2] Markus Huber, Christian Kölbl, Robert Lorenz, Ronald Römer, and Günther Wirsching. Semantische Dialogmodellierung mit gewichteten Merkmal-Werte-Relationen. In Rüdiger Hoffmann, editor, *Elektronische Sprachsignalverarbeitung 2009. Tagungsband der 20. Konferenz. Dresden, 21. bis 23. September 2009*, volume 53 of *Studientexte zur Sprachkommunikation*, pages 25–32. TUDpress, September 2009.

[3] Markus Huber, Christian Kölbl, and Günther Wirsching. Gewichtete endliche Transduktoren als semantische Träger. In Bernd Kröger and Peter Birkholz, editors, *Elektronische Sprachsignalverarbeitung 2011. Tagungsband der 22. Konferenz. Aachen, 28. bis 30. September 2011*, volume 61 of *Studientexte zur Sprachkommunikation*, pages 176–183. TUDpress, September 2011.

[4] Alexandra Matt. *Unifikation semantischer Strukturen.* Diplomarbeit in Mathematik. Katholische Universität Eichstätt-Ingolstadt, 2009.

[5] Mehryar Mohri. Weighted Automata Algorithms. In Manfred Droste, Werner Kuich,

and Heiko Vogler, editors, *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science, pages 213–254. 2009.

[6] Günther Wirsching, Christian Kölbl, and Markus Huber. The confidence-probability semiring. Technical report, Angewandte Informatik, Universität Augsburg, 2010.

[7] Günther Wirsching, Christian Kölbl, and Markus Huber. Zur Logik von Bestenlisten in der Dialogmodellierung. In Bernd Kröger and Peter Birkholz, editors, *Elektronische Sprachsignalverarbeitung 2011. Tagungsband der 22. Konferenz. Aachen, 28. bis 30. September 2011*, volume 61 of *Studientexte zur Sprachkommunikation*, pages 309–316. TUDpress, September 2011.

Subsystems of fuzzy transition systems

Jelena Ignjatović[1], Miroslav Ćirić[1], and Vesna Simović[2]

[1]University of Niš, Faculty of Sciences and Mathematics, Niš, Serbia

miroslav.ciric@pmf.edu.rs,
jelena.ignjatovic@pmf.edu.rs

[2]High Economic School for Professional Studies Peć in Leposavić, Serbia

vesna.simovic78@gmail.com

A common problem that arises in many applications of transition systems and automata is to identify those sets of states that are closed under all transition relations, i.e., which with each of its states also contain all the states that are accessible from it. In the case of deterministic transition systems, if they are regarded as unary algebras, such sets are precisely subalgebras of these algebras. Also, it is often needed to identify sets of states which are closed under reverse transitions, i.e., which with each of its states also contain all the states that are coaccessible to it. Sets of states closed both for transitions and reverse transitions are actually components of decompositions of a transition system into a disjoint union of smaller transition systems with the property that there are no transitions between states from different components. Such decompositions are known as *direct sum decompositions* and are used when we want to detect and eliminate useless states and transitions of a transition system (cf. [3, 4, 6]).

When dealing with a fuzzy transition system $\mathcal{T}$, fuzzy sets of states which are closed under all fuzzy transition relations are called *subsystems* of $\mathcal{T}$. They were introduced and studied by Malik, Mordeson and Sen in [9], who described some of their fundamental properties (see also [10]). Later, subsystems were investigated by Das in [7], who showed that they form a topological closure system and identified a number of their topological properties. From a topological point of view, subsystems were also discussed in [12]. All these papers dealt with fuzzy transition systems over the Gödel structure. Here we study subsystems in a more general context, for fuzzy transition systems over a complete residuated lattice. We also introduce and examine *reverse subsystems*, which are defined as those fuzzy sets of states which are closed under reverse fuzzy transition relations, and *double subsystems*, which are defined as fuzzy sets of states which are both subsystems and reverse subsystems. We show that all three types of subsystems can be considered as solutions to some particular systems of fuzzy relation inequalities and equations. Especially important role in our research play the fuzzy quasi-order Q_δ and and fuzzy equivalence E_δ generated by fuzzy transition relations, which can be efficiently computed using the well-known algorithms for computing the transitive closure of a fuzzy relation. In particular, we characterize subsystems, reverse subsystems and double subsystems respectively as eigen fuzzy sets of Q_δ, Q_δ^{-1} and E_δ (in the sense of Sanchez [11]), and we also characterize them as linear combinations of aftersets and foresets of Q_δ and equivalence classes of E_δ. We also show that subsystems, reverse subsystems and double subsystems of a fuzzy transition system $\mathcal{T}$ form both closure and opening systems in the lattice of fuzzy subsets of A, where A is the set of states of $\mathcal{T}$, and we provide efficient procedures for computing related closures and openings of an arbitrary fuzzy subset of A. These procedures simply boil down to computing the fuzzy quasi-order Q_δ or the fuzzy equivalence E_δ (cf. [8]).

The obtained results generalize the results from [7, 9, 10, 12] on subsystems of fuzzy transition systems over the Gödel structure, and the results from [3, 4, 5, 6] on subsystems, reverse subsystems, double subsystems and direct sum decompositions of ordinary transition systems. Moreover, they are closely related to results from [1, 2] on closure and opening operators defined by fuzzy relations. Note that Das in [7] proposed a method for computing closures related to subsystems, but his approach requires computation of all composite fuzzy transition relations, whose number can be exponential in the number of states, and for some structures of truth values this number may even be infinite. This makes his approach computationally inefficient. Our approach does not require computation of composite fuzzy transition relations but boils down to computing the transitive closure of a fuzzy relation, and our algorithms work in polynomial time.

References

[1] U. Bodenhofer, A unified framework of opening and closure operators with respect to arbitrary fuzzy relations, Soft Computing 7 (2003) 220–227.

[2] U. Bodenhofer, M. De Cock, E. E. Kerre, Openings and closures of fuzzy preorderings: theoretical basics and applications to fuzzy rule-based systems, International Journal of General Systems 32 (2003), 343–360.

[3] M. Ćirić, S. Bogdanović, Lattices of subautomata and direct sum decompositions of automata, Algebra Colloquium 6 (1) (1999) 71–88.

[4] M. Ćirić , S. Bogdanović, J. Kovačević, Direct sum decompositions of quasi-ordered sets and their applications, Filomat 12:1 (1998) 65–82.

[5] M. Ćirić, S. Bogdanović, T. Petković, The lattice of positive quasi-orders on an automaton, Facta Universitatis (Niš), Series Math. and Inform. 11 (1996) 143–156.

[6] M. Ćirić, S. Bogdanović, T. Petković, The lattice of subautomata of an automaton - A survey, Publ. Inst. Math. (Beograd) 64 (78) (1998) 165–182.

[7] P. Das, A fuzzy topology associated with a fuzzy finite state machine, Fuzzy Sets and Systems 105 (1999) 469–479.

[8] J. Ignjatović, M. Ćirić, V. Simović, Fuzzy relation equations and subsystems of fuzzy transition systems, Knowledge-Based Systems (2012) doi: 10.1016/j.knosys.2012.02.008.

[9] D. S. Malik, J. N. Mordeson, M. K. Sen, On subsystems of a fuzzy finite state machine, Fuzzy Sets and Systems 68 (1994) 83–92.

[10] J. N. Mordeson, D. S. Malik, Fuzzy Automata and Languages: Theory and Applications, Chapman & Hall/CRC, Boca Raton, London, 2002.

[11] E. Sanchez, Resolution of eigen fuzzy sets equations, Fuzzy Sets and Systems 1 (1978) 69–74.

[12] A. K. Srivastava, S. P. Tiwari, On relationships among fuzzy approximation operators, fuzzy topology, and fuzzy automata, Fuzzy Sets and Systems 138 (2003) 197–204.

Fuzzy automata: Determinization using simulations

Ivana Jančić, Zorana Jančić, Jelena Ignjatović, and Miroslav Ćirić

University of Niš, Faculty of Sciences and Mathematics,
Niš, Serbia
ivanajancic84@gmail.com, zoranajancic329@gmail.com
jelena.ignjatovic@pmf.edu.rs, miroslav.ciric@pmf.edu.rs

Two types of simulation fuzzy relations for fuzzy automata, forward and backward simulations, have been introduced recently in [1] and studied in [1, 2], and more general concepts of a weak forward simulation and a weak backward simulation have been introduced and studied in [4]. In this talk we present the results from [3], on determinization of fuzzy automata over complete residuated lattices using simulation fuzzy relations.

For a given fuzzy automaton $\mathcal{A}$ over a complete residuated lattice, and a forward simulation φ on $\mathcal{A}$, we construct the automaton $\mathcal{A}_\varphi$, which is crisp-deterministic and language-equivalent to the original fuzzy automaton $\mathcal{A}$. In particular, when φ is the equality relation on the set of states of $\mathcal{A}$, we obtain the Nerode automaton $\mathcal{A}_N$, which was previously used in determinization of fuzzy and weighted automata. When φ is taken to be a fuzzy quasi-order, we show that the automaton $\mathcal{A}_\varphi$ has smaller number of states than the Nerode automaton $\mathcal{A}_N$. The smallest number of states is achieved when φ is taken to be the greatest forward simulation on $\mathcal{A}$.

We provide an efficient algorithm which computes the automaton $\mathcal{A}_\varphi$ for a given fuzzy finite automaton $\mathcal{A}$ and a forward simulation φ on $\mathcal{A}$, and we show by example that in some cases this algorithm produces the minimal crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$. To achieve that $\mathcal{A}_\varphi$ has as small as possible number of states, we should take φ to be the greatest forward simulation on $\mathcal{A}$, which can be computed in a polynomial time using the algorithm developed in [2]. Using the presented algorithm we can mitigate the exponential blow-up in the number of states, which may happen in the construction of the Nerode automaton, since the number of states of $\mathcal{A}_\varphi$ grows more slowly.

We give a similar algorithm starting from the greatest weak forward simulation on a fuzzy finite automaton $\mathcal{A}$. In that case the resulting automaton is the minimal crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$. Similar constructions can also be given using backward and weak backward simulations on a fuzzy automaton.

References

[1] M. Ćirić, J. Ignjatović, N. Damljanović, M. Bašić, Bisimulations for fuzzy automata, Fuzzy Sets and Systems (2011), doi: 10.1016/ j.fss.2011.07.003.

[2] M. Ćirić, J. Ignjatović, I. Jančić, N. Damljanović, Computation of the greatest simulations and bisimulations between fuzzy automata, submitted to Fuzzy Sets and Systems.

[3] I. Jančić, Z. Jančić, J. Ignjatović, M. Ćirić, Determinization of fuzzy automata using simulations, (in preparation).

[4] I. Jančić, Weak bisimulations for fuzzy automata, (in preparation).

Fuzzy and weighted automata: Canonization methods

Zorana Jančić, Ivana Jančić, Jelena Ignjatović, and Miroslav Ćirić

University of Niš, Faculty of Sciences and Mathematics,
Niš, Serbia
zoranajancic329@gmail.com, ivanajancic84@gmail.com
jelena.ignjatovic@pmf.edu.rs, miroslav.ciric@pmf.edu.rs

Canonization of a weighted (resp. fuzzy) automaton is a determinization procedure whose output is the minimal crisp-deterministic weighted (resp. fuzzy) automaton which is language-equivalent to the original automaton. In this talk we present two canonization methods for fuzzy and weighted automata.

First of them is the Brzozowski's type determinization method for weighted finite automata over a commutative semiring, developed in [4]. For a given a weighted finite automaton over a commutative semiring this method produces a crisp-deterministic weighted automaton which is equivalent to the original weighted automaton. We prove that the obtained automaton is a minimal crisp-deterministic automaton which is language-equivalent to the given weighted automaton. Two types of series have been introduced, right and left series corresponding to states, and it has been shown that the right series of states play an important role in determining whether the given crisp-deterministic weighted automaton is minimal. According to the fact that complete residuated lattices and lattice-ordered monoids are special cases of a commutative semiring, the determinization algorithm presented here is applicable to fuzzy automata over complete residuated lattices and lattice-ordered monoids as well.

The second method is the determinization using closures for fuzzy automata over a complete residuated lattice, developed in [5]. For a given a fuzzy finite automaton over a complete residuated lattice, this method produces a minimal crisp-deterministic fuzzy automaton which is language-equivalent to the original fuzzy automaton. This determinization procedure is a generalization of the determinization algorithm using closures provided in [1, 2] for nondeterministic automata.

The third canonization method, developed in [3] for on fuzzy automata, which is based on the use of weak simulations, will be presented in another talk.

References

[1] R. van Glabbeek, B. Ploeger, Five determinization algorithms, in: O.H. Ibarra and B. Ravikumar (eds.), CIAA 2008, Lecture Notes in Computer Science 5148 (2008) 161–170.

[2] R. van Glabbeek, B. Ploeger, Five determinization algorithms, CS-Report 08-14, Eindhoven University of Technology, 2008.

[3] I. Jančić, Z. Jančić, J. Ignjatović, M. Ćirić, Determinization of fuzzy automata using simulations, (in preparation).

[4] Z. Jančić, Brzozowski's type determinization algorithm for weighted automata over commutative semiring, (in preparation).

[5] Z. Jančić, I. Jančić, M. Ćirić, J. Ignjatović, Determinization of fuzzy automata using closures, (in preparation).

Multi-threaded composition of weighted finite-state transducers

Bryan Jurish[1] and Kay-Michael Würzner[2]

[1]Berlin-Brandenburg Academy of Sciences and Humanities

[2]University of Potsdam

Developments in processing speed of CPUs have slowed down remarkably during recent years. At the same time, the availability of systems with multiple processing units has increased tremendously. In order to accommodate increasing amounts of data and application complexity, traditional algorithms and data structures must be adapted to make use of multiprocessor architectures [7].

Composition is an important operation in constructing and querying complex systems based on weighted finite-state transducers (WFSTs). We present a novel adaptation of a traditional composition algorithm which allows for the application of multiple processors, and empirically evaluate its performance on a test set of random automata.

We consider WFSTs over *commutative* and *complete* semirings [3] (as for example defined in [5], Definition 1). Informally, the composition $\mathfrak{T}_1 \circ \mathfrak{T}_2$ of two WFSTs $\mathfrak{T}_1$ and $\mathfrak{T}_2$ maps inputs of $\mathfrak{T}_1$ to outputs of $\mathfrak{T}_2$ by matching transitions from $\mathfrak{T}_1$ and $\mathfrak{T}_2$ whenever the respective output and input labels coincide.

Implementation With minor modifications, the composition algorithm given in [5] is used in most available implementations of the WFST algebra, e.g. [4, 2]. It serves as a baseline for our parallel specification. The algorithm uses a queue to store as-yet unvisited pairs of states from its operands. These pairs are checked for matching transitions. If the pair of destination states of two matching transitions has not yet been visited, it is itself enqueued for visitation. Such "while queue is not empty" constructs are frequently used in WFST algorithms (i.a. for determinization and synchronization, cf. [5]). Parallelization of such constructs is however difficult to implement, since the number of iterations is not known in advance. In our implementation therefore, the block of computations involved in "visiting" a single pair of states (i.e. lines 7-17 in Table 3.1) is itself delegated to one of a set of concurrently executing program threads. A finite pool of such "worker threads" is globally initialized and maintained, receiving new visitation requests from the queue and delegating these to the workers.

We implemented our adapted algorithm in `C++`, using the `boost` and `TBB` libraries [1, 6] for common data structures and programming idioms.

Experiment The parallelization scheme briefly outlined above does not influence the asymptotic complexity of the composition algorithm. To investigate the merits of using multiple worker threads, we compared running times of the serial algorithm with those of the parallel one.

Materials We generated a pair $(\mathfrak{A}, \mathfrak{B})$ of random WFSTs for each triple (Σ, Q, j) with $\Sigma \in \bigcup_{i=0}^{4}\{2^{2i+1}\}$, $Q \in \bigcup_{i=7}^{11}\{2^i\}$, and $j \in \{0, 1, 2, 4, 8\}$ by generating random tries with $|Q|$ states and $|\Sigma|$ labels and adding $j|Q|$ random arcs to them. A total of 250 WFSTs were thus created.

Method For each pair $(\mathfrak{A}, \mathfrak{B})$ as described above, we performed $\mathfrak{A} \circ \mathfrak{A}$, $\mathfrak{A} \circ \mathfrak{B}$, $\mathfrak{B} \circ \mathfrak{A}$ and $\mathfrak{B} \circ \mathfrak{B}$ with both serial and parallel composition using 1 to 8 worker threads. Running times were recorded and used as the dependent variable of a linear regression analysis.

Results We found that the properties of the transducer returned by the composition algorithm – most notably, the number of transitions (Fig. 3.1) – had a substantial influence not only on the running times of all composition implementations tested, but also on the relative utility of the parallel implementation with respect to its traditional serial counterpart.

Require: Two WFST $\mathfrak{T}_1$ and $\mathfrak{T}_2$
Ensure: The weighted composition $\mathfrak{T}$ of $\mathfrak{T}_1$ and $\mathfrak{T}_2$

$Q \leftarrow \{(q_{0_1}, q_{0_2})\}$
$q_0 \leftarrow (q_{0_1}, q_{0_2})$
$\texttt{enqueue}(S, (q_{0_1}, q_{0_2}))$
while $S \neq \emptyset$ **do**
  $(q_1, q_2) \leftarrow \texttt{head}(S)$
  $\texttt{dequeue(S)}$
  if $(q_1, q_2) \in F_1 \times F_2$ **then**
    $F \leftarrow F \cup \{(q_1, q_2)\}$
    $\rho(q_1, q_2) \rightarrow \rho(q_1) \otimes \rho(q_2)$
  end if
  for all $(e_1, e_2) \in E[q_1] \times E[q_2]$ **s. t.** $o[e_1] = i[e_2]$ **do**
    if $(n[e_1], n[e_2]) \notin Q$ **then**
      $Q \leftarrow Q \cup \{(n[e_1], n[e_2])\}$
      $\texttt{enqueue}(S, (n[e_1], n[e_2]))$
    end if
    $E \leftarrow E \cup \{(q_1, q_2), (n[e_1], n[e_2]), i[e_1], o[e_2], w[e_1] \otimes w[e_2]\}$
  end for
end while
return $\mathfrak{T}$

Table 3.1.: Pseudo-code for the traditional WFST composition algorithm

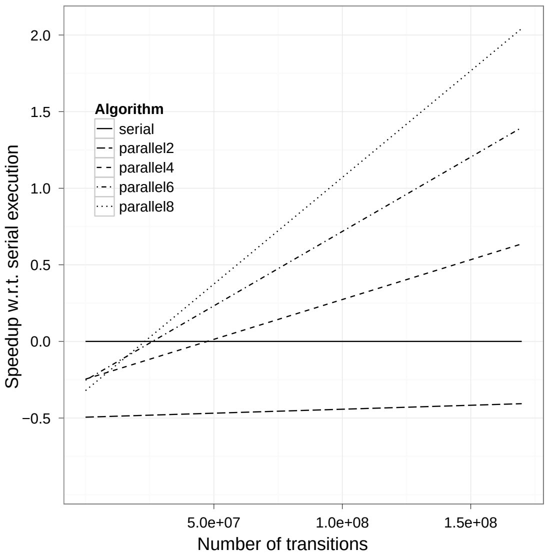


Figure 3.1.: Speedup of composition running times w.r.t serial execution by number of transitions in the result transducer for parallel implementations.

References

[1] Boost C++ libraries, version 1.48.0. http://boost.org, 2011.

[2] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In Jan Holub and Jan Žd́árek, editors, *Implementation and Application of Automata*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer, Berlin/Heidelberg, 2007.

[3] Zoltán Ésik and Werner Kuich. Equational Axioms for a Theory of Automata. In Carlos Martín Vide, Victor Mitrana, and Gheorghe Păun, editors, *Formal Languages and Applications*, volume 148 of *Studies in Fuzziness and Soft Computing*, chapter 10, pages 183–196. Springer, Berlin, Heidelberg, 2004.

[4] Thomas Hanneforth. FSM<2.0> – C++ library for manipulating (weighted) finite automata. http://www.fsmlib.org, 2004.

[5] Mehryar Mohri. Weighted automata algorithms. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science, chapter Concepts of Weighted Recognizability, pages 213–254. Springer, Berlin, Heidelberg, 2009.

[6] James Reinders. *Intel Threading Building Blocks*. O'Reilly, Sebastopol, CA, 2007.

[7] Herb Sutter. The free lunch is over – a fundamental turn toward concurrency in software. *Dr. Dobb's Journal*, 30(3), 2005.

Distance between formulas of the five-valued Lukasiewicz logic and the uncertainty measure of expert statements

Elizaveta Kabanova

Novosibirsk State University, Russia
iameye@mail.ru

In this work statements of experts are represented as formulas of the five-valued Lukasiewicz logic[1]. Likewise the case of the classical binary logic [2], [3], using model theory [4], the following notions was defined as:

1. *Normalized distance ρ between φ and ψ*

$$\rho(\varphi, \psi) = \frac{1}{5^{|S(\Sigma)|}} \sum_{k=0}^{4} \sum_{l=0}^{4} \frac{|k-l|}{4} M(\frac{k}{4}, \frac{l}{4}), \tag{3.1}$$

where $5^{|S(\Sigma)|}$ is a quantity of all models, $S(\Sigma)$ is a set of variables, $M(\frac{k}{4}, \frac{l}{4})$ is a quantity of models in which the formula φ takes value $\frac{k}{4}$, and ψ – $\frac{l}{4}$. Also we consider the case where some values of variables are known:

Let $x_1, ..., x_p$, $x_i \in S(\varphi) \cup S(\psi)$, $i = 1, ..., p$, $p = |S(\varphi) \cup S(\psi)|$ accordingly take $m_1, ..., m_p$, $m_i \leq 5$ values. Then the formula (3.1) is transformed into:

$$\rho(\varphi, \psi) = \frac{1}{m_1 \cdot ... \cdot m_p} \sum_{k=0}^{4} \sum_{l=0}^{4} \frac{|k-l|}{4} M(\frac{k}{4}, \frac{l}{4}). \tag{3.2}$$

It is obvious that the formula (3.1) is a particular case of (3.2) with $m_1 = ... = m_p = 5$.

2. *Normalized uncertainty measure* of statements defined as the distance (3.1) between the logical formula corresponding to a certain statement and identically true formula (an axiom):

$$\mu(\varphi) = \rho(\varphi, 1) = \frac{1}{5^{|S(\Sigma)|}} \sum_{k=0}^{3} \frac{4-k}{4} M(\frac{k}{4}), \tag{3.3}$$

where $M(\frac{k}{4})$ is a quantity of models in which the formula φ takes logical value $\frac{k}{4}$.

In this work properties of introduced notions are defined and proved. It takes into account a semantics of similarity and differences of information contained in statements. These properties are similar to the properties of distance and uncertainty measure designated in binary case [2], [3] and differ from [4].

Introduced distance and measure can be applied for construction of expert systems based on many-valued statements.

In this work examples of grouping a set of statements using the hierarchical clustering algorithm are considered. In this case the uncertainty measure is a stopping criterion of clustering procedure.

References

[1] Karpenko A. S., *Lukasiewicz logics and prime numbers (Russian),* Science, Moscow, 2000.

[2] Lbov G. S., Startseva N. G., *Logic decision functions and questions of statistical stability of solutions (Russian),* Publishing house of institute of mathematics, Novosibirsk, 1999.

[3] Vikentév A. A., Lbov G. S., *Setting the metric and informativeness on statements of experts,* Pattern Recognition and Image Analysis V.7, No.2 (1997), 175 - 183.

[4] Vikentév A. A., *Measure refutability statements of experts, a distance of many-valued logic and processes of adaptation (Russian),* XIV International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008.

Decidability, Undecidability, and PSPACE-Completeness of the Twins Property in the Tropical Semiring

Daniel Kirsten

Humboldt-Universität zu Berlin
Institut für Informatik
Unter den Linden 6
D-10099 Berlin, Germany
`http://www2.informatik.hu-berlin.de/~kirsten`

We solve a problem already investigated by Mohri in 1997 [2]: we show that the twins property for weighted finite automata over the tropical semiring is decidable and PSPACE-complete. We also point out that it is undecidable whether two given states are twins.

The results of the talk are already published [1].

References

[1] D. Kirsten. Decidability, undecidability, and PSPACE-completeness of the twins property in the tropical semiring. *Theoretical Computer Science*, to appear in 2012.

[2] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:269–311, 1997.

The Complexity of Computing the Behaviour of Weighted Büchi Automata over Lattices

Karsten Lehmann and Rafael Peñaloza

Theoretical Computer Science
TU Dresden, Germany
karsten.lehmann@mailbox.tu-dresden.de
penaloza@tcs.inf.tu-dresden.de

Automata have long been recognized as a useful mechanism for solving logic-based reasoning tasks. For example, to decide whether a formula is satisfiable, one can construct an automaton that recognizes all the (well-structured) models of this formula, and decide whether the language accepted by the automaton is empty—in which case the formula is unsatisfiable—or not—and then the formula has a model, i.e. is satisfiable.

When logical reasoning is reduced to the (non-)emptiness of an automaton, as in the previous example, it is usually possible to use a simplified alphabet of cardinality one: the models of the input formula are described by the accepting runs, rather than by the recognized language.

This kind of construction has been generalized to weighted automata over lattices, as a means to deal with non-standard logical semantics. Briefly, every model of the input formula is associated with a weight, and we are interested in finding the supremum of the weights of all these models. Suppose that we can associate every transition of the constructed automaton with a weight, in such a way that the infimum of the weights of all the transitions appearing in a successful run (that is, the weight of the run itself) corresponds exactly to the weight of the model it represents. Then, reasoning in this non-standard semantics reduces to computing the behaviour of a weighted automaton. To understand the complexity of our non-standard reasoning tasks, we first study how hard it is to compute the behaviour of weighted automata over lattices.

For Büchi automata, if the underlying lattice is known to be distributive, then the behaviour is known to be computable in polynomial time [1, 2]. Unfortunately, we cannot always assume distributivity. For arbitrary lattices, we show that the behaviour can be computed by a black-box mechanism that tests emptiness of exponentially many unweighted Büchi automata, which yields an ExpTime upper bound, given an oracle for the lattice operations. If the lattice is finite, then this bound can be improved to PSpace.

We also provide NP and coNP lower bounds by showing that propositional satisfiability and validity can be decided by computing the behaviour of a weighted automaton over a non-distributive lattice. Although we have been so far unsuccessful in closing the gap left by these complexity bounds, we describe some ideas that could be helpful in that direction.

References

[1] Franz Baader and Rafael Peñaloza. Automata-based axiom pinpointing. *Journal of Automated Reasoning*, 45(2):91–129, August 2010. Special Issue: Selected Papers from IJCAR'08.

[2] Manfred Droste, Werner Kuich, and George Rahonis. Multi valued MSO logics over words and trees. *Fundamenta Informaticae*, 84(3-4):305–327, 2008.

On the weighted closure problem

Sylvain Lombardy[1] and Jacques Sakarovitch[2]

[1]LIGM, Université Paris-Est

[2]LTCI, CNRS / Telecom ParisTech, Paris.

The origin of this work is the implementation in the VAUCANSON platform of an ϵ-transition removal algorithm for automata with weights in $\mathbb{Q}$ or $\mathbb{R}$.

The equivalence between non-deterministic finite automata and non-deterministic finite automata with ϵ-transitions is a basic result in the theory of finite automata. The same problem for weighted automata is not less interesting but its solution proves to be more complicated, and may even not exist: if the graph of ε-transitions contains a cycle then the sum of the weights along the paths following this cycle may well be not defined, as for instance in the automaton of Fig. 3.2(a). Most works on automata have ruled out the possibility of having cycles of ε-transitions in automata, either explicitely with the hypothesis of *cycle-free* automata, or implicitely by considering the *discrete topology* on the weight semiring. As a result, the automaton of Fig. 3.2(b) is considered in these works as not having a defined behaviour.

Figure 3.2.: A non-valid automaton (a), and one that should be valid (b)

Yet probabilistic automata, with weights precisely in $\mathbb{Q}$ or $\mathbb{R}$, or distance automata, with weights in $\mathbb{Z}\mathsf{min}$, are natural computational models that it is legitimate to deal with, discrete topology is not the most natural one on $\mathbb{Q}$ or $\mathbb{R}$, and some progress have been made since Zeno.

There are indeed two distinct, although related, questions:

Q1 Is it decidable whether a given $\mathbb{K}$-automaton is valid or not?

Q2 How to compute a *proper* automaton equivalent to a given valid $\mathbb{K}$-automaton?

These questions are related as the decision algorithm for Q1 cannot be so different from the computing one in Q2. But Q1 itself conceals two problems of different nature: one is the description of sufficient conditions on $\mathbb{K}$ such that the answer to Q1 is positive; another one is hidden underneath:

Q3 What is the definition of a *valid* $\mathbb{K}$-automaton?

This work puts forward a consistent answer to these three questions. One should answer first to Q3, then to Q2, and finally to Q1, but it is the solution to Q2 that controls the way Q3 and Q1 are addressed.

The solution to Q2 is a mere generalisation of the algorithm in the Boolean case, but both its *correctness* and *termination* require more care. As for Q3, a $\mathbb{K}$-automaton $\mathcal{A}$ is said to be valid if every rational family of computations yields a result and this is consistent with the operations performed on automata. Finally, a positive answer to Q1 is given if $\mathbb{K}$ is a *topological ordered positive semiring* with the property that the domain of the star operator is closed downward, and this opens the way, via the *absolute value*, to the solution for $\mathbb{Q}$ and $\mathbb{R}$, which fulfils our programme.

Pushing for weighted tree automata *

Andreas Maletti and Daniel Quernheim

Institute for Natural Language Processing, Universität Stuttgart
Azenbergstraße 12, 70174 Stuttgart, Germany
{Andreas.Maletti,Daniel.Quernheim}@ims.uni-stuttgart.de

In this contribution, we consider *pushing* [12, 4] for weighted tree automata [1, 5] over commutative semifields [7, 6]. Roughly speaking, pushing moves transition weights along a path. If the weights are properly selected, then pushing can be used to canonicalize a (bottom-up) deterministic weighted tree automaton [2]. The obtained canonical representation has the benefit that it can be minimized using unweighted minimization [9, 8], in which the weight is treated as a transition label. This strategy has successfully been employed in [12, 4] for deterministic weighted (finite-state) string automata, and we adapt it here for tree automata. In particular, we improve the currently best minimization algorithm [10] for deterministic weighted tree automata from $O(|M| \cdot |Q|)$ to $O(|M| \log |Q|)$, which coincides with the complexity of minimization in the unweighted case [9].

The improvement is achieved by a careful selection of signs of life [10]. Intuitively, a sign of life for a state q is a context which takes q into a final state. In particular, equivalent states will receive the same sign of life, which ensures that their pushing weights are determined using the same evaluation context. This property sets our algorithm apart from the similar algorithm in [10, Algorithm 1] and allows a proper canonicalization. After the pushing weights are determined we perform pushing, which we define for general (potentially nondeterministic) weighted tree automata. We prove that the semantics is preserved and that equivalent states have equally weighted corresponding transitions after pushing, which allows us to reduce minimization to the unweighted case [9].

Secondly, we apply pushing to equivalence testing. The currently fastest algorithm [3] for checking equivalence of two deterministic weighted tree automata M_1 and M_2 runs in time $O(|M_1| \cdot |M_2|)$. Our algorithm that computes signs of life can also handle states in different automata with the help of a particular sum construction. The pushing weight (and the evaluation context) is determined carefully, so that equivalent states in different automata receive the same sign of life. This allows us to minimize both input automata and then only test the corresponding unweighted automata for isomorphism. This approach reduces the run-time complexity to $O(|M| \log |Q|)$, where $|M| = |M_1| + |M_2|$ and $|Q| = |Q_1| + |Q_2|$ is the number of total states.

References

[1] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoret. Comput. Sci.*, 18(2):115–148, 1982.

[2] B. Borchardt. *The Theory of Recognizable Tree Series.* PhD thesis, TU Dresden, 2005.

[3] Frank Drewes, Johanna Högberg, and Andreas Maletti. MAT learners for tree series — an abstract data type and two realizations. *Acta Inform.*, 48(3):165–189, 2011.

*The results presented in this talk have already been published [11]. Both authors were supported by the German Research Foundation (DFG) grant MA/4959/1-1.

[4] Jason Eisner. Simpler and more general minimization for weighted finite-state automata. In *Proc. Joint Meeting of the Human Language Technology Conference and the North American Chapter of the ACL*, pages 64–71. ACL, 2003.

[5] Zoltán Fülöp and Heiko Vogler. Weighted tree automata and tree transducers. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, EATCS Monographs in Theoretical Computer Science, chapter 9, pages 313–403. Springer, 2009.

[6] J. S. Golan. *Semirings and their Applications.* Kluwer Academic Publishers, Dordrecht, 1999.

[7] U. Hebisch and H.J. Weinert. *Semirings – Algebraic Theory and Applications in Computer Science.* World Scientific, Singapore, 1998.

[8] Johanna Högberg, Andreas Maletti, and Jonathan May. Bisimulation minimisation for weighted tree automata. In *Proc. 11th Int. Conf. Developments in Language Theory*, volume 4588 of LNCS, pages 229–241. Springer, 2007.

[9] Johanna Högberg, Andreas Maletti, and Jonathan May. Backward and forward bisimulation minimization of tree automata. *Theoret. Comput. Sci.*, 410(37):3539–3552, 2009.

[10] Andreas Maletti. Minimizing deterministic weighted tree automata. *Inform. Comput.*, 207(11):1284–1299, 2009.

[11] Andreas Maletti and Daniel Quernheim. Pushing for weighted tree automata. In Filip Murlak and Piotr Sankowski, editors, *Proc. 36th Int. Conf. Mathematical Foundations of Computer Science*, volume 6907 of LNCS, pages 460–471. Springer, 2011.

[12] M. Mohri. Finite-state transducers in language and speech processing. *Comput. Linguist.*, 23(2):269–311, 1997.

Parameterized Model Checking of Weighted Networks

Ingmar Meinecke and Karin Quaas

Institut für Informatik, Universität Leipzig,
D-04109 Leipzig, Germany
{meinecke,quaas}@informatik.uni-leipzig.de

We consider networks of many identical processes. Such a network consists of one control process and an arbitrary number of user processes which behave according to the same finite-state process definition. A single process can either execute an internal transition or communicate with another process via handshake synchronization. Each transition may be assigned some value which represents, e.g., emission, energy, money or time that is produced or consumed during the transition. The resulting model, which we call *weighted networks*, can thus be used to model concurrent systems with resources.

We are interested in verifying *qualitative* and *quantitative* properties of weighted networks. For instance, we would like to decide whether the control process can always reach a certain state without breaking a given local budget. Or we may ask for global properties, e.g.: If the controller is in some critical state, do all user processes keep their energy level within a certain interval? Hereby, a certain property should hold *independently* on the number of instantiated user processes, i.e., the number of processes is not fixed.

German and Sistla [6] investigated such parameterized model checking problems for unweighted networks by using Linear Temporal Logic (LTL) as a specification language. To incorporate quantitative aspects, we extend LTL to a quantitative version, called *weighted LTL (wLTL)* by assigning intervals I from the weight structure $\mathcal{D}$ as a constraint to the temporal operators:

$$\varphi ::= \texttt{true} \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc_I \varphi \mid \varphi_1 \mathcal{U}_I \varphi_2$$

where p is an atomic proposition and $I \subseteq \mathcal{D}$ is an interval. In this way, wLTL is syntactically very similar to the well-established real-time logic MTL [7, 2, 9, 10] and its variants [1, 4, 8], or extensions of MTL with several cost functions [3]. However, the interval constraints in our logic wLTL may stem from more general structures than in MTL. We allow intervals from *monotone ordered monoids*, i.e., monoids with a linear order such that the operation is monotone with respect to the order. Also the transitions of the networks are equipped with weights from these structures. Monotone ordered monoids cover the non-negative reals with addition, used in MTL, as well as the non-negative rationals with maximum or the positive integers with multiplication.

The semantics of wLTL formulas are languages of *weighted words*, i.e., every letter of the infinite word is made up of a set of atomic propositions and a weight. Whenever we restrict the weights that may occur in a weighted word to a finite set, we can adapt a construction for classical LTL, cf. [5], and translate *weighted* LTL formulas into *unweighted* Büchi automata recognizing the semantics of the formula over the restricted alphabet. This translation works for every monotone ordered monoid which is *bounded locally finite*, a property that ensures that the monoid, generated by finitely many values up to a certain bound, is finite. The size of the Büchi automaton can be optimized depending on the underlying monoid.

Our translation profits from the elaborated work done for a timed setting. In [8], the authors explore durational Kripke structures and model checking with TCTL and TLTL. There, transitions are labeled with intervals of non-negative integers. The authors show that TLTL model checking is EXPSPACE-complete whereas a fragment of TLTL can be decided in PSPACE. We rewin these results for tight durational Kripke structures, i.e., those with singleton intervals at the transitions, by our translation of wLTL into Büchi automata.

As a main result, we show decidability of the *controller model checking problem* of parameterized networks in a quantitative setting, i.e., it is decidable whether each execution of the control process satisfies a wLTL formula *independently* on the number of user processes. An analogue result holds for user model checking. Using the translation of a weighted formula into an unweighted automaton, we can treat the weights of the processes as mere labels. This way, we can reuse significant concepts and results of [6] for our quantitative model checking procedure.

Weighted LTL formulas express *local* properties of the controller or a user. We also investigate *global* properties specifying both control and user processes of weighted networks at the same time. Model checking networks against formulas from an extension of LTL with universal process quantifiers is undecidable already in the unweighted setting [6]. In our setting, simultaneous satisfaction of a wLTL formula by all control and user processes also turns out to be undecidable. Thus, we restrict to problems concerning the feasability of computations for which the overall weight for every process stays within certain bounds. By reduction on the controller model checking problem, we can show decidability of several of those bounded weight problems. However, if we consider a more global weight model, then even a quite simple global model checking problem turns out undecidable.

References

[1] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

[2] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[3] P. Bouyer, K. G. Larsen, and N. Markey. Model checking one-clock priced timed automata. *Logical Methods in Computer Science*, 4:1–28, 2008.

[4] P. Bouyer, N. Markey, J. Ouaknine, and J. Worrell. On expressiveness and complexity in real-time model checking. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP (2)*, volume 5126 of *LNCS*, pages 124–135. Springer, 2008.

[5] S. Demri and P. Gastin. Specification and verification using temporal logics. In D. D'Souza and P. Shankar, editors, *Modern applications of automata theory*, volume 2 of *IISc Research Monographs*. World Scientific, 2011. To appear.

[6] S. M. German and A. P. Sistla. Reasoning about systems with many processes. *Journal of the ACM*, 39(3):675–735, 1992.

[7] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[8] F. Laroussinie, N. Markey, and Ph. Schnoebelen. On model checking durational Kripke structures. In M. Nielsen and U. Engberg, editors, *FoSSaCS*, volume 2303 of *LNCS*, pages 264–279. Springer, 2002.

[9] J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. In L. Aceto and A. Ingólfsdóttir, editors, *FoSSaCS*, volume 3921 of *LNCS*, pages 217–230. Springer, 2006.

[10] J. Ouaknine and J. Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3(1), 2007.

Büchi-Type Theorems for Unambiguous, Functional and Multi-Weighted Automata

Vitaly Perevoshchikov

Institut für Informatik, Universität Leipzig,
04109 Leipzig, Germany
perev@informatik.uni-leipzig.de

Weighted automata are nondeterministic automata which are equipped with a transition weight function. They can model quantitative aspects of systems like memory or power consumption. The value of a run can be computed, for example, as the maximum, limit average, or discounted sum of transition weights. In multi-weighted automata, transition carry tuples of weights and can model, for example, ratio between rewards and costs, the efficiency of use of some primary resource under some upper bound constraint on some secondary resource. The behavior of these automata cannot be described using semirings or even valuation monoids. The classes of unambiguous and functional automata are special classes of nondeterministic weighted automata. Some important verification problems are decidable for unambiguous or functional weighted automata while they are undecidable for the full class of weighted automata. Here, we establish a connection between multi-weighted automata and multi-weighted logics. For the case of finite and infinite words, we show that suitable weighted MSO logics and these new weighted automata are expressively equivalent. As consequence, we establish Büchi-type theorems for unambiguous and functional weighted automata which hold also for automata over semirings and valuation monoids.

Consistency of Probabilistic Context-Free Grammars

Torsten Stüber

Faculty of Computer Science
Technische Universität Dresden
D-01062 Dresden, Germany
torsten.stueber@tu-dresden.de

Probabilistic context-free grammars (for short: pcfg) [1], which are models widely used in, e.g., natural-language processing [7], are usually required to be *proper*, i.e., the probabilities of all rules having the same left-hand side sum up to one. However, properness does not guarantee that the pcfg is *consistent*, i.e., whether the sum of the probabilities of all derivations of the grammar equals one. It is an important problem to decide whether a given proper pcfg is also consistent.

According to [2], a proper pcfg is consistent if $\rho < 1$ and it is inconsistent if $\rho > 1$, where ρ is the largest eigenvalue of the first-moment matrix of the pcfg. This frequently cited result is unsatisfactory for three reasons: (i) it does not apply if the grammar is not reduced, (ii) it is inconclusive if $\rho = 1$ and (iii) Booth and Thompson provide no algorithm for deciding whether $\rho < 1$ or $\rho > 1$ and, i.e., whether the pcfg is consistent. These issues have been solved recently by [5]. In particular, Etessami and Yannakakis address the special case $\rho = 1$ and demonstrate that problem to decide whether a pcfg is consistent can be solved in polynomial time by reducing this problem to linear programming [8].

In this talk we provide an alternative algorithm that solves this decision problem. Our proof is considerably different from the proofs in the literature [2, 5]: it is neither based on Perron-Frobenius theory [6] nor on linear programming. Our algorithm has time complexity $\mathcal{O}(n^3)$ in the unit-cost model of computation and, therefore, is faster than the linear programming approach in [5].

In many applications the probabilities of a pcfg are generated by an automatic training procedure, particularly by means of maximum-likelihood estimation [4]. It is well-known that for pcfg such a training procedure always yields a consistent grammar [3]. We show that this result follows as a simple corollary from the theory developed in our talk.

References

[1] T.L. Booth. Probabilistic representation of formal languages. In *IEEE Conference Record of 1969 Tenth Annual Symposium on Switching and Automata Theory*, pages 74–81. IEEE, 1969.

[2] T.L. Booth and R.A. Thompson. Applying probability measures to abstract languages. *Computers, IEEE Transactions on*, 100(5):442–450, 1973.

[3] Z. Chi and S. Geman. Estimation of probabilistic context-free grammars. *Computational Linguistics*, 24(2):299–305, 1998.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[5] K. Etessami and M. Yannakakis. Recursive markov chains, stochastic grammars, and monotone systems of nonlinear equations. *Journal of the ACM (JACM)*, 56(1):1–66, 2009.

[6] P. Lancaster and M. Tismenetsky. *The Theory of Matrices: with Applications*. Academic Press, 1985.

[7] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*, volume 59. MIT Press, 1999.

[8] J. Matoušek and B. Gärtner. *Understanding and Using Linear Programming*. Springer-Verlag Berlin, 2007.

Probabilistic Automata and Probabilistic Logic

Thomas Weidner

Institut für Informatik, Universität Leipzig
D-4009 Leipzig, Germany
weidner@informatik.uni-leipzig.de

Probabilistic automata are well-studied and allow a wide range of applications. Recently the concept of probabilistic automata has been transferred to ω-words by Baier and Grösser. We consider the behaviors of probabilistic automata as functions mapping finite or infinite words to a probability value. In spite of the success of Büchi's and Elgot's characterizations of recognizable languages by MSO logic, no logic characterization of the behavior of probabilistic automata has been found yet.

We introduce a *probabilistic MSO logic* (PMSO), which is obtained from classical MSO by the addition of a second-order "expected value" operator $\mathbb{E}_p X$. Within the scope of this operator, formulas $x \in X$ are considered to be true with probability p. In our main result we establish the desired coincidence of the behaviors of probabilistic automata and the semantics of PMSO sentences for finite and infinite words. We also obtain a characterization of probabilistically recognizable functions in terms of classical recognizable languages and Bernoulli measures.

Weighted MSO logic is a different quantitative MSO logic. We can embed PMSO into weighted MSO using an effective syntactic transformation.

Due to the effectiveness of our transformations between probabilistic automata and PMSO sentences, we can prove many decidability and undecidability results for PMSO sentences based on the corresponding results for probabilistic automata.