Weighted Automata — Theory and Applications

Dresden, Germany, May 13–16, 2008

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 13–16, 2008. This workshop covered all aspects of weighted automata, ranging from the theory of formal power series to applications of tree automata, natural language processing, and multi-valued logics. The workshop was attended by 43 participants from 12 countries.

Two tutorials were given by

Z. Ésik (Szeged, Hungary and Tarragona, Spain)
K. Knight (Los Angeles, USA)

In addition, seven survey lectures were presented by

F. Drewes (Umeå, Sweden)
B. Gerla (Varese, Italy)
A. Maletti (Berkeley, USA)
G. Rahonis (Thessaloniki, Greece)
S. Gaubert (Rocquencourt, France)
W. Kuich (Vienna, Austria)
W. Martens (Dortmund, Germany)

Furthermore, 17 talks were selected as technical contributions.

The workshop was organized jointly by the Chair for Automata and Formal Languages of *Leipzig University* and the Chair for Foundations of Programming of *Technische Universität Dresden*. For further financial support we would like to thank the activity "Gesellschaft von Freunden und Förderern der TU Dresden" and the "International Center for Computational Logic".

Call for Papers

The journal *Acta Cybernetica* has agreed to publish a special issue on this topic. Submissions related to this topic could be either survey articles or research papers and will be refereed as usual. Participation in the above workshop is encouraged, but is not a prerequisite for a submission.

Authors are asked to submit their contribution preferably in PostScript or PDF to both of the editors of the special issue. Please send your files to

`droste@informatik.uni-leipzig.de` and `vogler@inf.tu-dresden.de` .

Deadline for submissions is **July 11, 2008**. We intend to ensure a quick refereeing process. Authors of published papers will be provided with 50 reprints free of charge.

Manfred Droste and Heiko Vogler

Part I

Scientific Programme

Tuesday, May 13, 2008

08:30–09:00		Registration
09:00–10:30	K. Knight *An overview of weighted automata in natural language processing (I)*	Tutorial
10:30–11:00		Break
11:00–12:00	W. Kuich *Why we need semirings in automata theory*	Survey Lecture
12:00–13:30		Lunch
13:30–14:30	F. Drewes *Learning: from string languages to tree series*	Survey Lecture
14:35–15:00	F. Denis, A. Habrard, R. Gilleron, M. Tommasi, É. Gilbert *On probability distributions for trees: representations, inference and learning*	Technical Contribution
15:00–15:30		Break
15:30–15:55	A. Eckl *Prediction of subalphabets and ranking in DAWG's for natural languages*	Technical Contribution
15:55–16:20	T. Hanneforth, K.-M. Würzner *Statistical language models within the algebra of weighted rational languages*	Technical Contribution
16:20–16:45		Break
16:45–17:10	K. Quaas, M. Droste *A Kleene-Schützenberger theorem for weighted timed automata*	Technical Contribution
17:10–17:35	D. Kirsten, S. Lombardy *Deciding unambiguity and sequentiality from a polynomially ambiguous min-plus automaton*	Technical Contribution

Wednesday, May 14, 2008

08:30–09:00		REGISTRATION
09:00–10:30	K. Knight *An overview of weighted automata in natural language processing (II)*	TUTORIAL
10:30–11:00		BREAK
11:00–12:00	W. Martens *XML research for formal language theorists*	SURVEY LECTURE
12:00–13:30		LUNCH
13:30–14:30	A. Maletti *Minimization of weighted automata*	SURVEY LECTURE
14:35–15:00	E. Mandrali, G. Rahonis *Weighted tree automata with discounting*	TECHNICAL CONTRIBUTION
15:00–15:30		BREAK
15:30–15:55	Z. Fülöp, M. Steinby *Varieties of recognizable tree series over fields*	TECHNICAL CONTRIBUTION
15:55–16:20	Z. Fülöp, L. Muzamel *Weighted tree-walking automata*	TECHNICAL CONTRIBUTION
16:20–16:45		BREAK
16:45–17:10	C. Mathissen *Weighted logics for nested words and algebraic formal power series*	TECHNICAL CONTRIBUTION
17:10–17:35	T. Stüber, H. Vogler, Z. Fülöp *Decomposition of weighted multioperator tree automata*	TECHNICAL CONTRIBUTION

Thursday, May 15, 2008

08:30–09:00		REGISTRATION
09:00–10:30	Z. Ésik *Iteration theories as an axiomatic foundation of automata and language theory (I)*	TUTORIAL
10:30–11:00		BREAK
11:00–12:00	B. Gerla *Many-valued logic and fuzzy automata*	SURVEY LECTURE
12:00–13:30		LUNCH
13:30–14:30	G. Rahonis *Multi-valued automata: theory and applications*	SURVEY LECTURE
14:35–15:00	I. Meinecke *On the expressive power of a weighted μ-calculus*	TECHNICAL CONTRIBUTION
15:00–15:30		BREAK
15:30–15:55	M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković *State reduction of fuzzy automata*	TECHNICAL CONTRIBUTION
15:55–16:20	J. Ignjatović, M. Ćirić, T. Petković *Relationships between FFA-recognizability and DFA-recognizability of fuzzy languages*	TECHNICAL CONTRIBUTION
16:20–16:45		BREAK
16:45–18:00		WORKSHOP
19:00–open		CONFERENCE DINNER

Friday, May 16, 2008

08:30–09:00		REGISTRATION
09:00–10:30	Z. Ésik *Iteration theories as an axiomatic foundation of automata and language theory (II)*	TUTORIAL
10:30–11:00		BREAK
11:00–12:00	S. Gaubert *To be announced*	SURVEY LECTURE
12:00–13:30		LUNCH
13:30–13:55	V. Halava, T. Harju, E. Lehtonen *A survey of integer weighted finite automata*	TECHNICAL CONTRIBUTION
13:55–14:20	D. Kuske *From unweighted to weighted traces — alternative proofs*	TECHNICAL CONTRIBUTION
14:20–14:45	S. Schwarz, R. Winter *Recognizability of iterative picture languages*	TECHNICAL CONTRIBUTION
14:45–15:15		BREAK
15:15–15:40	A. Koprowski, J. Waldmann *Max/Plus tree automata for termination of term rewriting*	TECHNICAL CONTRIBUTION
15:40–16:05	A. Gebhardt, J. Waldmann *Weighted automata define a hierarchy of terminating string rewriting systems*	TECHNICAL CONTRIBUTION
16:05–16:15		BREAK
16:15–open		JOINT RESEARCH

Part II

Abstracts

Tutorials

Iteration Theories as an Axiomatic Foundation of Automata and Language Theory

Zoltán Ésik

Dept. of Computer Science, University of Szeged

GRLMC, Rovira i Virgily University

Fixed points and fixed point computations occur in just about every field of Computer Science. They are often used to give semantics to recursion, in automata and language theory, programming languages and abstract data types, concurrency and logic, to mention only a few applications. For one familiar example, one can canonically associate with each context free grammar a vector valued function over the domain of all subsets of the free monoid over the set of terminals, so that the language generated by the grammar becomes a component of the least fixed point of the function.

Typical questions about fixed points are: when do fixed points exist, and what are their properties. There are several fixed point theorems that have found applications in Computer Science, each guaranteeing the existence of certain *canonical* fixed points under certain conditions. Examples of such fixed point theorems are Tarski's fixed point theorem and several of its variants, involving complete lattices or cpo's and monotone or continuous functions, categorical generalizations of these theorems, Banach's fixed point theorem involving proper contractions over complete metric spaces, etc. Regarding the properties of the fixed point operations, it has been shown that all fixed point operations share the same equational laws. Letting these equational laws the axioms, together with some axioms special to a discipline such as languages, concurrency, we obtain an axiomatic basis for that discipline. It is then interesting to know how far one can get with the axiomatic approach.

The use of equations has several advantages. Proofs can be separated into two parts, where the first part establishes the equational axioms, and the second is based on simple equational reasoning. Such proofs have a transparent structure and are usually very easy to understand, since manipulating equations is one of the most common way of mathematical reasoning. Moreover, since many results depend on the same equations, the first part of such proofs usually provides a basis to several results. Finally, the results obtained by equational reasoning have a much broader scope, since many models share the same equations.

The aim of this tutorial is to provide an introduction to that part of the theory of fixed points that has applications to weighted automata. We start with a treatment of fixed points in the ordered setting and review some basic theorems guaranteeing the existence of least (or greatest) fixed points. Then we establish several (equational) properties of the least fixed point operation including the Bekić identity, asserting that systems of fixed point equations can be solved by the technique of successive elimination. Then we use the Bekić identity and some other basic laws to introduce the axiomatic frameworks of Conway and iteration theories. We provide several axiomatizations of these notions and show that iteration theories capture the equational properties of the fixed point operation in a large class of models.

We also treat fixed points of linear functions over semirings and semimodules. The main results show that for such functions, the fixed point operation can be characterized by a star operation, possibly in conjunction with an omega operation. We show that the equational properties of the fixed point operation are reflected by corresponding properties of the star and omega operations.

As a main application of the theory of fixed points, we will show that Kleene's theorem, be it formulated for classical automata, weighted automata, or weighted tree automata, or Buchi automata, rests on the same axiomatics. As a second main application, we will cover the axiomatization of the algebra of regular languages and rational power series.

An Overview of Weighted Automata in Natural Language Processing

Kevin Knight
Information Sciences Institute and Computer Science Department
University of Southern California
knight@isi.edu

Natural Language Processing tackles a number of practical problems, e.g.:

- automated language translation (e.g., Chinese to English)
- speech recognition
- information retrieval
- question answering
- grammar checking
- speech synthesis
- automatic summarization

etc.

These problems are not solved with concise algorithms alone—rather, solutions must be powered by tremendous amounts of formalized knowledge about words, pronunciations, syntax, semantics, and the world.

Weighted automata form an elegant and satisfying way to represent such knowledge. Furthermore, learning algorithms associated with weighted automata permit us to obtain large amounts of linguistic knowledge automatically from online text and speech corpora.

This tutorial will cover the use of weighted automata across many problems in natural language processing. We will also illustrate, in depth, major issues in automated language translation, a challenging problem that requires both analysis of source-language sentences and generation of new, grammatical target-language sentences that have never been uttered before.

We will also track historical developments in both automata theory and natural language processing. These two fields were tightly knit in the middle of the 20th century, but over time they drifted apart, with neither theory nor practice significantly informing one another. Finite-state methods returned to make a dramatic impact on natural language in the 1990s, when they were coupled with automatic knowledge acquisition methods. In this century, tree automata have received renewed interest, being able to capture linguistic transformations (such as observed in natural language translation data) that pose difficulties for string-based automata.

Finally, we will examine a wide variety of automata models from the point of view of what is needed in contemporary practical natural language systems. We find very good synergy—many automata theorems find wonderful application in language systems (greatly simplifying their design), while demands of practical systems raise challenging questions for the theory side.

Survey Lectures

Learning: from String Languages to Tree Series

Frank Drewes

Department of Computing Science, Umeå University
S-901 87 Sweden
drewes@cs.umu.se

Abstract. The talk gives an overview on algorithmic learning, focusing on the field of grammatical inference. After a very brief overview on algorithmic learning in general, some of the major models and approaches used in grammatical inference of string languages are explained. Finally, grammatical inference of tree languages and tree series is discussed.

1 Algorithmic Learning

Unsurprisingly, algorithmic learning is about algorithms that "learn". However, what does this mean? Usually, research in this field focuses on algorithms that (a) adjust, and thus improve, their behaviour over time or (b) use limited information (such as training samples) to, eventually, correctly recognize a general concept or compute a function. In fact, this is merely a matter of perspective. For example, the bottom line of learning to avoid mistakes is to learn the concept *mistake*. Conversely, an algorithm that learns a function f from argument-value pairs can be seen as an algorithm that improves its behaviour when being asked for the value of $f(x)$. Furthermore, it is clear that concept learning (e.g., learning the concept 'prime number' or 'picture of an apple') is a special case of function learning, namely of learning the characteristic function of the concept.

Several areas in Computer Science study aspects of algorithmic learning:

(1) *Machine Learning* studies learning from the point of view and using the methods of Artificial Intelligence.

(2) *Pattern Recognition* is algorithmic learning whenever it is concerned with discovering general patterns in input data.

(3) *Inductive Inference*, an area founded by Solomonoff in the years around 1960 [14], focuses on learning a concept or function, usually from observations (i.e., examples). Often, statistical methods are used, and correctness means correctness with a high degree of probability. See [3] for a survey.

(4) *Grammatical Inference* addresses the problem of learning formal languages, with the additional requirement that the algorithm shall produce an explicit grammatical or automata-theoretic representation of the target language. See [7, 10] to obtain an initial overview of the field.

Obviously, these areas are not disjoint from each other. To some extent, one may see the list above as a series of specializations, i.e., (1) $\supseteq$ (2) $\supseteq$ (3) $\supseteq$ (4).

Research in the field of algorithmic learning may or may not belong to theoretical computer science. The part that does, defines the area of Computational Learning Theory (COLT). As Angluin puts it in her survey [2], the goal of COLT is to "give a rigourous, computationally detailed and plausible account of how learning can be done." See also the book by Kearns and Vazirani [11].

2 Grammatical Inference of String Languages

As mentioned, the purpose of a grammatical inference algorithm (called a learner in the following) is to learn a target language $L \subseteq \Sigma^*$ by constructing an appropriate grammar or automaton. Whether this is possible depends not only on the class of languages considered, but also on the learning model: which kind of information is available to the learner, how does it get this information, and what the criteria of success? Some of the best-known settings are the following.

Learning from Examples and Identification in the Limit Gold [8] defines two of the most natural settings for grammatical inference: *learning from text*, where the learner is given an exhaustive sequence of positive examples $u_1, u_2, \ldots$, i.e., $L = \{u_1, u_2, \ldots\}$, and *learning from an informant*, where the sequence contains both positive and negative examples $(u_1, b_1), (u_2, b_2), \ldots$, i.e., $\{(u_i, b_i) \mid i \in \mathbb{N}\} = L \times \{1\} \cup \overline{L} \times \{0\}$. Furthermore, Gold proposes a criterion of success: After each piece of information received, the learner answers with a new hypothesis h_i, being an automaton or a grammar consistent with the information seen so far. L is *identified in the limit* if, for some $i \in \mathbb{N}$, $h_i = h_{i+1} = \cdots$ and $L(h_i) = L$.

Probably Approximately Correct Learning In Valiant's PAC learning [15], success is defined in a statistical manner. The learner is given additional parameters δ, ϵ $(0 < \delta, \epsilon < 1)$, and is then provided with examples drawn according to an unknown probability distribution D. Eventually, it returns the automaton A learned. The *error probability* of A is $\mathrm{err}(A) = \mathrm{Prob}[u \in L \triangle L(A) \mid u \in \Sigma^*$ drawn according to $D]$ (where $\triangle$ denotes symmetric difference). Now, the correctness requirement is that $\mathrm{err}(A) \leq \epsilon$ (approximate correctness) with probability at least δ (i.e, probably).

Query learning Angluin [1] invented query (or active) learning, where the learner can ask an oracle, the *teacher*, certain types of queries. The most popular teacher of this sort is the so-called minimal adequate teacher (MAT). Given that $\mathcal{A}$ is the class of automata of interest, the learner can ask *membership queries*: "Does w belong to the target language?" and *equivalence queries*: "Does $A \in \mathcal{A}$ represent the target language L? If not, give me a counterexample $w \in L(A) \triangle L$." Angluin's L_* learner learns any regular language in polynomial time from a MAT, using a modified version of Gold's observation table.

3 Inference of Tree Languages

Results on regular languages can usually be generalised to regular tree languages, and this is true even for inference algorithms. For instance, the notions of k-reversibility, k-testability, and function distinguishability mentioned above can be generalised to regular tree languages, and imply efficient learnability from text. The same holds for Angluin's L_* learner [13, 5]. Results like these are particularly interesting in view of the negative results regarding the learnability of context-free languages, as they show that context-free languages can be learned if structural information about the strings in the language is available.

Let us briefly describe the idea behind the L_* learner for the tree case. As usual, a *context* is a tree c with a unique occurrence of a variable x, and $c \cdot t$ denotes the tree obtained by substituting x in c with a tree t. For a (regular) target language L, define the Myhill-Nerode congruence $\equiv_L$ on T_Σ by $s \equiv_L t$ iff $c \cdot s \in L \iff c \cdot t \in L$ for all contexts c. It is well known that the index of $\equiv_L$ (i.e., the number of equivalence classes) is finite iff L is regular, and that the corresponding unique minimal deterministic bottom-up tree automaton A_L is obtained by using the congruence classes as states.

Now, let T be the infinite table given as follows. The rows (columns) are indexed by trees t (contexts c, resp.), and entry $T(t, c)$ is 1 if $c \cdot t \in L$ and 0 otherwise. By definition, $s \equiv_L t$ iff the rows of s and t are equal. Thus, a finite subtable of T suffices to define A_L, because there are only finitely many pairwise distinct rows and columns. Starting with the empty table T_0, the L_* learner builds such a finite subtable T_n of T. It repeatedly constructs the automaton A_i given by the current table T_i to ask an equivalence query. If $L(A_i) \neq L$, the counterexample received can be used to extend the table by new rows and columns, yielding A_{i+1}. Membership queries are mainly needed to fill in new cells of the table. At most $\mathrm{index}(L)$ loop execution are needed to discover A_L.

4 Inference of Tree Series

A natural next step is to extend grammatical inference algorithms to the case of tree series $\psi\colon T_\Sigma \to S$, for some semiring S. The number of papers addressing this problem is still rather small. One may roughly divide them into two categories. The first deals with the special case of stochastic tree languages, i.e., where S is the field $\mathbb{R}$, $(\psi, t) \in [0,1]$ for all $t \in T_\Sigma$, and $\sum_{t\in T_\Sigma}(\psi, t) = 1$. Stochastic languages have received particular interest in natural language processing. When dealing with the learnability of stochastic tree languages, it is probably most natural to consider a learning-from-text-like setting: positive examples are drawn according to a probability distribution D, and the goal is to learn D in the limit by, e.g., constructing a weighted tree automaton (wta). For the case where ψ is recognisable, Denis and Habrard have recently presented such a learner [4].

The second category of learners works on tree series that are not restricted to stochastic ones. There seem to be only two results of this kind, both using Angluin's MAT model and her general algorithmic idea based on an observation table. For this, membership queries are generalised to *coefficient queries*: given a tree $t \in T_\Sigma$, the teacher replies with (ψ, t). Of course, equivalence queries have to be extended to the type of wta considered. The entries of the observation table are now the values $(\psi, c{\cdot}t)$. One of the learners, proposed by Drewes and Vogler and improved by Maletti [6, 12], learns a deterministically recognisable tree series ψ over a commutative semifield by constructing the corresponding minimal deterministic wta. The second learner, proposed by Habrard and Oncina [9] learns a recognisable tree series ψ over a field, by constructing the corresponding minimal nondeterministic wta. Note that a field is assumed, which explains why nondeterministic devices can be learned by Angluin's method, whereas nothing similar has yet been achieved for the Boolean case, where no appropriate finite algebraic characterization is known.

References

1. Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, 1987.
2. Dana Angluin. Computational learning theory: survey and selected bibliography. In *Proc. 24th Annual ACM Symposium on Theory of Computing (STOC 1992)*, pages 351–369. ACM Press, 1992.
3. Dana Angluin and Carl H. Smith. Inductive inference: Theory and methods. *ACM Computing Surveys*, 15:237–269, 1983.
4. François Denis and Amaury Habrard. Learning rational stochastic tree languages. In M. Hutter, R.A. Servedio, and E. Takimoto, editors, *Proc. 18th International Conference on Algorithmic Learning Theory (ALT 2007)*, volume 4754 of Lecture Notes in Computer Science, pages 242–256, 2007.
5. Frank Drewes and Johanna Högberg. Query learning of regular tree languages: How to avoid dead states. *Theory of Computing Systems*, 40:163–185, 2007.
6. Frank Drewes and Heiko Vogler. Learning deterministically recognizable tree series. *Journal of Automata, Languages and Combinatorics*, 2008. To appear.
7. Henning Fernau and Colin de la Higuera. Grammar induction: An invitation for formal language theorists. *Formal Grammars*, 7:45–55, 2004.
8. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
9. Amaury Habrard and José Oncina. Learning multiplicity tree automata. In Y. Sakakibara, S. Kobayashi, K. Sato, T. Nishino, and E. Tomita, editors, *Proc. 8th International Colloquium on Grammatical Inference: Algorithms and Applications (ICGI 2006)*, volume 4201 of Lecture Notes in Computer Science, pages 268–280, 2006.
10. Colin de la Higuera. A bibliographical study of grammatical inference. *Pattern Recognition*, 38:1332–1348, 2005.
11. Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.
12. Andreas Maletti. Learning deterministically recognizable tree series – revisited. In S. Bozapalidis and G. Rahonis, editors, *Proc. 2nd International Conference on Algebraic Informatics (CAI 2007)*, volume 4728 of Lecture Notes in Computer Science, pages 218–235, 2007.
13. Yasubumi Sakakibara. Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, 76:223–242, 1990.
14. Ray Solomonoff. A formal theory of inductive inference, parts I and II. *Information and Control*, 7:1–22 and 224–254, 1964.
15. Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

Many-valued logic and fuzzy automata

Brunella Gerla

Dept. Informatics and Communications, University of Insubria
21100 Varese, Italy
brunella.gerla@uninsubria.it

In the last decades, the interest in fuzzy sets and fuzzy logic has grown from different points of view. From one side, many engineering applications have been proposed based on the use of fuzzy sets as a tool to solve non-linear phenomena through a linguistic representation. From another side, fuzzy sets have motivated a renewed interest in truth-functional logics with an enlarged set of truth values. Indeed a deep study of such logics has being portrayed in the last years and many-valued logic has been proposed to model phenomena in which uncertainty and vagueness are involved.

Very general classes of many-valued propositional logics are the Basic logic defined in [8] as the logic of continuous t-norms and the MTL logic defined in [7] as the logic of left-continuous t-norm. We shall give a few details on such structures. Special cases of propositional many-valued logics are Łukasiewicz, Gödel and Product logic. In particular Łukasiewicz logic has been deeply investigated, together with its algebraic counterpart, *MV-algebras*, introduced by Chang in [1] to prove completeness theorem of Łukasiewicz logic. MV-algebras can be thought of as a special generalization of Boolean algebras in which the idempotency of conjunction and the excluded middle low are not valid.

MV-algebras have nice algebraic properties and can be considered as intervals of lattice-ordered groups. Łukasiewicz disjunction and conjunction are interpreted by the operations $\oplus$ and $\odot$ of the MV-algebra $[0, 1]$ given by

$$x \oplus y = \min\{1, x + y\}, \quad x \odot y = \max\{0, x + y - 1\}.$$

In spite of satisfying theoretical results regarding Łukasiewicz logic, all the attempts to use it as an instrument to deal with uncertainty phenomena, for example in the fuzzy context, had to deal with one of its main characteristic: conjunction and disjunction do not distribute one with respect to the other. This makes difficult to use it as a generalization of Boolean logic.

We stress that operations $\odot$ and $\oplus$ in any MV-algebra A both are related to the same operation in the lattice ordered group associated with A. In order to model the notion of conjunction and disjunction one have instead to consider a lattice operation $\wedge$ (or dually, $\vee$) together with the MV-algebraic operation $\oplus$ (or dually $\odot$). MV-algebras have many semiring reducts, as for example those given by considering operations $\odot, \vee$ or operations $\oplus, \wedge$ or even $\wedge, \vee$.

MV-algebras operations have symmetric properties, since the negation in an MV-algebra A is actually an isomorphism of the monoid $(A, \odot, 1)$ onto $(A, \oplus, 0)$ (indeed MV-algebras are De Morgan algebras). In general, other structures related with other many-valued logics do not have such a symmetry. Nevertheless, each of these structures has a semiring reduct.

In [2] we suggested to consider the pairs of connectives forming the semiring reducts of MV-algebras in order to handle the mathematics behind many fuzzy systems. This approach has been then extended to a generalization of linear algebra to a fuzzy context in [3].

In order to show in which way this representation can be useful to model fuzzy phenomena we give examples in the field of automata following the approach of [6], where semirings have been proposed to give a generalization of automata, the so called K-Σ- automata.

K-Σ- automata can be consider as fuzzy automata, in the sense that are non-deterministic automata in which every transition from one state to another happens with some degree. Then each word is recognized with a degree that must be computed by the degrees of the single transitions, hence each fuzzy automaton accepts a fuzzy language, that is a fuzzy subset of the set of all finite words over a given alphabet.

More recently, automata with values in semirings over the natural numbers or the real numbers sets have been deeply investigated both to finding results on nondeterminism or infinite behavior of finite automata, in the context of formal power series and of weighted automata (see [4],[5],[9], [10]). We shall give a description of automata having values in semirings associated with BL-algebras and MV-algebras.

References

1. C.C. Chang. Algebraic analysis of many valued logics. *Trans. Amer. Math. Soc.*, 88:467–490, 1958.
2. A. Di Nola and B. Gerla. Algebras of Łukasiewicz's logic and their semiring reducts. *Journal of Algebra and its Applications*, 5:417–439, 2006.
3. A. Di Nola, A. Lettieri, I. Perfilieva and V. Novák. Algebraic analysis of fuzzy systems, *Fuzzy Sets and Systems*, 158:1-22, 2007.
4. M. Droste, P. Gastin. Wighted Automata and Wieghted Logics. *Theoretical Computer Science*, 380: 513–525, 2007.
5. M. Droste, W. Kruich, G. Rahonis. Multi-valued MSO logics over words and trees *Fundamenta Informaticae*, 2008, to appear.
6. S. Eilenberg. *Automata, Languages, and Machines.* Academic Press, 1974.
7. Esteva, F., Godo, L.: Monoidal t-norm based logic: towards a logic for left-continuous t-norms. Fuzzy Sets ans Systems. 124:271–288, 2001.
8. P. Hájek, *Metamathematics of fuzzy logic.* Dordrecht: Kluwer Academic Publishers, 1998.
9. D. Krob. Some automata-theoretic aspects of min-max-plus semirings. In J. Gunawardena, editor. Idempotency Analysis. Cambridge University Press, 1998.
10. I. Simon. Recognizable sets with multiplicities in the tropical semiring. In M.P.Chytil et al., editor, *Lect. Notes in Comput. Sci.*, number 324, pages 107–120, 1988.

Why we need semirings in automata theory

Werner Kuich

Technische Universität Wien

kuich@tuwien.ac.at

The use of semirings, formal power series, matrices and fixed point theory in formal language and automata theory yields the following advantages:

(i) The constructions needed in the proofs are mainly the usual ones.

(ii) The descriptions of the constructions by formal series and matrices do not need as much indexing as the usual descriptions.

(iii) The proofs are separated from the constructions and do not need the intuitive contents of the constructions. Often they are shorter than the usual proofs.

(iv) The results are more general than the usual ones. Depending on the semiring used, the results are valid for classical grammars and automata, classical grammars and automata with ambiguity considerations, probabilistic grammars or automata, etc.

(v) The use of formal power series and matrices gives insight into the mathematical structure of problems and yields new results and solutions to unsolved problems that are difficult, if not impossible, to obtain by other means.

In our lecture we give examples that illustrate these advantages.

Minimization of Weighted Automata

Andreas Maletti*

International Computer Science Institute
Berkeley, CA 94704, USA
maletti@icsi.berkeley.edu

Weighted automata are used in a variety of applications (e.g., natural language processing, probabilistic model checking, etc). We will review minimization algorithms for weighted automata with a strong emphasis on tree automata. In addition, partial but efficient minimization procedures such as bisimulation minimization are considered. Special attention will be given to the runtime complexity of the algorithms and whenever available we will substantiate the results with practical experience gained from implementations.

* Author on leave from *Technische Universität Dresden, Faculty of Computer Science, 01062 Dresden, Germany*, with the help of financial support by a DAAD (German Academic Exchange Service) grant.

XML Research for Formal Language Theorists

Wim Martens

Technical University of Dortmund

wim.martens@udo.edu

Formal Language Theory plays a dominant role in XML research. The design of the predominant XML schema languages is based on context-free grammars and tree automata, and widely used navigation and transformation languages such as XPath and XSLT are closely tied to regular expressions and tree transducers. The investigation of these schema and query languages therefore significantly benefits from the large corpus of results in Formal Language Theory.

Conversely, XML research is also a motivation and a source of inspiration for Formal Language Theory. Static analysis questions in XML research, for instance, motivate the deeper study of problems such as membership testing, containment, equivalence, and minimization for various forms of regular expressions and finite automata.

I will give an overview of this synergy between XML research and Formal Language Theory.

Multi-valued automata: theory and applications

George Rahonis
Department of Mathematics, Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
grahonis@math.auth.gr

We present *multi-valued automata over bounded distributive lattices* acting on finite words. They constitute a special subclass of weighted automata over arbitrary semirings, and they have nice properties due to the lattice operations. For instance for every multi-valued automaton we can effectively construct an equivalent deterministic one, which moreover can be minimized. Furthermore, the equivalence problem is decidable for the behaviors of multi-valued automata. If the underlying lattice has a negation function, then we can show the expressive equivalence of multi-valued automata with *multi-valued monadic second order* sentences (cf. [1] for a more general treatment). Usual fuzzy automata over the interval $[0,1]$ is a great paradigm of multi-valued automata.

We deal also with *multi-valued Büchi* and *Muller automata* investigated in [2]. By extending a well-known result for classical automata on infinite words, we prove that the families of the behaviors of the two models coincide. We show the expressive equivalence of our automata with multi-valued monadic second order sentences provided that the underlying lattice has a negation mapping. Then, we compare our models with the *Büchi lattice automata* of Kupferman and Lustig [3]. A subclass of Büchi lattice automata, which in fact coincides with our multi-valued Büchi automata over De Morgan algebras, is related to *lattice linear temporal logic* which in turn is related to important *multi-valued model checking* applications. We investigate the relation among the multi-valued monadic second order logic and the lattice linear temporal logic. On the other hand, we highlight future research lines motivated by the following fact. A critical point for the (multi-valued) automata-theoretic approach of (multi-valued) model checking, is the complexity bound for the constructions on (multi-valued) automata. For instance, we are interested in the complexity bound for complementing (multi-valued) automata over infinite words. It turns out that several constructions on our multi-valued Muller automata [2] have much lower complexity bounds than the corresponding ones for Büchi lattice automata [3]. Therefore, it should be interesting to investigate the contribution of multi-valued Muller automata to the automata-theoretic approach of multi-valued model checking.

References

1. M. Droste, P. Gastin, Weighted automata and weighted logics, *Theoret. Comput. Sci.* 380(2007) 69-86; extended abstract in: *Proceedings of ICALP 2005*, *LNCS* 3580(2005) 513-525.
2. M. Droste, W. Kuich, G. Rahonis, Multi-valued MSO logics over words and trees, *Fund. Inform.*, in press.
3. O. Kupferman, Y. Lustig, Lattice automata, in: *Proceedings of VMCAI 2007, LNCS* 4349(2007) 199-213.

Technical Contributions

State Reduction of Fuzzy Automata

Miroslav Ćirić[1], Aleksandar Stamenković[1], Jelena Ignjatović[1], Tatjana Petković[2]

[1]Faculty of Sciences and Mathematics, University of Niš
Višegradska 33, 18000 Niš, Serbia
jejaign@yahoo.com, ciricm@bankerinter.net

[2]Nokia
Joensuunkatu 7, FIN-24100 Salo, Finland
tatjana.petkovic@nokia.com

In this talk we will present the results from [1, 2] concerning state reduction of fuzzy automata. It has been shown in [1, 2] that the size reduction problem for fuzzy automata is related to the problem of solving a particular system of fuzzy relation equations. This system consists of infinitely many equations, and finding its general solution is a very difficult task. From that reason we consider certain special cases. One of them is a finite system whose solutions, called right invariant fuzzy equivalences, are common generalizations of right invariant or well-behaved equivalences used in reduction of non-deterministic automata, and congruences on fuzzy automata studied in [8]. A procedure for constructing the greatest right invariant fuzzy equivalence contained in a given fuzzy equivalence has been given in [1], and it has been shown that the method for reduction of fuzzy automata based on right invariant fuzzy equivalences gives better results than all other methods developed in [3–8].

In [2], an analogue of a right invariant fuzzy equivalence, called a left invariant fuzzy equivalence, has been considered. It has been shown that the combination of reduction methods based on right invariant and left invariant fuzzy equivalences can give better results than using only one of these methods. It has been also proved that using quasi-orders can give even better results than using fuzzy equivalences.

Acknowledgment. Research supported by Ministry of Science, Republic of Serbia, Grant No. 144011

References

1. Ćirić, M., Stamenković, A., Ignjatović, J., Petković, T.: Factorization of fuzzy automata. In: Csuhaj-Varju, E., Ésik, Z. (eds.), FCT 2007. Lecture Notes in Computer Science, vol. 4639, pp. 213–225. Springer, Heidelberg (2007)
2. Ćirić, M., Stamenković, A., Ignjatović, J., Petković, T.: Fuzzy relation equations and reduction of fuzzy automata. submitted for publication
3. Basak, N.C., Gupta, A.: On quotient machines of a fuzzy automaton and the minimal machine. Fuzzy Sets and Systems 125, 223–229 (2002)
4. Cheng, W., Mo, Z.: Minimization algorithm of fuzzy finite automata. Fuzzy Sets and Systems 141, 439–448 (2004)
5. Lei, H., Li, Y.M.: Minimization of states in automata theory based on finite lattice-ordered monoids. Information Sciences 177, 1413–1421 (2007)
6. Malik, D.S., Mordeson, J. N., Sen, M.K.: Minimization of fuzzy finite automata. Information Sciences 113, 323–330 (1999)
7. Mordeson, J.N., Malik, D.S.: Fuzzy Automata and Languages: Theory and Applications. Chapman & Hall/CRC, Boca Raton, London (2002)
8. Petković, T.: Congruences and homomorphisms of fuzzy automata. Fuzzy Sets and Systems 157, 444–458 (2006)

On Probability Distributions for Trees: Representations, Inference and Learning

François Denis[1], Amaury Habrard[1], Rémi Gilleron[2], Marc Tommasi[2], Édouard Gilbert[3]

[1]Laboratoire d'Informatique Fondamentale de Marseille (L.I.F.)
UMR CNRS 6166 — http://www.lif.univ-mrs.fr

[2]INRIA Futurs and Lille University, LIFL, Mostrare Project
http://www.grappa.univ-lille3.fr/mostrare

[3]ÉNS de Cachan, Brittany extension
INRIA Futurs and Lille University, LIFL, Mostrare Project

We study probability distributions over free algebras of trees. Probability distributions can be seen as particular *(formal power) tree series* [2, 7], i.e. mappings from trees to a semiring K. A widely studied class of tree series is the class of *rational* (or *recognizable*) tree series which can be defined either in an algebraic way or by means of multiplicity tree automata. We argue that the algebraic representation is very convenient to model probability distributions over a free algebra of trees. First, as in the string case, the algebraic representation allows to design learning algorithms for the whole class of probability distributions defined by rational tree series. Note that learning algorithms for rational tree series correspond to learning algorithms for weighted tree automata where both the structure and the weights are learned. Second, the algebraic representation can be easily extended to deal with unranked trees (like XML trees where a symbol may have an unbounded number of children). Both properties are particularly relevant for applications: nondeterministic automata are required for the inference problem to be relevant (recall that Hidden Markov Models are equivalent to nondeterministic string automata); nowadays applications for Web Information Extraction, Web Services and document processing consider unranked trees.

1 Representation Issues

Trees, either ranked or unranked, arise in many application domains to model data. For instance XML documents are unranked trees; in natural language processing (NLP), syntactic structure can often be considered as treelike. From a machine learning perspective, dealing with tree structured data often requires to design probability distributions over sets of trees. This problem has been addressed mainly in the NLP community with tools like probabilistic context free grammars [8].

Weighted tree automata and tree series are powerful tools to deal with tree structured data. In particular, probabilistic tree automata and stochastic series, which both define probability distributions on trees, allow to generalize usual techniques from probabilistic word automata (or hidden markov models) and series.

Tree Series and Weighted Tree Automata In these first two paragraphs, we only consider the case of ranked trees. A tree series is a mapping from the set of trees into some semiring K. Motivated by defining probability distributions, we mainly consider the case $K = \mathbb{R}$. A *recognizable tree series* [2] S is defined by a finite dimensional vector space V over K, a mapping μ which maps every symbol of arity p into a multilinear mapping from V^p into V (μ uniquely extends into a morphism from the set of trees into V), and a linear form λ. $S(t)$ is defined to be $\lambda(\mu(t))$. Tree series can also be defined by *weighted tree automata* (WTA). A WTA A is a tree automaton in which every rule is given a weight in K. For every run r on a tree t (computation of the automaton according to rules over t), a weight $A(t, r)$ is computed multiplying weights of rules used in the run and the final weight of the state at the root of the tree. The weight $A(t)$ is the sum of all $A(t, r)$ for all runs r over t.

For commutative semirings, recognizable tree series in the algebraic sense and in the automata sense coincide because there is an equivalence between summation at every step and summation over all runs. It can be shown, as in the string case, that the set of recognizable tree series defined by deterministic WTA is strictly included in the set of recognizable tree series. A Myhill-Nerode Theorem can be defined for WTA over fields [1].

Probability Distributions and Probabilistic Tree Automata A probability distribution S over trees is a tree series such that, for every t, $S(t)$ is between 0 and 1, and such that the sum of all $S(t)$ is equal to 1. Probabilistic tree automata (PTA) are WTA verifying normalization conditions over weights of rules and weights of final states. They extend probabilistic automata for strings and we recall that nondeterministic probabilistic string automata are equivalent to hidden Markov models (HMMs). As in the string case [5], not all probability distributions defined by WTA can be defined by PTA. However, we have proved that any distribution defined by a WTA *with non-negative coefficients* can defined by a PTA, too.

While in the string case, every probabilistic automaton defines a probability distribution, this is no longer true in the tree case. Similarly to probabilistic context-free grammars [9], probabilistic automata may define inconsistent (or improper) probability distributions: the probability of all trees is less than one. We have defined a sufficient condition for a PTA to define a probability distribution and a polynomial time algorithm for checking this condition.

Towards unranked trees Until this point, we only have considered ranked trees. However, unranked trees can be expressed by ranked ones using an isomorphism defined by an algebraic formulation ([3], chapter 8). It consists in using the right adjonction operator defined by $f(t_1,\ldots,t_{n-1})@t_n = f(t_1,\ldots,t_n)$; any tree can then be written as an expression whose only operator is @, and thus as a binary tree: e.g., $b(a,a,c(a,a))$ corresponds to $@(@(@(b,a),a),@(@(c,a),a))$. WTA for unranked trees can be defined as WTA for ranked trees applied to the algebraic formulation. We call such automata weighted stepwise tree automata (WSTA).

Hedge automata are automata for unranked trees. Each rule of a hedge automaton [3] is written $f(L) \to q$ where L is a regular language of word with the set of states of the automata as its alphabet. For weighted hedge automata (WHA), the weight of the rule $f(u) \to q$ is the product of a weight given to the whole rule $f(L) \to q$ and the weight of u according to a weighted word automata associated to $f(L) \to q$. When K is commutative, WSTA and WHA define the same weight distributions on unranked trees.

Probabilistic hedge automata can be defined by adding the same kind of summation conditions than on WHA, but it has yet to be shown that they can be expressed by PTA through algebraic formulation. We don't know yet weither defining series on unranked trees directly is possible, although it can be achieved using the algebraic formulation.

2 Learning Probability Distributions

Inference and Training PTA can be considered as generative models for trees. The two classical **inference problems** are : given a PTA A and given a tree t, compute $p(t)$ which is defined to the sum over all of all $p(t,r)$; and given a tree t, find the most likely (or Viterbi) labeling (run) $\hat{r}$ for t, i.e. compute $\hat{r} = \arg\max_r p(r|t)$. It should be noted that the inference problems are relevant only for nondeterministic PTA. The **training problem** is: given a sample set S of trees and a PTA, learn the best real-valued parameter vector (weights assigned to rules and to states) according to some criteria. For instance, the likelihood of the sample set or the likelihood of the sample over Viterbi derivations. Classical algorithms for inference (the message passing algorithm) and learning (the Baum-Welch algorithm) can be designed for PTA over ranked trees and unranked trees.

Learning Weighted Automata The **learning problem** extends over the training problem. Indeed, for the training problem, the structure of the PTA is given by the set of rules and only weights have to be found. In the learning problem, the structure of the target automaton is unknown. The learning problem is: given a sample set S of trees drawn according to a target rational probability distribution, learn a WTA according to some criteria. If the probability distribution is defined by a deterministic PTA, a learning algorithm extending over the unweighted case has been defined in [4]. However, this algorithm works only for deterministic PTA. We recall that the class of probability distributions defined by deterministic PTA is strictly included in the class of probability distributions defined by PTA [1].

Learning Recognizable Tree Series and thus learning WTA can be achieved thanks to an algorithm proposed by Denis and Habrard [6]. This algorithm, which benefits from the existence of a canonical linear representation of series, can be applied to series which take their values in $\mathbb{R}$ or $\mathbb{Q}$ to learn stochastic tree languages. It should be noted that the algebraic view allows to learn probability distributions defined by nondeterministic WTA. Learning probability distributions for unranked trees is ongoing work.

References

1. Björn Borchardt. The myhill-nerode theorem for recognizable tree series. In Zoltán Ésik and Zoltán Fülöp, editors, *Developments in Language Theory*, volume 2710 of *Lecture Notes in Computer Science*, pages 146–158. Springer Verlag, 2003.
2. Jean Berstel and Christophe Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18:115–148, 1982.
3. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.
4. Rafael C. Carrasco, José Oncina, and Jorge Calera-Rubio. Stochastic inference of regular tree languages. *Machine Learning*, 44(1/2):185–197, 2001.
5. François Denis, Yann Esposito, and Amaury Habrard. Learning rational stochastic languages. In Gabor Lugosi and Hans Ulrich Simon, editors, *Learning theory*, Lecture Notes in Computer Science. Springer Verlag, 2006.
6. François Denis and Amaury Habrard. Learning rational stochastic tree languages. In Markus Hutter, Rocco A. Servedio, and Eiji Takimoto, editors, *Algorithmic Learning Theory, 18th International Conference*, volume 4754 of *Lecture Notes in Artificial Intelligence*, pages 242–256. Springer Verlag, 2007.
7. Z. Esik and W. Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8:219 – 285, 2003.
8. C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, 1999.
9. C. S. Wetherell. Probabilistic languages: A review and some open questions. *ACM Comput. Surv.*, 12(4):361–379, 1980.

Prediction of Subalphabets and Ranking in DAWG's for Natural Languages

Alexander Eckl
Lehrstuhl für Informatik II, Universität Würzburg
Am Hubland, 97074 Würzburg, Germany
eckl@informatik.uni-wuerzburg.de

A new compressed representation of DAWG's (directed acyclic word graphs [2]) was developed for large sets of words for natural languages. The representation was used in applications like navigation in large digital encyclopedias or search applications, e. g. with up to more than 800,000 words.

The advantage of a DAWG is the fast access time for stored words ($O(n)$ with n: maximum length of the stored words). But the disadvantage is the memory consumption, especially if the DAWG is naively implemented by an array of $|\Sigma|$ pointers for each node ($|\Sigma| \times 4$ B per node), where $|\Sigma|$ is the size of the underlying alphabet. This is particularly true for natural languages, e. g. with alphabets of size 30 to 100, since in this case each node has, as an average, only a few children and most of the stored pointers are null.

One possible solution for the memory problem is the usage of bit vectors for each DAWG node. A bit in the vector of size $|\Sigma|$ is set if and only if the son of the corresponding character exists. For each node a bit vector and, optionally, pointers to existing sons and weights for ranking of the represented words are stored.

A new approach was developed for DAWG's with large alphabets. The technique was derived from text compression with finite context models. Algorithms like the PPM family (prediction by partial matching [1], [3]) are using preceding characters to predict and compress the following characters of a text. For example, in English texts it is very probable that `q` is followed by `u`. PPM is one of the best methods for text compression.

In an analogous manner the characters preceding a DAWG node are used to predict the local subalphabet of a node and a much smaller bit vector has to be stored.

References

1. J. G. Cleary and I. H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32:396–402, April 1984.
2. M. Crochemore and R. Vérin. Direct construction of compact directed acyclic word graphs. In A. Apostolico and J. Hein, editors, *Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching*, number 1264, pages 116–129, Aarhus, Denmark, 1997. Springer-Verlag, Berlin.
3. A. Moffat. Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, 38:1917–1921, 1990.

Weighted Tree-Walking Automata*

Zoltán Fülöp and Loránd Muzamel

Department of Foundations of Computer Science, University of Szeged
Árpád tér 2., H-6720 Szeged, Hungary
{fulop,muzamel}@inf.u-szeged.hu

The concept of a *tree-walking automaton* (for short: twa) was introduced in [1] for modelling the syntax-directed translations from strings to strings. Recently its importance grew in XML theory. A twa A is a sequential finite-state tree acceptor with finitely many transition rules, which, obeying its state-behaviour, walks along the edges of an input tree $s \in T_\Sigma$, where Σ is the input ranked alphabet of A. Then A accepts s if there is an *accepting run on* s, i.e., a finite walk on s from the initial state to the accepting state. The tree language recognized by a twa is effectively regular, however there exists a regular tree language that cannot be recognized by any twa [2]. There are several extensions of twa which still recognize regular tree languages, such as twa with weak pebbles [4], strong pebbles [5] and also invisible pebbles [6].

We introduce the weighted version of a twa. In a *weighted tree-walking automaton* A (for short: wtwa), every transition rule has a weight taken from a commutative semiring K. We assume that A is noncircular, i.e., it does not enter into a loop of transitions. The *weight of a run of* A *on an input tree* s is the product of the weights of the applied transition rules, while the *weight of* s *computed by* A is the sum of the weights of all the accepting runs of A on s. Since A is noncircular, it has only finitely many accepting runs on s. In this way, A recognizes a tree series $S_A : T_\Sigma \to K$, where $S_A(s)$ is the weight of s for every input tree s.

We investigate the recognizing power of wtwa. For this we consider the *reduced weighted MSO logic* of [3] which characterize effectively the class of regular tree series over a commutative semiring. We show that the tree series recognizable by noncircular wtwa can be defined in reduced weighted MSO logic.

References

1. A. V. Aho and J. D. Ullman. Translations on a context–free grammar. *Inform. Control*, 19:439–475, 1971.
2. M. Bojańczyk and T. Colcombet. Tree-walking automata do not recognize all regular languages. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing (STOC '05)*, pages 234–243, New York, NY, USA, 2005. ACM Press.
3. M. Droste and H. Vogler. Weighted tree automata and weighted logics. *Theoret. Comput. Sci.*, 366:228–247, 2006.
4. J. Engelfriet and H. J. Hoogeboom. Tree-walking pebble automata. In *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 72–83, London, UK, 1999. Springer-Verlag.
5. J. Engelfriet and Hendrik Jan Hoogeboom. Automata with nested pebbles capture first-order logic with transitive closure. Technical Report 05-02, Leiden University, The Netherlands, April 2005.
6. Joost Engelfriet, Hendrik Jan Hoogeboom, and Bart Samwel. Xml transformation by tree-walking transducers with invisible pebbles. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '07)*, pages 63–72, New York, NY, USA, 2007. ACM Press.

* This research was supported by the Hungarian Scientific Fund.

Varieties of Recognizable Tree Series over Fields*

Zoltán Fülöp[1] and Magnus Steinby[2]

[1]Department of Foundations of Computer Science, University of Szeged
Árpád tér 2., H-6720 Szeged, Hungary
fulop@inf.u-szeged.hu

[2]Department of Mathematics, University of Turku
FIN-20014 Turku, Finland
steinby@utu.fi

Our aim is to develop a theory of varieties of weighted tree languages. More specifically, we consider varieties of tree series of the kind studied by Berstel and Reutenauer [1], i.e., tree series over a field.

Let K be a field, Σ a ranked alphabet and X a leaf alphabet. Then a *$K\Sigma X$-tree series* is a map $S : T_\Sigma(X) \to K$, where $T_\Sigma(X)$ is the set of ΣX-trees, and a *$K\Sigma$-algebra* is a system $\mathcal{C} = (C, +, 0, \Sigma)$, where $(C, +, 0)$ is a K-vector space and each symbol $\sigma \in \Sigma$ is realized as a multilinear operation on C of the appropriate arity. Our syntactic algebras of tree series are essentially those considered in [2] and derived from the corresponding notion for string series [4]. Hence, the *syntactic $K\Sigma$-algebra* $\mathrm{SA}(S)$ of a recognizable $K\Sigma X$-tree series S is a finite-dimensional $K\Sigma$-algebra. Rather than using syntactic ideals as in [4, 2], we start with *syntactic congruences* that have a more obvious intuitive meaning. Similarly as in the case of ordinary tree languages (cf. [5]), it is convenient to present the basic theory of syntactic congruences and syntactic algebras for series over general Σ-algebras.

A *variety of $K\Sigma$-tree series* is a family $\mathcal{V} = \{\mathcal{V}(X)\}_X$ of tree series with certain natural closure properties, where for each X, $\mathcal{V}(X)$ is a set of recognizable $K\Sigma X$-tree series. The variety theorem, akin to Eilenberg's [3] fundamental theorem, establishes – via syntactic algebras – a correspondence between them and varieties of finite-dimensional $K\Sigma$-algebras.

References

1. Berstel, J. and Reutenauer, C.: Recognizable power series on trees. *Theoretical Computer Science* **18** (1982), 115–148.
2. Bozapalidis, S. and Alexandrakis, A.: Representation matricielles de séries d'arbre reconnaissables. *Theoretical Informatics and Applications* **23**(4) (1989), 449–459.
3. Eilenberg, S.: *Automata, Languages, and Machines.* Vol. B., Academic Press, New York 1976.
4. Reutenauer, C.: Séries formelles et algèbres syntactiques. *Journal of Algebra* **66** (1980), 448–483.
5. Steinby, M.: A theory of tree language varieties. *Tree Automata and Languages* (eds. M. Nivat and A. Podelski), North-Holland, Amsterdam 1992, 57–81.

* This research was supported by the Hungarian Scientific Fund.

Weighted Automata Define a Hierarchy of Terminating String Rewriting Systems

Andreas Gebhardt and Johannes Waldmann
Hochschule für Technik, Wirtschaft und Kultur (FH) Leipzig
Fb IMN, PF 30 11 66, D-04251 Leipzig, Germany

Rewriting is pattern replacement in context. It serves as a model of computation which is Turing-complete. Thus all "interesting" semantic properties are undecidable, including the very natural question of *termination*: for a given rewriting system, are all derivations finite? Since the problem is significant in practice, e.g. for the analysis of software, one is interested in semi-algorithms: computable methods of proving termination that are sound, but not complete.

One such method to prove termination of string rewriting is "matrix interpretation" [HW06]. These interpretations are in fact $\mathbb{N}$-weighted finite automata. The method has been generalized from string rewriting to term rewriting [EWZ06]. Several automated termination provers now implement this method.

The method in fact solves a more general problem: that of *relative termination.* A rewriting system R terminates relative to a rewriting system S if each mixed derivation (containing R and S steps in any order) contains only finitely many R steps. While being an interesting concept in itself, relative termination helps to solve standard termination problems because it allows to compose termination proofs: if R terminates relative to S then termination of $R \cup S$ follows from termination of S, and the latter can be proved separately. This corresponds to a lexicographic combination of interpretations.

One direction for extension of the matrix method is to pick a weight semi-ring that strictly includes $\mathbb{N}$. In [GHW07] we reported on some experiments with non-negative rationals. In the present note, we provide a basis for a systematic approach to compare these (and other) termination methods, by defining a suitable hierarchy, and we prove some of its properties.

Automata and Rewriting Systems. A weighted automaton A is called *weakly (strictly, resp.) compatible* with a rewriting system R if for each rewrite step $u \to_R v$, the sequence of weights $A(u), A(v)$ computed by the automaton is weakly (strictly, resp.) decreasing.

There is a local criterion on A that can effectively be checked and that implies compatibility as defined here. Basically, it is enough to compare interpretations of left-hand sides and right-hand sides of rules (as matrices).

If an automaton A with a well-founded weight domain W is strictly compatible with a rewriting system R and weakly compatible with a rewriting system S, then R is terminating relative to S.

A Notation for Termination Proofs by Rule Removals. We denote by $\mathfrak{M}(W, n)$ the set of pairs of rewriting systems (R, S) for which an automaton exists with weight domain W and n states that is strictly compatible with $R \setminus S$ and weakly compatible with S. We also write $R \left|\!\frac{\mathfrak{M}(W,n)}{}\right. S$. This notation indicates that the termination problem of R can be reduced to the termination problem of S by removing the rules in $R \setminus S$ due to an interpretation computed by an automaton with the given parameters.

The relational notation also suggests composability. For any sequence of rewriting systems R_i and relations P_i, from $R_0 \left|\!\frac{P_1}{}\right. \dots \left|\!\frac{P_n}{}\right. R_n$ it follows that $R_0 \setminus R_n$ terminates relative to R_n. If $R_n = \emptyset$, then R_0 terminates.

If $P_1 = \ldots = P_n = P$, we write $R_0 \left|\!\frac{P^*}{}\right. R_n$ or $(R_0, R_n) \in P^*$.

We abbreviate $\cup_{n \geq 1} \mathfrak{M}(W, n)$ by $\mathfrak{M}(W)$. Then in our notation $\mathfrak{M}(\mathbb{N})$ is the set of all rewriting systems that have a one-step termination proof using some natural-weighted automaton, and $\mathfrak{M}(\mathbb{N})^*$ is the set of all systems with a multi-step termination proof using such automata.

Number of States. For each $d \leq d'$, $\mathfrak{M}(W, d) \subseteq \mathfrak{M}(W, d')$. This follows easily since we can introduce useless states in an automaton of size d, to obtain an automaton of size d' that computes the same function.

Using the Amitsur-Levitzki theorem, for each d we find some $d' > d$ such that the inclusion is strict. This implies that the hierarchy is infinite. It remains open whether it is strict at each level. It is known that $\mathfrak{M}(W, 1) \subset \mathfrak{M}(W, 2) \subset \mathfrak{M}(W, 3)$.

Choice of Weight Domain. For each d, $\mathfrak{M}(\mathbb{N}, d) \subseteq \mathfrak{M}(\mathbb{Q}_{\geq 0}, d)$. This is clear since $\mathbb{N}$ is a sub-semiring of $\mathbb{Q}_{\geq 0}$.

We give an example $(R, S) \in \mathfrak{M}(\mathbb{Q}_{\geq 0}, 3)^2 \setminus \mathfrak{M}(\mathbb{N})^*$, that is, with a two-step termination proof of rational-weighted automata of size 3, but no natural-weighted termination proof of any size and number of steps.

Proofs with one or many steps. Obviously $\mathfrak{M}(W, d) \subseteq \mathfrak{M}(W, d)^*$.

There is an example $(R, \emptyset) \in \mathfrak{M}(\mathbb{N}, 2)^2 \setminus \mathfrak{M}(\mathbb{N})$, that is, a system with a two-step proof using a two-state automaton, but no one-step proof (for automata of any size) [HW06].

References

[EWZ06] Jörg Endrullis, Johannes Waldmann, and Hans Zantema. Matrix interpretations for proving termination of term rewriting. In Ulrich Furbach and Natarajan Shankar, editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 574–588. Springer, 2006.

[GHW07] Andreas Gebhardt, Dieter Hofbauer, and Johannes Waldmann. Matrix evolutions. In *Workshop on Termination (WST07)*, 2007.

[HW06] Dieter Hofbauer and Johannes Waldmann. Termination of string rewriting with matrix interpretations. In Frank Pfenning, editor, *RTA*, volume 4098 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2006.

A Survey of Integer Weighted Finite Automata

Vesa Halava, Tero Harju, Eero Lehtonen

Department of Mathematics
FI-20014 Turku, Finland
{vesa.halava, tero.harju, elleht}@utu.fi

The integer weighted automata, denoted by $FA(\mathbb{Z})$, are closely related to 1-turn automata as considered especially by Ibarra [3]. In our model the counter is replaced by a weight function of the transitions, and while doing so, the finite automaton becomes independent of the counter. To be precise, the weight function is calculated additively and the input is accepted if and only if the weight of its path is zero.

In this survey, we concentrate on undecidability results concerning integer weighted automata. First we show that the *universe problem* for the $FA(\mathbb{Z})$ is undecidable [1]. This is done by giving an explicit 4-stated unimodal integer weighted automaton that accepts every word in A^* if and only if a given instance of *Post Correspondence Problem* has a solution.

We also give a matrix representation [2] of integer weighted finite automata via Laurent polynomials. This leads to an analogue of a fundamental result in the theory of rational series and also gives an undecidability result for these matrices.

The main purpose of this survey is to present the basics and also some highlights of the theory of integer weighted automata. We will mainly focus on original considerations made by the first two authors. Also, some open problems are discussed.

References

1. V. Halava and T. Harju. *Undecidability in Integer Weighted Finite Automata*, Fundamenta Informaticae **39** (1999), 189–200.
2. V. Halava and T. Harju. *Undecidability in Matrices over Laurent Polynomials*, Adv. in Appl. Math. **33**, Issue 4 (2004), 747–752.
3. O. H. Ibarra. *Restricted one-counter machines with undecidable universe problems*, Math. Systems Theory **13** (1979), 181–186.

Statistical Language Models within the Algebra of Weighted Rational Languages

Thomas Hanneforth[1] and Kay-Michael Würzner[2]

[1]University of Potsdam

[2]Berlin-Brandenburg Academy of Science

Finite State Machines (FSMs) have been used in the field of statistical language modeling for a very long time. The majority of existing approaches uses them merely as a convenient data structure, mostly disregarding the underlying algebra with its well-defined operations like union and intersection. Instead of that, a bunch of specialized algorithms for the construction and application of statistical language models (LMs) are used (e.g. [1], [2]).

We think that this way of conceiving and using LMs is not desirable since it compromises the modularity of larger applications built on FSMs. Alternatively, the algebra of weighted rational languages (WRLs) and transductions (WRTs, cf. [3]) should suffice. As a case study, we present an alternative method for constructing LMs (N-gram models, including discounting, back-off, and interpolation as well as class-based and discontinuous N-gram models) in a completely algebraic way. Besides the usual rational operations, we need in addition only simple trivially weighted finite-state transducers depending on the order N and the alphabet Σ under consideration. These transducers map prefixes and/or suffixes of N-gram strings to ε or other symbols.

Creating an N-gram model consists of counting N-grams in a corpus, normalizing these counts, handling sparseness by smoothing, and finally applying an algorithm resulting in an weighted FSM accepting strings of arbitrary length. To give an impression how to handle all these steps within the algebra of weighted rational languages, we exemplify our approach focusing on the step of transforming N-gram frequencies into conditional probabilities.

The conditional probability of an N-gram is computed by equation (1).

$$\Pr(w_i|w_{i-N+1}^{i-1}) = \frac{\mathrm{C}(w_{i-N+1}^{i-1}w_i)}{\sum\limits_{w}\mathrm{C}(w_{i-N+1}^{i-1}w)} \tag{1}$$

Given an WRL $\mathcal{C}_N : \Sigma^N \rightarrow \mathbb{R}^+$ mapping N-grams to their frequencies, this normalization is performed in two steps: To represent the denominator in equation (1) we introduce an WRT $\mathcal{E}_N^k : \Sigma^N \times \Sigma^N \rightarrow \mathbb{R}^+$ which maps all k-gram suffixes to each other (what in effect assigns each weight to every symbol):

$$\mathcal{E}_N^k(x,y) = (\Sigma^{N-k} \cdot (\Sigma \times \Sigma)^k)(x,y) \tag{2}$$

Setting $k = 1$, the application $\mathcal{E}_N^1[\mathcal{C}_N]$ performs the summing over the unigram suffixes of all N-grams sharing the same $N-1$-gram prefix as demanded by equation (1)*. The second step is the division of the corresponding N and $N-1$-gram counts. To model this arithmetic operation, one can take advantage of two properties of the real semiring ($\mathcal{R} = \langle\mathbb{R}^+, +, \cdot, 0, 1\rangle$): 1) The abstract semiring multiplication $\otimes$ is instantiated in $\mathcal{R}$ with the actual multiplication of real numbers. 2) $\mathcal{R}$ is a division semiring ([4]) that is, $\forall a \neq \overline{0} \in \mathbb{K}$, $\exists b \in \mathbb{K}$ such that $a \otimes b = \overline{1}$. Given these properties and the fact that weighted intersection combines weights by $\otimes$ ([5]), it is possible to represent division by intersection with multiplicative inverses. We therefore introduce an operation called $\otimes$-negation denoted by $^{-1}$ which replaces every weight with its multiplicative inverse. It is now possible to define the following WRL $\mathcal{P}_N^c : \Sigma^N \rightarrow \mathbb{R}^+$ which represents the conditional probabilities of the counts in $\mathcal{C}_N$.

$$\mathcal{P}_N^c(x) = \mathcal{C}_N \cap (\mathcal{E}_N^1[\mathcal{C}_N])^{-1}(x) \tag{3}$$

* Note that the application $\mathcal{T}[\mathcal{L}]$ is an abbreviation for the second projection of the composition of $ID(\mathcal{L})$ and $\mathcal{T}$.

Our approach is implemented on the basis of weighted finite-state machines (WFSM) corresponding to the given WRLs. This enables us to make use of the usual optimization procedures for WFSMs. Moreover, the character of the given WRLs allows for representing the corresponding WFSMs in a virtual way permitting access to states and transitions in constant time and consuming only a constant amount of memory independent of the size of N and Σ. The cross-product of Σ used in $\mathcal{E}_N^k$ would otherwise need a quadratic number of transitions relative to the size of Σ.

We will show that the complexity of the complete algebraic specification – which leads to a minimal LM, as long as the N-gram frequencies are represented as a minimal WFSM – is linear in the size of the given training corpus.

References

1. Giuseppe Riccardi, Roberto Pieraccini, and Enrico Bocchieri. Stochastic Automata for Language Modeling. *Computer Speech & Language*, 10(3):265–293, 1996.
2. David Llorens, Juan Miguel Vilar, and Francisco Casacuberta. Finite State Language Models Smoothed Using n-Grams. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(3):275–289, 2002.
3. Fernando C.N. Pereira and Michael D. Riley. Speech Recognition by Composition of Weighted Finite Automata. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, volume 12 of *Language, Speech, and Communication*, chapter 15, pages 433–453. The MIT Press, Cambridge, MA, 1997.
4. Jason Eisner. Simpler and More General Minimization for Weighted Finite-State Automata. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 64–71, Morristown, NJ, 2003. Association for Computational Linguistics.
5. Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted Automata in Text and Speech Processing. In William Wahlster, editor, *ECAI 96. 12th European Conference on Artificial Intelligence*. John Wiley & Sons, Ltd., 1996.

Relationships between FFA-recognizability and DFA-recognizability of Fuzzy Languages

Jelena Ignjatović[1], Miroslav Ćirić[1], and Tatjana Petković[2]

[1]Faculty of Sciences and Mathematics, University of Niš
Višegradska 33, 18000 Niš, Serbia
jejaign@yahoo.com, ciricm@bankerinter.net

[2]Nokia
Joensuunkatu 7, FIN-24100 Salo, Finland
tatjana.petkovic@nokia.com

In this talk we will present the results from [4, 5], concerning relationships between recognizability of fuzzy languages by fuzzy finite automata (FFA-recognizability) and their recognizability by deterministic finite automata (DFA-recognizability). Equivalence between FFA-recognizability and DFA-recognizability of fuzzy languages over a locally finite complete lattice was established by Bělohlávek [1]. A more general result was obtained by Li and Pedrycz [6], who studied fuzzy automata over a lattice ordered monoid $\mathscr{L}$, and proved that FFA-recognizability is equivalent to DFA-recognizability if and only if the semiring reduct $\mathscr{L}^*$ of $\mathscr{L}$ (with respect to the join and multiplication operations) is locally finite. Bělohlávek [1] and Li and Pedrycz [6] gave a method for determinization of fuzzy automata, which results in a finite automaton if and only if $\mathscr{L}^*$ is locally finite. Another method, developed in [4] for fuzzy languages over a complete residuated lattice $\mathscr{L}$, can result in a finite automaton even if $\mathscr{L}^*$ is not locally finite, and always gives a smaller automaton than the method by Bělohlávek and Li and Pedrycz. Certain criterions for finiteness of the resulting deterministic automaton have been obtained in [4, 5], where it has been shown that this automaton is a minimal deterministic automaton recognizing all fuzzy languages which can be recognized by the original fuzzy automaton.

In [5] the authors studied DFA-recognizability of fuzzy languages with membership values in an arbitrary set having two distinguished elements 0 and 1, which are needed to take crisp languages into consideration. DFA-recognizability of these fuzzy languages has been characterized through their syntactic right congruences and syntactic congruences, and it has been proved that for any fuzzy language there exists a minimal deterministic automaton recognizing it, which is unique up to an isomorphism. This automaton has been constructed by means of derivatives of a fuzzy language, as well as by means of derivatives of certain crisp languages associated with a fuzzy language (kernel and cut languages), and an algorithm for minimization of a deterministic automaton which recognizes a fuzzy language has been given. A similar algorithm, for deterministic automata recognizing fuzzy languages over a distributive lattice, has been recently given by Li and Pedrycz [7].

Recognizability of fuzzy languages by finite monoids (FM-recognizability) has been recently studied by Bozapalidis and Louscou-Bozapalidou [2, 3], who have established certain relationships between FM-recognizability and FFA-recognizability of fuzzy languages. In [5] the authors have shown that FM-recognizability of fuzzy languages is equivalent to DFA-recognizability.

Acknowledgment. Research supported by Ministry of Science, Republic of Serbia, Grant No. 144011

References

1. Bělohlávek, R.: Determinism and fuzzy automata. Information Sciences 143, 205–209 (2002)
2. Bozapalidis, S., Louscou-Bozapalidou, O.: On the recognizability of fuzzy languages I. Fuzzy Sets and Systems 157, 2394–2402 (2006)
3. Bozapalidis, S., Louscou-Bozapalidou, O.: On the recognizability of fuzzy languages II. Fuzzy Sets and Systems 159, 107-113 (2008)

4. Ignjatović, J., Ćirić, M., Bogdanović, S.: Determinization of fuzzy automata with membership values in complete residuated lattices. Information Sciences 178, 164–180 (2008)
5. Ignjatović, J., Ćirić, M., Bogdanović, S., Petković, T.: Myhill-Nerode type theory for fuzzy languages and automata. submitted for publication
6. Li, Y.M., Pedrycz, W.: Fuzzy finite automata and fuzzy regular expressions with membership values in lattice ordered monoids. Fuzzy Sets and Systems 156, 68–92 (2005)
7. Li, Y.M., Pedrycz, W.: Minimization of lattice finite automata and its application to the decomposition of lattice languages. Fuzzy Sets and Systems 158, 1423–1436 (2007)

Deciding Unambiguity and Sequentiality from a Polynomially Ambiguous min-plus Automaton*

Daniel Kirsten[1,†] and Sylvain Lombardy[2]

[1]University Leipzig, Institute for Computer Science
04009 Leipzig, Germany
www.informatik.uni-leipzig.de/~kirsten/

[2]Institut Gaspard Monge, Université de Marne-la-Vallée
77454 Marne-la-Vallée Cedex 2, France
igm.univ-mlv.fr/~lombardy/

The sequentiality/unambiguity problem is one of the most intriguing open problems in the theory of min-plus automata: decide (constructively) whether some given min-plus automaton admits a sequential/unambiguous equivalent. This problem is wide open despite it was studied by several researchers, e.g. [1, 2, 4, 5].

In 2004, it was shown by Klimann, Lombardy, Mairesse, and Prieur that the sequentiality/unambiguity problem is decidable for finitely ambiguous min-plus automata [2].

The class of polynomially ambiguous min-plus automata lies strictly between the classes of finitely ambiguous and arbitrary min-plus automata.

In the talk, we generalize the result from [2] by showing that the sequentiality/unambiguity problem is decidable for polynomially ambiguous min-plus automata. For this, we have to handle several problems:

1. The equivalence problem for polynomially ambiguous min-plus automata is undecidable [3]. Thus, we cannot decide state equivalence in our proofs.
2. The key construction in [2] relies on a decomposition of the given finitely ambiguous min-plus automaton into a finite family of unambiguous min-plus automata. Such a decomposition is not possible for polynomially ambiguous min-plus automata. We overcome this problem by developing a theory of so-called metatransitions.
3. The proof in [2] relies on pumping techniques. We can show by an example that pumping techniques are not sufficient to decide the sequentiality/unambiguity problem for finitely ambiguous min-plus automata. We develop nested pumping techniques which lead us to interesting Burnside type problems for matrices over the tropical semiring.

References

1. I. Klimann, S. Lombardy, J. Mairesse, and C. Prieur. Deciding the sequentiality of a finitely ambiguous max-plus automaton. In Z. Ésik and Z. Fülöp, editors, *DLT'03 Proceedings*, volume 2710 of *Lecture Notes in Computer Science*, pages 373–385. Springer-Verlag, Berlin, 2003.
2. I. Klimann, S. Lombardy, J. Mairesse, and C. Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004.
3. D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4(3):405–425, 1994.
4. S. Lombardy and J. Sakarovitch. Sequential? *Theoretical Computer Science*, 356:224–244, 2006.
5. M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:269–311, 1997.

* A full paper is available on the authors homepages.

† The main results were achieved during a three months stay of the author at the Institute Gaspard-Monge at the Université Marne-la-Vallée which was funded by the CNRS.

Max/Plus Tree Automata for Termination of Term Rewriting

Adam Koprowski[1] and Johannes Waldmann[2]

[1]TU Eindhoven, The Netherlands
http://www.win.tue.nl/~akoprows/

[2]HTWK Leipzig, Germany
http://www.imn.htwk-leipzig.de/~waldmann/

Term rewriting is a model of computation. It serves as the basis for functional programming and for formal (algebraic) specification. Termination of rewriting therefore is an interesting property. It is undecidable in general, but there are several semi-algorithms, used by automated termination provers.

One method of proving termination is interpretation into a well-founded algebra. While polynomial interpretations (over the naturals) are well-known, a recent development is the matrix method [HW06,EWZ06] that uses linear interpretations over vectors of naturals, equivalently, $\mathbb{N}$-weighted automata. In [Wal06,Wal07] we extended this method (for string rewriting) to arctic automata, i.e. on the max/plus semi-ring on $\{-\infty\} \cup \mathbb{N}$. Its implementation in the termination prover Matchbox [Wal04] contributed to this prover winning the string rewriting division of the 2007 termination competition [Mar04].

The first contribution of the present work is a *generalization of arctic termination to term rewriting.* We use interpretations given by weighted tree automata. We restrict to the special case of automata where each transition function is of the form $(x_1, \ldots, x_n) \mapsto M_0 + M_1 \cdot x_1 + \ldots + M_n \cdot x_n$. Here, x_i are (column) vector variables, M_0 is a vector and $M_1, \ldots$ are square matrices. Operations are understood in the semi-ring.

Since the max operation is not strictly monotonic in single arguments, we do not obtain monotone interpretations, but only weakly monotone interpretations. These cannot prove termination, but only top termination, where rewriting steps are only applied at the root of terms. This is a restriction but it fits with the framework of the dependency pairs method [AG00] that transforms a termination problem to a top termination problem.

The second contribution is a *generalization from arctic naturals to arctic integers*, i.e. $\{-\infty\} \cup \mathbb{Z}$. Arctic integers allow e.g. to interpret function symbols by the predecessor function, and this matches the "intrinsic" semantics of some termination problems. There is previous work on polynomial interpretations with negative coefficients [HM04]. It uses ad-hoc max operations in several places. The semi-ring of arctic integers provides a general framework (under the restriction that the polynomials are linear).

The third contribution is that all definitions, theorems and proofs have been *formalized with the proof assistant Coq* [BC04]. This extends previous work [KZ08] and will become part of the CoLoR project [BDCG$^+$06] that gathers formalizations of termination techniques and employs them to certify termination proofs found automatically. In 2007, the certified category of the termination competition was won by the termination prover TPA [Kop06] that uses CoLoR.

A method to search for arctic interpretations is implemented for the termination prover Matchbox. It works by transformation to a boolean satisfiability problem, and applying a state-of-the-art SAT solver. For several termination problems that could not be solved in last year's termination competition it finds proofs via arctic tree automata and the new CoLoR version certifies them.

References

[AG00] Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theor. Comput. Sci.*, 236(1-2):133–178, 2000.

[BC04] Yves Bertot and Pierre Casteran. *Interactive Theorem Proving and Program Development.* Springer, 2004.

[BDCG+06] Frédéric Blanqui, William Delobel, Solange Coupet-Grimal, Sébastien Hinderer, and Adam Koprowski. CoLoR, a Coq library on rewriting and termination. In *Eighth International Workshop on Termination (WST 06')*, 2006. http://color.loria.fr.

[EWZ06] Jörg Endrullis, Johannes Waldmann, and Hans Zantema. Matrix interpretations for proving termination of term rewriting. In Ulrich Furbach and Natarajan Shankar, editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 574–588. Springer, 2006.

[HM04] Nao Hirokawa and Aart Middeldorp. Polynomial interpretations with negative coefficients. In Bruno Buchberger and John A. Campbell, editors, *AISC*, volume 3249 of *Lecture Notes in Computer Science*, pages 185–198. Springer, 2004.

[HW06] Dieter Hofbauer and Johannes Waldmann. Termination of string rewriting with matrix interpretations. In Frank Pfenning, editor, *RTA*, volume 4098 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2006.

[Kop06] Adam Koprowski. TPA: Termination proved automatically. In Frank Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006, Seattle, WA, USA, August 12-14, 2006, Proceedings*, volume 4098 of *Lecture Notes in Computer Science*, pages 257–266. Springer, 2006. current information at http://www.win.tue.nl/tpa/.

[KZ08] Adam Koprowski and Hans Zantema. Certification of proving termination of term rewriting by matrix interpretations. In Viliam Geffert, Juhani Karhumäki, Alberto Bertoni, Bart Preneel, Pavol Návrat, and Mária Bieliková, editors, *SOFSEM*, volume 4910 of *Lecture Notes in Computer Science*, pages 328–339. Springer, 2008.

[Mar04] Claude Marche. Termination competition. http://www.lri.fr/ marche/termination-competition/, 2004.

[Wal04] Johannes Waldmann. Matchbox: A tool for match-bounded string rewriting. In Vincent van Oostrom, editor, *Proc. 15th Int. Conf. Rewriting Techniques and Applications RTA-98*, number 3091 in Lecture Notes in Comput. Sci., pages 85–94, 2004. current information at http://dfa.imn.htwk-leipzig.de/matchbox/.

[Wal06] Johannes Waldmann. Weighted automata for termination of string rewriting. In *Workshop on Weighted Automata Theory and Applications (WATA06)*, 2006.

[Wal07] Johannes Waldmann. Arctic termination. In *Workshop on Termination (WST07)*, 2007. http://dfa.imn.htwk-leipzig.de/matchbox/methods/arctic.pdf.

From unweighted to weighted traces
– alternative proofs –

Dietrich Kuske
Institut für Informatik
Universität Leipzig

A large body of theoretical computer science deals with properties of languages as sets of finite words. These words can be understood as the sequence of events performed by some system. This modelling works fine for sequential systems because of the linear nature of words. Mazurkiewicz, in 1977, proposed a generalization of words nowadays called Mazurkiewicz traces that allows to also model some concurrency. Since its introduction, much work has been devoted to the transfer of results on word languages to trace languages (cf. the handbook "The book of Traces"). One such result is Kleene's theorem equating the recognizable and the rational languages. Ochmański succeeded in transfering this result to trace languages showing that the recognizable trace languages are precisely the c-rational ones.

For sequential systems, it is not just interesting to ask whether a particular word is generated, but also to know the number of different ways it can be generated. This question developed into the theory of weighted automata and formal power series. A fundamental result is Schützenberger's theorem from 1961, equating the behaviors of weighted automata with the set of rational formal power series.

These two distinct generalizations of Kleene's theorem were re-joint by Droste & Gastin in 1999 who investigated weighted trace automata and formal power series over partially commuting variables.

The theorems by Kleene, by Schützenberger, by Ochmański, and by Droste & Gastin show that all rational languages, formal power series, trace languages, or formal power series over partially commuting variables are recognizable. All the proofs follow the line of Kleene's proof (namely showing the closure of recognizable objects with respect to all rational operations) albeit with non-trivial additions.

In this talk, we present an alternative proof of Droste & Gastin's characterisation of the behavior of weighted trace automata. The *main novelty* lies in the fact that we derive their result as a corollary to Ochmański's theorem. In other words, we derive a result on *weighted* trace automata from a theorem on *unweighted* trace automata.

Weighted tree automata with discounting

Eleni Mandrali and George Rahonis

Department of Mathematics, Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
{elemandr,grahonis}@math.auth.gr

We introduce the model of *weighted top-down tree automata* (WTTA for short) *with discounting*. These automata are usual weighted top-down tree automata, where the *discounted weight* of a run on an input tree, is computed by discounting the weight of every node according to its distance from the root of the tree. More precisely, for a ranked alphabet Σ and a semiring K, a Φ-discounting over Σ and K is a family $\Phi = (\Phi_\sigma)_{\sigma \in \Sigma}$ of endomorphisms of K indexed by the alphabet Σ. Given a WTTA $\mathcal{M}$ over Σ and K, a tree $t \in T_\Sigma$, and a run r_t of $\mathcal{M}$ over t, the weight of every node $w \in dom(r_t)$ is discounted by Φ according to the path from the root of t to w. In this way, the nodes occurring at the same level of the tree get a weight discounted by the same grade. We show that the class $K^{\Phi-rec}\langle\langle T_\Sigma \rangle\rangle$ of formal power tree series recognized by WTTA over a ranked alphabet Σ and a commutative semiring K with a Φ-discounting, coincides with the class of Φ-rational tree series over Σ and K, i.e. a Kleene theorem. Here, for our Φ-rational tree operations, it suffices to incorporate the Φ-discounting in top-concatenation and in α-concatenation. By considering the identity discounting, we obtain as a special case the Kleene theorem of Droste, Pech and Vogler [2]. Furthermore, by applying our result to monadic ranked alphabets (i.e. ranked alphabets with symbols of rank 0 and 1), we get the Kleene-Schützenberger theorem of Droste and Kuske [1] for skew word series.

Then, we introduce a weighted MSO logic with discounting for finite trees. In fact, we use the logics of Droste and Vogler [4] and we incorporate the discounting only in the semantics of the first order universal quantifications. For this logic, we prove the expressive equivalence of Φ-recognizable tree series with two fragments of Φ-definable sentences. The first one called *restricted* is semantically determined. The latter called *almost existential* is syntactically defined, and for the equivalence result we require that the additive monoid of the underlying semiring is locally finite.

For our constructions it is convenient to work with weighted bottom-up tree automata with discounting. Trivially the classes of the behaviors of the two models coincide.

In the second part of the paper, we consider weighted Muller tree automata with discounting over the max-plus $\mathbb{R}_{\max}$ and the min-plus $\mathbb{R}_{\min}$ semirings. By using the discounting parameters, we get rid of the completeness axioms of the underlying semirings required in [5]. Then, we enrich our weighted MSO logics (for finite trees) with the formulas $x = y$ and $\forall X . \varphi$, but we still discount the semantics of the weighted formulas only in the first order universal quantifications. We show that the class of discounted Muller recognizable tree series coincides with two fragments of weighted sentences of our logics. The *restricted* which is defined semantically and the *incomplete universal* which is defined syntactically. If we restrict ourselves to monadic alphabets, then we can drop the second order universal quantifiers from our logic. Therefore, we obtain as a special case a recent result of Droste and Rahonis [3] connecting discounted ω-recognizable word series with infinitary series definable by discounted MSO-sentences. This fact highlights the robustness of the theory of discounted weighted logics for infinite words and trees.

References

1. M. Droste, D. Kuske, Skew and infinitary formal power series, *Theoret. Comput. Sci.* 366(2006) 199-227.
2. M. Droste, C. Pech, H. Vogler, A Kleene theorem for weighted tree automata, *Theory of Computing Systems* 38(2005) 1-38.
3. M. Droste, G. Rahonis, Weighted automata and weighted logics with discounting, in: *Proceedings of CIAA 2007, LNCS* 4783(2007) 73-84

4. M. Droste, H. Vogler, Weighted tree automata and weighted logics, *Theoret. Comput. Sci.* 366(2006) 228-247.
5. G. Rahonis, Weighted Muller tree automata and weighted logics, *J. Autom. Lang. Comb.*, in press.

Weighted Logics for Nested Words and Algebraic Formal Power Series

Christian Mathissen

Institut für Informatik, Universität Leipzig
D-04009 Leipzig, Germany
mathissen@informatik.uni-leipzig.de

Model checking of finite state systems has become an established method for automatic hardware and software verification and led to numerous verification programs used in industrial application. In order to verify recursive programs it is necessary to model them as pushdown systems rather than finite automata. This has motivated Alur and Madhusudan [3, 4] to define the classes of nested word languages and visibly pushdown languages, which is a proper subclass of the context-free languages and exceeds the regular languages. These classes gained huge interest and set a starting point for a new research field, see e.g. [1, 2, 5, 6] among many others.

The goal of this contribution will be: 1. to introduce a quantitative automaton model and a quantitative logic for nested words being equally expressive, 2. to establish a connection between nested words and series-parallel-biposets which have been studied by Ésik & Németh [9] and Hashiguchi et. al. (e.g. [10]) and 3. to give a characterization algebraic formal power series by means of weighted logic.

On order to be able to model quantitative properties of systems, extensions of existing models to quantitative models as for example weighted automata have been investigated. We introduce and investigate weighted nested word automata which we propose as a quantitative model for sequential programs with recursive procedure calls. Due to the fact that we define them over arbitrary semirings they are very flexible and can e.g. model probabilistic or stochastic systems. As the first main result, we characterize their expressiveness using weighted logic as introduced by Droste and Gastin [8], generalizing a result of Alur and Madhusudan.

To show our result we establish a new connection between so-called series-parallel-biposets and nested words. The class of sp-biposets forms the free bisemigroup which has been investigated by Hashiguchi et. al. (e.g. [10]) and a language theory for series-parallel-biposets has been developed by Ésik and Németh [9]. We anticipate that the connection between nested words and sp-biposets can be utilized to obtain further results.

Using projections of nested word series and applying the above mentioned result we obtain the second main result, a characterization of algebraic formal power series in terms of weighted logic generalizing a result of Lautemann, Schwentick and Thérien [13] for context-free languages.

References

1. R. Alur, S. Chaudhuri, and P. Madhusudan. A fixpoint calculus for local and global program flows. In *Proc. of the 33rd ACM POPL, Charleston*, pages 153–165. ACM, 2006.
2. R. Alur, V. Kumar, P. Madhusudan, and M. Viswanathan. Congruences for visibly pushdown languages. In *Proc. of the 32nd ICALP, Lisbon*, volume 3580 of *Lecture Notes in Computer Science*, pages 1102–1114, 2005.
3. R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proc. of the 36th STOC, Chicago*, pages 202–211. ACM, 2004.
4. R. Alur and P. Madhusudan. Adding nesting structure to words. In *Proc. of the 10th DLT, Santa Barbara*, volume 4036 of *Lecture Notes in Computer Science*, pages 1–13, 2006.
5. M. Arenas, P. Barceló, and L. Libkin. Regular languages of nested words: Fixed points, automata, and synchronization. In *Proc. of the 34th ICALP, Wroclaw*, volume 4596 of *Lecture Notes in Computer Science*, pages 888–900, 2007.
6. V. Bárány, C. Löding, and O. Serre. Regularity problems for visibly pushdown languages. In *Proc. of the 23rd STACS, Marseille*, volume 3884 of *Lecture Notes in Computer Science*, pages 420–431, 2006.

7. N. Chomsky and M.P. Schützenberger. The algebraic theory of context-free languages. In *Computer Programming and Formal Systems*, pages 118–161. Amsterdam, North-Holland, 1963.
8. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380:69–86, 2007.
9. Z. Ésik and Z. L. Németh. Higher dimensional automata. *Journal of Automata, Languages and Combinatorics*, 9(1):3–29, 2004.
10. K. Hashiguchi, S. Ichihara, and S. Jimbo. Formal languages over free bionoids. *Journal of Automata, Languages and Combinatorics*, 5(3):219–234, 2000.
11. W. Kuich. Semirings and formal power series. In G. Rozenberg and A. Salomaa, editors, *Word, Language, Grammar*, volume 1 of *Handbook of Formal Languages*, chapter 9, pages 609–677. Springer-Verlag, 1997.
12. W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1986.
13. C. Lautemann, T. Schwentick, and D. Thérien. Logics for context-free languages. In *Proc. of the 8th CSL, Kazimierz*, volume 933 of *Lecture Notes in Computer Science*, pages 205–216, 1994.
14. C. Mathissen. Definable transductions and weighted logics for texts. In *Proc. of the 11th DLT, Turku*, volume 4588 of *Lecture Notes in Computer Science*, pages 324–336, 2007.

On the expressive power of a weighted μ-calculus

Ingmar Meinecke

Institut für Informatik, Universität Leipzig
04009 Leipzig, Germany
meinecke@informatik.uni-leipzig.de

The μ-calculus (cf. [1, 6]) is a well-established and important notion in computer science. It combines advantages both of logic (a well-structured notation) and of automata (algorithmic problems are solved by computing fixed points). Different temporal logics are a fragment of the μ-calculus.

In recent years, multi-valued and weighted logics attracted more and more interest. A weighted monadic second-order logic over finite words was introduced [3]. Here, weights from an arbitrary commutative semiring are appended. A fragment of this logic turned out as semantically equivalent to the behaviors of weighted finite automata. But for the description of temporal properties the use of modal operators seems more reasonable. Several papers (cf. [5, 2, 4, 7]) deal with such multi-valued temporal logics and attack the model checking problem in a multi-valued setting. The values are taken from certain finite distributive lattices $\mathcal{L}$ (De Morgan algebras). Then multi-valued Kripke structures are considered, i.e., atomic propositions in the states and/or the transitions of the structure take values in $\mathcal{L}$. For several temporal logics and the μ-calculus over these multi-valued Kripke structures the model checking problem was solved (either by a reduction to the classical case or by attacking it directly).

Here, we turn our attention to the expressive power. We define a weighted μ-calculus on finite and infinite words and show the coincidence of a conjunction-free fragment with the class of ω-rational formal power series. Hereby, the weights are taken from a distributive complete lattice. Moreover, we discuss for which other semirings the result may carry over.

References

1. A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.
2. G. Bruns and P. Godefroid. Model checking with multi-valued logics. In *Proceedings of ICALP 2004 (31st International Colloquium on Automata, Languages and Programming)*, volume 3142 of *Lect. Notes in Comp. Sc.*, pages 281–293. Springer, 2004.
3. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380:69–86, 2007.
4. G. E. Fainekos. An introduction to multi-valued model checking. Technical Report MS-CIS-05-16, Dept. of CIS, University of Pennsylvania, September 2005.
5. A. Gurfinkel and M. Chechik. Multi-valued model checking via classical model checking. In *Proceedings of CONCUR 2003*, volume 2761, pages 266–280. Springer, 2003.
6. D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
7. O. Kupferman and Y. Lustig. Lattice automata. In *Proceedings of Verification, Model Checking, and Abstract Interpretation, 8th International Conference, VMCAI 2007, Nice, France*, volume 4349 of *Lect. Notes in Comp. Sc.*, pages 199–213. Springer, 2007.

A Kleene-Schützenberger Theorem for Weighted Timed Automata

Karin Quaas and Manfred Droste

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

During the last years, weighted timed automata (wta) have received much interest in the real-time community. Weighted timed automata are an extension of timed automata [2] and allow to assign weights (costs) to both locations and edges. This model has been introduced independently by Alur et al. [3] and Behrmann et al. [4]. It allows the modelling of continuous consumption of resources, and thus, enables to represent e.g. scheduling and planning problems. Consequently, there has been much research on problems as optimal reachability and model checking [8], [10], [1]. However, there has been no algebraic characterization of the behaviour of wta so far. We attempt to fill this gap by providing a Kleene-Schützenberger theorem for wta [14]. We apply the theory of weighted finite automata [6], [16], [15], and define wta over a semiring, resulting in a model that subsumes previous definitions in the literature, e.g. [3], where costs for reaching a location are computed by taking the infimum of the running weights of all runs, or [9], a multi-priced variant of a wta. For giving a Kleene-Schützenberger theorem, we combine the approach of Schützenberger [17] as well as a recent approach of a Kleene-type theorem for (unweighted) timed automata by Bouyer and Petit [11]. Our main result also implies Kleene-type theorems for several subclasses of wta, i.e., weighted finite automata, timed automata, timed automata with stopwatch observers [12].

Currently, we are investigating whether there is a Büchi-type theorem for wta, i.e., are wta expressively equivalent to some weighted timed version of monadic second-oder logic. For this we are trying to combine methods of Wilke [18], Droste and Gastin [13] and Bouyer [7].

References

1. R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability in weighted timed games. In J. Díaz, J. Karhumäki, A. Lepistö, and D. Sannella, editors, *ICALP*, volume 3142 of *LNCS*, pages 122–133. Springer, 2004.
2. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
3. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In Benedetto and Sangiovanni-Vincentelli [5], pages 49–62.
4. G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In Benedetto and Sangiovanni-Vincentelli [5], pages 147–161.
5. M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors. *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 2001, Proceedings*, volume 2034 of *LNCS*. Springer, 2001.
6. J. Berstel and C. Reutenauer. *Rational Series and their Languages.* Springer-Verlag New York, Inc., New York, NY, USA, 1988.
7. P. Bouyer. A logical characterization of data languages. *Information Processing Letters*, 84(2):75–85, October 2002.
8. P. Bouyer, T. Brihaye, V. Bruyère, and J.-F. Raskin. On the optimal reachability problem on weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, October 2007.
9. P. Bouyer, E. Brinksma, and K. G. Larsen. Optimal infinite scheduling for multi-priced timed automata. *Formal Methods in System Design*, 2007. To appear.
10. P. Bouyer, K. G. Larsen, and N. Markey. Model-checking one-clock priced timed automata. In H. Seidl, editor, *FoSSaCS*, volume 4423 of *LNCS*, pages 108–122. Springer, 2007.

11. P. Bouyer and A. Petit. A Kleene/Büchi-like theorem for clock languages. *J. Autom. Lang. Comb.*, 7(2):167–186, 2001.
12. T. Brihaye, V. Bruyère, and J.-F. Raskin. On model-checking timed automata with stopwatch observers. *Inf. Comput.*, 204(3):408–433, 2006.
13. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380(1-2):69–86, 2007.
14. M. Droste and K. Quaas. A Kleene-Schützenberger theorem for weighted timed automata. In R. Amadio, editor, *FoSSaCS 2008*, volume 4962 of *LNCS*, pages 142–156. Springer, 2008. To appear.
15. W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1986.
16. A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer New York, 1978.
17. M. P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.
18. T. Wilke. Specifying Timed State Sequences in Powerful Decidable Logics and Timed Automata. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *LNCS*, pages 694–715, Lübeck, Germany, 1994. Springer-Verlag.

Recognizability of Iterative Picture Languages

Sibylle Schwarz and Renate Winter
Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg, Germany
[schwarzs,winter]@informatik.uni-halle.de

Picture languages generated by an iterative process occur in image compression [11] and the creation of fractal pictures [6].

Every letter (m, n) in the alphabet $A_k = \{0, \ldots, k-1\}^2$ is interpreted as a position in a $k \times k$-square and every word $w \in A_k^*$ as position

$$\mathsf{Pos}(w) = \left(\sum_{i=0}^{|w|-1} \pi_1(w_i) k^{|w|-i-1}, \sum_{i=0}^{|w|-1} \pi_2(w_i) k^{|w|-i-1} \right)$$

in a square of side $k^{|w|}$ where $|w|$ is the length of the word $|w|$.

For a semiring W, we will consider pictures with "colors" in W, i.e. functions $p : \{0, \ldots, m\} \times \{0, \ldots, n\} \to W$. Then every W-valued word language $L : A_k^* \to W$ (also known as formal power series [1]) defines a picture language

$$\mathsf{picture}(L) = \{p_i : \{0, \ldots, k^i - 1\}^2 \to W \mid i \in \mathbb{N}\}$$

where for every $i \in \mathbb{N}$ and every $(m, n) \in \{0, \ldots, k^i - 1\}^2$

$$p_i(m, n) = L\left(\mathsf{Pos}^{-1}(m, n)\right),$$

i.e. $p_i(m, n)$ is the value in L of the word addressing position (m, n). Picture languages $\mathsf{picture}(L)$ that are defined by a word language L we call *iterative.*

A W-valued word language $L : A_k^* \to W$ is *recognizable* if there is a W-weighted automaton (WFA) $\mathcal{A}$ such that L is the behavior of $\mathcal{A}$. If the semiring W is locally finite (i.e. every finitely generated subsemiring of W is finite) then for every W-recognizable word language $L : A_k^2 \to W$, all pictures in $\mathsf{picture}(L)$ have colors from a finite subset of W.

Recognizability of picture languages over a finite set of colors is defined by tiling systems [5] and coincides with recognizability of picture languages by several other computational devices (nondeterministic 4-way-automata, on-line tessellation automata) and definability in existential monadic second order logic.

Our main result is the following theorem:

Theorem 1. *For every locally finite semiring W and every W-recognizable word language L : $A_k^2 \to W$, the picture language* $\mathsf{picture}(L)$ *is recognizable.*

This is proven by a connection to two-dimensional Lindenmayer systems [8], for which recognizability of the generated picture languages was shown in [9].

Using a result in [7], we present a non-recognizable word language L such that the picture language $\mathsf{picture}(L)$ is recognizable. Hence the recognizability of $\mathsf{picture}(L)$ does not imply the recognizability of the word language L.

The RGB model [3] is a standard color format for digital images. Every color is represented by a triple (r, g, b) (intensities of colors red, green, blue). The set of all colors in the RGB model forms a locally finite MV-algebra [2] with operations defined in [10]. Hence by [4] this algebra has semiring a reduct (with the operations of pointwise maximum and truncated addition) that can serve as weight semiring for weighted automata. By our theorem, the set of all pictures generated by a WFA-encoding [11] of an RGB image is a recognizable picture language.

References

1. Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs on Theoretical Computer Science*. Springer Verlag, 1988.
2. Roberto L. O. Cignoli, Itala M. Loffredo D'Ottaviano, and Daniele Mundici. *Algebraic Foundations of Many-Valued Reasoning*, volume 7 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, November 1999.
3. Mike Cowlishaw. Fundamental requirements for picture presentation. Technical report, Society for Information Display, 1985.
4. Antonio Di Nola and Brunella Gerla. Algebras of Łukasiewicz logic and their semiring reducts. *Contemp. Math. (AMS)*, 377:131–144, 2005.
5. Dora Giammaresi and Antonio Restivo. Two-dimensional languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 215–268. Springer-Verlag, 1997.
6. Helmut Jürgensen, Ludwig Staiger, and Hideki Yamasaki. Finite automata encoding geometric figures. *Theor. Comput. Sci.*, 381(1-3):33–43, 2007.
7. Jarkko Kari and Cristopher Moore. Rectangles and squares recognized by two-dimensional automata. In Juhani Karhumäki, Hermann A. Maurer, Gheorghe Paun, and Grzegorz Rozenberg, editors, *Theory Is Forever*, volume 3113 of *Lecture Notes in Computer Science*, pages 134–144. Springer, 2004.
8. A. Lindenmayer. Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology*, 18:180–315, 1968.
9. Kristian Lindgren, Cristopher Moore, and Mats Nordahl. Complexity of two-dimensional patterns. Working Papers 97-03-023, Santa Fe Institute, March 1997. available at http://ideas.repec.org/p/wop/safiwp/97-03-023.html.
10. Dan Noje and Barnabás Bede. The MV-algebra structure of RGB model. *Studia Universitatis Babes-Bolyai Informatica*, XLVI 1:77–86, 2001.
11. Karel Čulik II and Jarkko Kari. Digital images and formal languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 599–616. Springer-Verlag, 1997.

Decomposition of Weighted Multioperator Tree Automata

Torsten Stüber[1,⋆], Heiko Vogler[1], Zoltán Fülöp[2]

[1]Department of Computer Science, Technische Universität Dresden
D-01062 Dresden, Germany
{stueber,vogler}@tcs.inf.tu-dresden.de

[2]Department of Foundations of Computer Science, University of Szeged
Árpád tér 2., H-6720 Szeged, Hungary
fulop@inf.u-szeged.hu

Weighted multioperator tree automata (for short: wmta) were introduced in [16]; they are finite-state bottom-up weighted tree automata in which the transition weights are finite sums of polynomials over variables, operations, and constants. The operations are taken from a multioperator monoid (for short: M-monoid) [15, 16], which is an algebraic structure $(A, +, 0, \Omega)$ such that $(A, +, 0)$ is a commutative monoid and (A, Ω) is an Ω-algebra. If the operations in Ω distribute over $+$ and $+$ is idempotent, then an M-monoid is called distributive Ω-magma in [4].

Here we consider a simplified version of wmta (henceforth also called wmta) in which the transition weights are operations taken from Ω (rather than finite sums of polynomials over Ω). More precisely, given a wmta M, the weight of the transition at some k-ary input symbol σ with some state behaviour $(q_1 \cdots q_k, q)$ is a k-ary operation $\omega \in \Omega$; let us denote this operation by $\mu_k(\sigma)_{q_1\cdots q_k,q}$. Then, for every run r of M on some input tree $t \in T_\Sigma$ and every position w of t, the weight of r on t at w, denoted by $[\![r]\!]_{M,t}(w) \in A$, is obtained by applying the operation $\mu_k(\sigma)_{q_1\cdots q_k,q}$ to the k elements $[\![r]\!]_{M,t}(w.1), \ldots, [\![r]\!]_{M,t}(w.k) \in A$ where σ is the label of t at w and $(q_1 \cdots q_k, q)$ is the state behaviour at w prescribed by r. All in all, M recognizes the tree series $[\![M]\!] \in A\langle\langle T_\Sigma\rangle\rangle$ defined for every $t \in T_\Sigma$ such that $([\![M]\!], t)$ is the sum of the values $[\![r]\!]_{M,t}(\varepsilon)$ taken over all runs r on t. We denote the class of all wmta recognizable tree series over A by $BOT(A)$.

Wmta have been investigated in [18, 20] where it was shown that they can easily simulate weighted tree automata over semirings [1, 3, 14, 8, 5] and tree series transducers over semirings [17, 7, 11, 9, 19, 21] (for surveys on weighted tree automata and tree series transducers cf. [8, 12]). In [10] it was proved that the wmta recognizable tree series over some M-monoid A are exactly the rational tree series over A.

In this paper we prove three main results. The first main result is the following characterization of $BOT(A)$:

$$BOT(A) = REL; FTA; HOM(A)$$

where REL and FTA are the classes of relabeling tree transformations and fta tree transformations, respectively (as defined in [6]); an fta tree transformation is a partial identity on a recognizable tree language; $HOM(A)$ is the class of all tree series which are recognizable by homomorphism wmta over A, where a homomorphism wmta is a wmta with exactly one state which is also final; the semicolon in the right hand side expression denotes the usual composition of relations. This result generalizes the decomposition of generalized sequential machine mappings [22] (also cf. Theorem 4.1 of [2]) and of bottom-up tree transducers (cf. Theorem 3.5 of [6]).

The second main result of this paper is derived from the first one. It shows the following characterization of the class $p\text{-}BOT(S)$ of tree series transformations computed by polynomial bottom-up tree series transducers (for short: polynomial bottom-up tst) over some semiring S:

$$p\text{-}BOT(S) = REL; FTA; HOM(S)$$

where $HOM(S)$ is the class of tree series transformations computed by bottom-up homomorphism tst; a tree series transformation over S is a mapping $\varphi : T_\Sigma \to S\langle\langle T_\Delta\rangle\rangle$. Polynomial bottom-up tst were investigated in, e.g., [7, 21, 12]. We note that a similar characterization has been proved

⋆ The work of this author was partially supported by Deutsche Forschungsgemeinschaft, project DFG VO 1011/4-1.

in Theorem 5.7 of [7]: $p\text{-}BOT(S) = QREL(S); b\text{-}HOM(S)$, where $QREL(S)$ and $b\text{-}HOM(S)$ denote the classes of tree series transformations computed by bottom-up finite state relabeling tst and by Boolean bottom-up homomomorphism tst, respectively. The difference between these two characterizations is the fact that in $REL; FTA; HOM(S)$ the classes REL and FTA contain tree transformations, i.e., mappings of the type $T_\Sigma \to \mathcal{P}(T_\Delta)$, and the weights only occur in the third class (viz. $HOM(S)$); in contrast to this, in $QREL(S); b\text{-}HOM(S)$ the semiring values are solely computed by $QREL(S)$ and the Boolean-valued bottom-up homomomorphism tst only produces the values 0 or 1. This organization of weights in $QREL(S); b\text{-}HOM(S)$ forced the combination of the relabeling and the state checking which were originally separated. Also we note that, for the Boolean semiring, our second main result is exactly the characterization of BOT proved in Theorem 3.5 of [6].

The third main result of our paper is also derived from the first one, and it shows a characterization of the class $\mathrm{Rec}(\Sigma, S)$ of tree series which are recognizable by weighted tree automata over some semiring S:

$$\mathrm{Rec}(\Sigma, S) = PROJ(\Sigma, S)(\mathscr{L}_{LOC}),$$

where $\mathscr{L}_{LOC}$ is the class of local tree languages (cf. Section 8 of [13]), and $PROJ(\Sigma, S)$ is the class of tree series transformations which are computed by bottom-up projection tst. We note that, for the Boolean semiring, our third main result is exactly the well-known characterization of recognizable tree languages in terms of projections of local tree languages.

References

1. A. Alexandrakis and S. Bozapalidis. Weighted grammars and Kleene's theorem. *Inform. Process. Lett.*, 24(1):1–4, 1987.
2. J. Berstel. *Transductions and Context-Free Languages.* Teubner Stuttgart, 1979.
3. S. Bozapalidis. Equational elements in additive algebras. *Theory Comput. Syst.*, 32(1):1–33, 1999.
4. B. Courcelle. Equivalences and transformations of regular systems - applications to recursive program schemes and grammars. *Theoret. Comput. Sci.*, 42:1–122, 1986.
5. M. Droste, Chr. Pech, and H. Vogler. A Kleene theorem for weighted tree automata. *Theory Comput. Syst.*, 38:1–38, 2005.
6. J. Engelfriet. Bottom-up and top-down tree transformations - a comparison. *Math. Systems Theory*, 9(3):198–231, 1975.
7. J. Engelfriet, Z. Fülöp, and H. Vogler. Bottom-up and top-down tree series transformations. *J. Autom. Lang. Comb.*, 7:11–70, 2002.
8. Z. Ésik and W. Kuich. Formal tree series. *J. Autom. Lang. Comb.*, 8(2):219–285, 2003.
9. Z. Fülöp, Z. Gazdag, and H. Vogler. Hierarchies of tree series transformations. *Theoret. Comput. Sci.*, 314:387–429, 2004.
10. Z. Fülöp, A. Maletti, and H. Vogler. A Kleene theorem for weighted tree automata over distributive multioperator monoids. *Theory Comput. Syst.*, 2007. accepted for publication.
11. Z. Fülöp and H. Vogler. Tree series transformations that respect copying. *Theory Comput. Syst.*, 36(3):247–293, 2003.
12. Z. Fülöp and H. Vogler. Weighted tree automata and tree series transducers. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata.* Springer-Verlag, 2008. to appear.
13. F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, chapter 1, pages 1–68. Springer-Verlag, 1997.
14. W. Kuich. Semirings and formal power series: Their relevance to formal languages and automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Vol. 1*, chapter 9, pages 609–677. Springer-Verlag, 1997.
15. W. Kuich. Formal power series over trees. In S. Bozapalidis, editor, *3rd International Conference on Developments in Language Theory, DLT 1997, Thessaloniki, Greece, Proceedings*, pages 61–101. Aristotle University of Thessaloniki, 1998.
16. W. Kuich. Linear systems of equations and automata on distributive multioperator monoids. In *Contributions to General Algebra 12 - Proceedings of the 58th Workshop on General Algebra "58. Arbeitstagung Allgemeine Algebra", Vienna University of Technology. June 3-6, 1999*, pages 1–10. Verlag Johannes Heyn, 1999.

17. W. Kuich. Tree transducers and formal tree series. *Acta Cybernet.*, 14:135–149, 1999.
18. A. Maletti. Relating tree series transducers and weighted tree automata. In C.S. Calude, editor, *DLT'04 – 8th International Conference on Developments in Language Theory, Auckland, New Zealand, December 13-17, 2004*, volume 3340 of *Lecture Notes in Comput. Sci.*, pages 321–333, 2004.
19. A. Maletti. Hasse diagrams for classes of deterministic bottom-up tree-to-tree-series transformations. *Theoret. Comput. Sci.*, 339:200–240, 2005.
20. A. Maletti. Relating tree series transducers and weighted tree automata. *Internat. J. Found. Comput. Sci.*, 16(4):723–741, 2005.
21. A. Maletti. Compositions of tree series transformations. *Theoret. Comput. Sci.*, 366:248–271, 2006.
22. M. Nivat. Transduction des langages de Chomsky. *Ann. de l'Inst. Fourier*, 18:339–456, 1968.