# Weighted Automata — Theory and Applications

## Dresden, Germany, June 1–5, 2004

edited by Manfred Droste and Heiko Vogler

## Preface

This technical report contains the programme and the abstracts of the lectures delivered at the workshop "Weighted Automata: Theory and Applications" which took place at Dresden University of Technology from June 1–5, 2004. The workshop covered the field of weighted automata, ranging from the theory of formal power series to max-plus-algebra and applications as e.g. in image processing. In total, 52 participants from 14 countries attended the workshop.

Four tutorials were given by

| | |
|---|---|
| P. Butkovič (Birmingham), | J. Kari (Iowa), |
| F. Rossi (Padova), | J. Sakarovitch (Paris). |

In addition, seven invited lectures were presented by

| | |
|---|---|
| J. Albert (Würzburg), | S. Bozapalidis (Thessaloniki), |
| Z. Ésik (Szeged), | F. Katritzke (Siegen), |
| D. Kirsten (Dresden), | W. Kuich (Wien), |
| L. Staiger (Halle). | |

Furthermore, ten talks were selected as contributed communications.

The workshop was organized jointly by the Ph.D. programme (Graduiertenkolleg) "Specification of discrete processes and systems of processes by operational models and logics" (GK 334) and the Institute of Algebra. It was financially supported by the *Herbert Quandt Foundation* and the "Freunde und Förderer der TU Dresden".

## Call for Papers

The Journal of Automata, Languages, and Combinatorics (JALC) has agreed to publish a special issue on this topic for which the two organizers of the workshop will act as editors. Submissions related to this topic could be either survey articles or research papers and will be refereed according to the usual standards. Participation in the above workshop is encouraged, but is not a prerequisite for a submission.

Authors are asked to submit preferably a PostScript file (or else manuscripts in triplicate) to one of the editors of this special issue. Please send your files to

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

Deadline for submissions is **July 15, 2004**. We intend to ensure a quick refereeing process.

Manfred Droste and Heiko Vogler

# Part I

# Programme

## Tuesday, June 1, 2004

| | | |
|---|---|---|
| 08:30–09:00 | | REGISTRATION |
| 09:00–09:45 | J. Sakarovitch<br>*Four lessons in the theory of weighted automata and transducers (I)* | TUTORIAL |
| 09:45–10:30 | J. Sakarovitch<br>*Four lessons in the theory of weighted automata and transducers (II)* | TUTORIAL |
| 10:30–11:00 | | BREAK |
| 11:00–11:45 | P. Butkovič<br>*Max-algebra and its role in combinatorial optimisation (I)* | TUTORIAL |
| 11:45–12:30 | P. Butkovič<br>*Max-algebra and its role in combinatorial optimisation (II)* | TUTORIAL |
| 12:30–14:00 | | LUNCH |
| 14:00–15:00 | Z. Ésik<br>*An axiomatic basis for weighted automata* | INVITED LECTURE |
| 15:00–16:00 | W. Kuich<br>*An algebraic generalization of Omega-languages* | INVITED LECTURE |
| 16:00–16:40 | | BREAK |
| 16:40–17:05 | K. Zimmermann<br>*Optimization problems with mixed* $(\bigoplus, \bigotimes)$*-linear constraints* | TECHNICAL CONTRIBUTION |
| 17:05–17:30 | S. Lombardy<br>*Rational series with multiplicities in* $(\min, +)$ *and* $(\max, +)$ *semirings and unambiguity* | TECHNICAL CONTRIBUTION |
| 17:30–17:45 | | BREAK |
| 17:45–18:00 | D. Kirsten, *I. Mäurer*<br>*On subsequential, path-neutral, and unambiguous weighted automata* | HOME-SPEAKER |
| 18:00–18:15 | M. Droste, *U. Püschmann*<br>*Weighted Büchi automata* | HOME-SPEAKER |

## Wednesday, June 2, 2004

| | | |
|---|---|---|
| 08:30–09:00 | | REGISTRATION |
| 09:00–09:45 | J. Sakarovitch<br>*Four lessons in the theory of weighted automata and transducers (III)* | TUTORIAL |
| 09:45–10:30 | J. Sakarovitch<br>*Four lessons in the theory of weighted automata and transducers (IV)* | TUTORIAL |
| 10:30–11:00 | | BREAK |
| 11:00–11:45 | P. Butkovič<br>*Max-algebra and its role in combinatorial optimisation (III)* | TUTORIAL |
| 11:45–12:30 | P. Butkovič<br>*Max-algebra and its role in combinatorial optimisation (IV)* | TUTORIAL |
| 12:30–14:00 | | LUNCH |
| 14:00–15:00 | D. Kirsten<br>*Desert automata and the star height problem* | INVITED LECTURE |
| 15:00–16:00 | S. Bozapalidis<br>*Recognizable pattern series* | INVITED LECTURE |
| 16:00–16:40 | | BREAK |
| 16:40–17:05 | S. Bozapalidis, *G. Rahonis*<br>*On the closure of recognizable tree series under tree homomorphisms* | TECHNICAL CONTRIBUTION |
| 17:05–17:30 | B. Borchardt, *Z. Fülöp*<br>Zs. Gazdag, A. Maletti<br>*Bounds for tree automata with polynomial costs* | TECHNICAL CONTRIBUTION |
| 17:30–17:45 | | BREAK |
| 17:45–18:00 | A. Maletti<br>*Relating tree transducers and weighted tree automata* | HOME-SPEAKER |
| 18:00–18:15 | B. Borchardt<br>*The theory of recognizable tree series* | HOME-SPEAKER |

## Thursday, June 3, 2004

| | | |
|---|---|---|
| 08:30–09:00 | | Registration |
| 09:00–09:45 | S. Bistarelli, U. Montanari, *F. Rossi*<br>*Semiring-based constraint logic programming and concurrent constraint programming (I)* | Tutorial |
| 09:45–10:30 | S. Bistarelli, U. Montanari, *F. Rossi*<br>*Semiring-based constraint logic programming and concurrent constraint programming (II)* | Tutorial |
| 10:30–11:00 | | Break |
| 11:00–11:45 | J. J. Kari<br>*Image compression and image transformations using weighted finite automata (I)* | Tutorial |
| 11:45–12:30 | J. J. Kari<br>*Image compression and image transformations using weighted finite automata (II)* | Tutorial |
| 12:30–14:00 | | Lunch |
| 14:00–14:15 | *D. Kuske*, R. Morin<br>*Weighted asynchronous cellular automata* | Home-Speaker |
| 14:15–14:30 | I. Meinecke<br>*Weighted branching automata — Combining parallelism and weights* | Home-Speaker |
| 14:30–15:50 | | Break |
| 15:50–18:00 | | Excursion |

## Friday, June 4, 2004

| | | | |
|---|---|---|---|
| 08:30–09:00 | | | REGISTRATION |
| 09:00–09:45 | S. Bistarelli, U. Montanari, *F. Rossi*<br>*Semiring-based constraint logic programming and concurrent constraint programming (III)* | | TUTORIAL |
| 09:45–10:30 | S. Bistarelli, U. Montanari, *F. Rossi*<br>*Semiring-based constraint logic programming and concurrent constraint programming (IV)* | | TUTORIAL |
| 10:30–11:00 | | | BREAK |
| 11:00–11:45 | J. J. Kari<br>*Image compression and image transformations using weighted finite automata (III)* | | TUTORIAL |
| 11:45–12:30 | J. J. Kari<br>*Image compression and image transformations using weighted finite automata (IV)* | | TUTORIAL |
| 12:30–14:00 | | | LUNCH |
| 14:00–15:00 | J. Albert<br>*Generalizations of weighted finite automata and applications for images, videos and animations* | | INVITED LECTURE |
| 15:00–16:00 | *F. Katritzke*, W. Merzenich, M. Thomas<br>*Domain pool administration for image compression using weighted finite automata* | | INVITED LECTURE |
| 16:00–16:40 | | | BREAK |
| 16:40–17:05 | G. Tischler<br>*Properties and applications of parametric weighted finite automata* | | TECHNICAL CONTRIBUTION |
| 17:05–17:30 | J.-M. Champarnaud,<br>*F. Guingne*, G. Hansel<br>*Cover $\mathcal{K}$-automata* | | TECHNICAL CONTRIBUTION |
| 17:30–17:45 | | | BREAK |
| 17:45–18:00 | C. Pech<br>*Formal series on infinite trees* | | HOME-SPEAKER |
| 18:00–18:15 | *S. Bader*, A. Scalzitti<br>*Semiring artificial neural networks and weighted automata* | | HOME-SPEAKER |
| 18:15–20:00 | | | BREAK |
| 20:00– | | | CONFERENCE DINER |

## Saturday, June 5, 2004

| | | |
|---|---|---|
| 08:30–09:00 | | Registration |
| 09:00–10:00 | L. Staiger<br>*Fractals, iterated function systems and weighted automata* | Invited Lecture |
| 10:00–10:10 | | Break |
| 10:10–10:35 | A. Yli-Jyrä<br>*Simplification of intermediate results during intersection of multiple weighted automata* | Technical Contribution |
| 10:35–11:00 | *M. Giraud*, D. Lavenier<br>*Dealing with size limits in a hardware encoding of weighted finite automata* | Technical Contribution |
| 11:00–11:40 | | Break |
| 11:40–12:05 | A, Kempe, F. Guingne, *F. Nicart*<br>*New algorithms for weighted multi-tape automata* | Technical Contribution |
| 12:05–12:30 | A. Zollmann<br>*A consistent and efficient estimator for data-oriented parsing* | Technical Contribution |
| 12:05–14:00 | | Lunch |

# Part II

# Abstracts

# Tutorials

# Max-Algebra and its Role in Combinatorial Optimisation

Peter Butkovič

School of Mathematics and Statistics
The University of Birmingham
Edgbaston, Birmingham B15 2TT
United Kingdom

p.butkovic@bham.ac.uk

Let

$$a \oplus b = \max(a, b)$$

and

$$a \otimes b = a + b$$

for $a, b \in \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$. By max-algebra we understand the analogue of linear algebra developed for the pair of operations $(\oplus, \otimes)$ extended to matrices and vectors formally in the same way as in linear algebra, that is if $A = (a_{ij})$, $B = (b_{ij})$ and $C = (c_{ij})$ are matrices with elements from $\overline{\mathbb{R}}$ of compatible sizes, we write $C = A \oplus B$ if $c_{ij} = a_{ij} \oplus b_{ij}$ for all $i, j$, $C = A \otimes B$ if $c_{ij} = \sum_k^{\oplus} a_{ik} \otimes b_{kj} = \max_k(a_{ik} + b_{kj})$ for all $i, j$ and $\alpha \otimes A = (\alpha \otimes a_{ij})$ for $\alpha \in \overline{\mathbb{R}}$.

The attractivity of max-algebra is related to the facts that $(\overline{\mathbb{R}}, \oplus, \otimes)$ is a commutative semiring and $(\overline{\mathbb{R}}^n, \oplus, \otimes)$ is a semimodule. Hence many of the basic tools used in linear algebra are available in max-algebra as well.

Max-algebra enables us to describe and study a class of nonlinear problems appearing for instance in machine-scheduling, information technology and discrete-event dynamic systems, by applying the linear-algebraic approach. We will present a selection of these applications, including one in digital signal processing.

We will also present an overview of results which demonstrate strong links between max-algebraic problems and combinatorial or combinatorial optimisation problems. These links are so significant that max-algebra may be regarded as a linear-algebraic encoding of a class of combinatorial problems. Instances of such combinatorial problems are: the set covering (which in max-algebra is the solvability of a linear system), the minimal set covering (unique solvability of a linear system), existence of a directed cycle (strong regularity of a matrix), existence of an even directed cycle (regularity of a matrix), maximal cycle mean (eigenvalue), longest-distances (eigenvectors), best principal submatrices (coefficients of a characteristic polynomial). Some results are related to matrix scaling and enable us to formulate a link between combinatorial problems so different as the assignment problem and the longest-distances problem.

It should be noted that if $\oplus$ is set to the minimum rather than to the maximum then we speak about min-algebra in which all the results remain valid after appropriately amending the formulation, for instance after replacing $\leq$ with $\geq$, "positive" with "negative", "longest" with "shortest" etc.

It is beyond the scope of this abstract to provide a comprehensive survey of all major existing results and works. But the following may help researchers who would like to know more about max-algebra: Although first papers related to max-algebra appeared as soon as in the 1950's or even before, the first comprehensive monograph seems to be [4] which was later extended and updated [5]. Algebraic aspects are comprehensively studied in [9]. Other recommended sources are: monograph [1] with many applications and various different approaches, paper [8] and thesis [6]. A summer school's proceedings [7] is a collection of a number of the state-of-the-art papers including a thorough historical overview of the field. A chapter on max-algebra can be found in [2]. A substantial part of our presentation is included in [3].

## References


[1] F. L. Baccelli, G. Cohen, G.-J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems.* Wiley, Chichester, New York, 1992. available online at: `http://www-rocq.inria.fr/metalau/cohen/SED/book-online.html`.

[2] R. B. Bapat and T. E. S. Raghavan. *Nonnegative Matrices and Applications*, volume 64 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1997.

[3] P. Butkovič. Max-algebra: The linear algebra of combinatorics? *Linear Algebra and its Applications*, 367:313–335, 2003.

[4] R. A. Cuninghame-Green. *Minimax Algebra*, volume 166 of *Lecture Notes in Economics and Mathematical Systems.* Springer, 1979.

[5] R. A. Cuninghame-Green. Minimax algebra and applications. In *Advances in Imaging and Electron Physics*, volume 90, pages 1–121. Academic Press, New York, 1995.

[6] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes (Theory of Linear Systems over Dioïds).* PhD thesis, École Nationale Supérieure des Mines de Paris, 1992.

[7] S. Gaubert et al. Algèbres max-plus et applications en informatique et automatique. In *26ème École de Printemps d'Informatique Théorique.* Noirmoutier, 1998.

[8] M. Gondran and M. Minoux. Linear algebra in dioïds: A survey of recent results. *Annals of Discrete Mathematics*, 19:147–164, 1984.

[9] U. Zimmermann. *Linear and Combinatorial Optimization in Ordered Algebraic Structures*, volume 10 of *Annals of Discrete Mathematics.* North Holland, Amsterdam, 1981.

# Semiring-based Constraint Logic Programming and Concurrent Constraint Programming

Stefano Bistarelli[1], Ugo Montanari[2], Francesca Rossi[3]

[1] Dipartimento di Scienze, Università degli Studi G. d'Annunzio, Viale Pindaro 87, I-65127 Pescara, Italy
[2] Dipartimento di Informatica, Università di Pisa, Via F. Buonarroti 2, I-56127 Pisa, Italy
[3] Dipartimento di Matematica Pura ed Applicata, Università degli Studi di Padova, Via Belzoni 7, I-35131 Padova, Italy

[1] `bista@sci.unich.it` [2] `ugo@di.unipi.it` [3] `frossi@math.unipd.it`

Soft constraints extend classical constraints to represent multiple consistency levels, and thus provide a way to express preferences, fuzziness, and uncertainty.

In this talk we present Soft Constraint Logic Programming, which is an extension of the Constraint Logic Programming (CLP) formalism to handle soft constraints. This allows one to perform in the same language both constraint solving and optimization. In fact, soft constraints are able to model both classical constraint solving and more sophisticated features like uncertainty, probability, fuzziness, and optimization. We then provide this class of languages with three equivalent semantics: model-theoretic, fix-point, and proof-theoretic, in the style of classical CLP programs.

Moreover, we also show how classical concurrent constraint (cc) programming can work with soft constraints, and we propose an extension of cc languages which can use soft constraints to prune and direct the search for a solution. We believe that this new programming paradigm, called soft cc (scc), can be also very useful in many web-related scenarios.

# Image Compression and Image Transformations Using Weighted Finite Automata

Jarkko Kari

Department of Computer Science
14 MLH University of Iowa
Iowa City, IA 52242
USA

jjkari@cs.uiowa.edu

Weighted finite automata (WFA) compute real functions over words of the input alphabet. When the words of the alphabet are interpreted in various ways, we obtain real functions over other domains. In digital imaging, we use a four letter alphabet $\{0, 1, 2, 3\}$, and interpret each word of length $k$ as the address of a unique cell in a partitioning of the unit square $[0, 1]^2 \subseteq \mathbb{R}^2$ into $2^k \times 2^k$ sub-squares. The value computed by the WFA is then taken as the gray-scale intensity of the cell, or the pixel value of the $2^k \times 2^k$ image that the WFA defines. We start by discussing this definition and by giving several examples. The images defined by WFA are self-similar: all states of the automaton specify images such that the quadrants of the images can be expressed as linear combinations of the same collection of images, where the coefficients of the linear combinations are the weights of the transitions in the weighted automaton.

It turns out that WFA provide compact image representations of smooth and fractal images. This leads to the natural idea of using WFA in image compression. In compression one needs efficient compression and decompression algorithms. Decompression is fairly easy and can be done very fast. It is harder to do the compression, that is, to infer a WFA that represents a given input image. In this talk we start with a theoretical algorithm that losslessly represents a given image using a minimum number of states. Then we develop a practical, lossy, image compression algorithm where the algorithm makes local compression decisions using a Lagrangian multiplier to optimally balance the trade-off between the compression rate and the image quality. We show compression results and examples, and compare the results with other known compression techniques.

Weighted finite transducers (WFT) compute real functions over pairs (or more generally, $n$-tuples) of words, so they define a weighted relation between words. Using the interpretation above, any WFT over alphabet $\{0, 1, 2, 3\}$ defines an image transformation. We show that several useful image operations can be implemented in terms of WFT including zooming, cropping, FIR filtering, etc. The implementation of image operations using WFT has the advantage that the operation can be applied to any WFA represented image directly in the WFA form, without decoding the image first. This saves processing time and space greatly, if the automaton and the transducer have a small number of states. Also, the result of the operation is precise and it can be correctly obtained at arbitrarily high resolutions.

Finally, we define parametric WFA, or PWFA, that compute functions from the words of the input alphabet into higher dimensional real spaces $\mathbb{R}^d$. In this case, we no longer need to interpret the input words as pixel addresses. Rather, we are interested in the range of the function. This is a subset of $\mathbb{R}^d$. The role of the input word is to be a parameter that ties the coordinates of the points together. We show how interesting subsets of $\mathbb{R}^2$ can be specified using PWFA with a small number of states. Examples include circles and the graphs of the sine and the exponential function. Also any Iterated Function System (IFS) can be simulated by PWFA.

# Four Lessons in the Theory of Weighted Automata and Transducers

Jacques Sakarovitch

Laboratoire Traitement et Communication de l'Information
CNRS / Télécom-Paris
46, rue Barrault, F-75634 Paris Cedex 13
France

sakarovitch@enst.fr

In these lectures, I will survey three aspects of the problem of realisability of systems, namely the questions of reduction, synchronisation, and sequentialisation, the solution of which is at the theoretical foundation of many applications of finite automata.

1. BASICS. I'll introduce, together with automata with multiplicity in $\mathbb{K}$, the notion of $\mathbb{K}$-covering which plays a role similar to morphisms for classical (Boolean) automata.

2. REDUCTION. I'll present an algorithm for the reduction of an automaton with multiplicity in a (skew) field, that is for finding an equivalent automaton with a minimal number of states.

3. SYNCHRONISATION of transducers may be seen as a particular case of a Fatou problem — that is the realisability of a weighted automaton using only a subsemiring of weights, induced by the behaviour of the automaton.

4. SEQUENTIALISATION is the problem of the realisation of a weighted automaton by a deterministic process. Whether it is decidable if the behaviour of a weighted automaton may be realised by a sequential one is a question whose solution depends on the semiring of weights. We shall consider the main classical cases, some being decidable, some undecidable, and some still open.

# Invited Lectures

# Generalizations of Weighted Finite Automata and Applications for Images, Videos and Animations

Jürgen Albert

Department of Computer Science
University of Würzburg
Am Hubland, D-97074 Würzburg
Germany

albert@informatik.uni-wuerzburg.de

Initially, nondeterministic finite automata with states and transitions labeled by real numbers had been studied by K. Culik II, J. Karhumäki, J. Kari et al. as finite state generators for smooth and fractal-like real functions over the unit interval. These Weighted Finite Automata (WFA) soon turned out to be very compact and efficient representations for the polynomials and many more interesting functions like e.g. specific wavelets. Furthermore, the construction-principles were easily extended to higher dimensions and WFA could generate functions over the unit square, the unit cube etc. It is just a matter of choosing the input alphabet and the interpretation of the input sequences of the WFA to decide about the dimension. This provides the basis to employ WFA for image-compression of grey-scale images, where the greyness value of a pixel is the function value of the WFA at that particular image address or an approximation hereof. In 1993 K. Culik II and J. Kari published an elegant recursive inference algorithm, which, given an image and a quality-parameter, produces a small WFA representing an approximation of that image. During the recursion steps internally a cost-function controls the coding decisions between spending more WFA states and transitions with weights versus allowing a higher local deviation of the pixel values.

These local decisions are biased by the global quality-parameter to achieve high compression rates or (alternatively) high reconstruction quality of the image. The next step to handle color images is easy since each color layer (red, green, blue (RGB)) can be treated separately as a grey-scale image. The compression results are improved drastically by applying a color transform first, (RGB) to (YUV) e.g., and further by predicting image regions from previous layers. This leads to a compression performance which is in general better than the well-known JPEG Baseline compression, especially for high compression rates ("low bitrate-coding"). The general principle of WFA, exploiting self-similarities, can be carried over easily from using several parts of a single grey-scale picture, to layers of a color image and quite naturally to sequences of pictures as found in videos or in 3D medical imaging.

Thus, as a coding-method WFA has much in common with fractal encoding, especially with M. Barnsley's Iterated Function Systems. Therefore, it is an intriguing question to specify a common framework for both approaches, which are incomparable in their generative power. This is one of the goals of Parametric Weighted Finite Automata, PWFA for short, which do define relations of addresses and "grey-scale intensities" instead of functions from addresses to intensities. Trivially, PWFA can produce any polynomial curves in any d-dimensional unit cube and they are strictly more powerful than WFA.

They also simulate Iterated Function Systems easily and with a small number of states. Amazingly simple representations exist for circle, sine- and exponential functions. To obtain an efficient inference algorithm for PWFA is part of current research.

# Recognizable Pattern Series

Symeon Bozapalidis

Department of Mathematics
Aristotle University of Thessaloniki
GR-54124 Thessaloniki
Greece

bozapali@math.auth.gr

The magmoids of Arnold and Dauchet seem to be the more convenient algebraic structure to study flow graphs (i.e. graphs with input and output). A magmoid is a doubly ranked set $M = (M_{mn})$ endowed with two operations

$$\circ : M_{mn} \times M_{nk} \to M_{mk}, \qquad \square : M_{mn} \times M_{m'n'} \to M_{m+m',n+n'}$$

which satisfy certain natural axioms.

Here we deal with series $S : mag(X) \to K$ over the free magmoid $mag(X)$ generated by the finite doubly ranked alphabet $X = (X_{mn})$ and having coefficients in a field $K$. The elements of $mag(X)$ are constructed by the symbols of $X$ by horizontal and vertical concatenation and are called *patterns* (from $X$).

A pattern series $S : mag(X) \to K$ is said to be *recognizable* whenever it can be represented by a locally finite dimensional magmoid. The set $REC(X, K)$ of such series is closed under sum, scalar product, Hadamard product, inverse magmoid morphisms and direct alphabetic morphisms.

The *syntactic representation* $M_S$ of a series $S : mag(X) \to K$ is constructed through (right) derivatives and it is shown that $S$ is recognizable iff $dim M_S < \infty$. The pattern size is a recognizable series whereas pattern height fails to be.

A pattern automaton $\mathcal{M}$ over the doubly ranked input alphabet $X = (X_{mn})$ consists of a finite state set $Q$, a (finite) transition set $E$ of triples $(u, x, v)$ with $u \in Q^m$, $x \in X_{mn}$, $v \in Q^n$ and two rational languages $I, T \subseteq Q*$ (*initial* and *terminal face languages*). The behavior of $\mathcal{M}$ is obtained by projecting the set $loc(\mathcal{M}) \subseteq mag(E)$ of all "composable" patterns whose faces belong to $I \times T$ into $mag(X)$. The important fact here is that the strong behavior of $\mathcal{M}$, that is the function sending each pattern of $mag(X)$ to the number of its distinct computations, is a recognizable pattern series.

Applications to 0-type grammars which constitute a special case of pattern automata can be obtained. Finally pattern automata weighted over a field $K$ are investigated.

# An Axiomatic Basis for Weighted Automata

Zoltán Ésik

Institute of Informatics
University of Szeged
Árpád tér 2, H-6720 Szeged
Hungary

ze@inf.u-szeged.hu

Conway semirings, iteration semirings and Conway and iteration semiring-semimodule pairs were introduced by Bloom and Ésik about 15 years ago with the aim of providing an axiomatic treatment of (weighted) finite automata, regular languages and regular omega-languages. In this talk we will describe several new classes of these structures and discuss some of their generalizations. The most recent results of the talk were obtained in collaboration with Werner Kuich.

# Domain Pool Administration for Image Compression using Weighted Finite Automata

Frank Katritzke[1], Wolfgang Merzenich[1], Michael Thomas[2]

[1] Department of Electrical Engineering and Computer Science, Universität Siegen, Postfach, D-57068 Siegen, Germany

[2] Department of Mathematics, Universität Siegen, Postfach, D-57068 Siegen, Germany

[1] {frankka,merze}@informatik.uni-siegen.de [2] michael@stat.math.uni-siegen.de

Weighted Finite Automata (WFAs) are data structures which may be used for the efficient storage of digital images. An important factor concerning compression efficiency is the administration of the domain pool that is used for fractal coding. In this paper we examine whether the domain pool administration should be made coherent for or independent of the WFA tree levels. The quality of the two techniques is compared by determining the compression performance of well-known test images.

# Desert Automata and the Star Height Problem


Daniel Kirsten[†]

Institute of Algebra
Dresden University of Technology
D-01062 Dresden
Germany

kirsten@math.tu-dresden.de
Personal webpage: http://www.math.tu-dresden.de/~kirsten/


In the talk, we introduce desert automata. Desert automata are non-deterministic finite automata with a set of marked transitions which are called water transitions. The weight of a path is defined as the length of a longest subpath which does not contain a water transition. The weight of a word is the minimum of the weights of all successful paths of the word. In this way, desert automata compute mappings from a free monoid to the integers.

We also show some extensions of this concept which include Hashiguchi's distance automata. We consider the limitedness problems of these automata, i.e., we show that it is PSPACE-complete to decide whether the range of the mapping of a given desert automaton is finite. We sketch some applications of this result, e.g., we show the decidability of the finite substitution problem as well as a new solution and the first upper complexity bound to the star height problem.

## References


[1] D. Kirsten. Desert automata I: A Burnside problem and its solution. Technical Report 2003-020, Laboratoire d'Informatique Algorithmique: Fondements et Applications (LIAFA), Paris, 2003.

[2] D. Kirsten. Desert automata and the finite substitution problem. In V. Diekert and M. Habib, editors, *21st Annual Symposium on Theoretical Aspects of Computer Science, STACS 2004, Montpellier, France, March 25–27, 2004, Proceedings*, volume 2996 of *Lecture Notes in Computer Science*, pages 305–316. Springer, 2004.

[3] D. Kirsten. Desert automata III: Towards a comprehension of the star height problem. Technical Report 2004-001, Laboratoire d'Informatique Algorithmique: Fondements et Applications (LIAFA), Paris, 2004.

[4] D. Kirsten. Distance desert automata and the star height one problem. In I. Walukiewicz, editor, *7th International Conference on Foundations of Software Science and Computation Structures, FoSSaCS 2004, Barcelona, Spain, March 29 – April 2, 2004, Proceedings*, volume 2987 of *Lecture Notes in Computer Science*, pages 257–272. Springer, 2004. EATCS Award "Best Theoretical Paper presented at ETAPS 2004".



[†]Supported by the research grant KI 822/1–1 of the German Research Foundation (Deutsche Forschungsgemeinschaft).

# An Algebraic Generalization of Omega-Languages

Werner Kuich

Institut für Algebra und Computermathematik
Technische Universität Wien
Wiedner Hauptstraße 8–10, A-1040 Wien
Austria

w.kuich@tuwien.ac.at

Based on the semiring-semimodule pairs introduced in the lecture of Zoltán Ésik, omega languages are generalized. We discuss this generalization in the case of omega-regular and omega-context-free languages by use of matrices and algebraic systems, respectively.

# Fractals, Iterated Function Systems and Weighted Automata

Ludwig Staiger

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg
von-Seckendorff-Platz 1, D–06099 Halle
Germany

staiger@informatik.uni-halle.de

There is a series of approaches to describe fractals. One of them is via *Iterated Function Systems* (IFS). An IFS is composed of a metric space $(\mathcal{X}, \rho)$ and a set $\{\mathcal{F}(0), \ldots, \mathcal{F}(r-1)\}$ of mappings on $\mathcal{X}$. If all $\mathcal{F}(i)$ are contractive, the IFS $(\mathcal{X}, \{\mathcal{F}(0), \ldots, \mathcal{F}(r-1)\})$ defines a fractal as the smallest nonempty and (topologically) closed solution of the equation $K = \bigcup_{i=0}^{r-1} \mathcal{F}(i)(K)$. Similarly, $K$ can be defined in the following way:

1. For every mapping $\xi : \omega \to \{0, \ldots, r-1\}$ the sequence[2]

$$\Big( \big( \mathcal{F}(\xi(1)) \circ \cdots \circ \mathcal{F}(\xi(j)) \big)(a) \Big)_{j=1}^{\infty} \tag{1}$$

   converges independently of the starting point $a \in \mathcal{X}$ to a limit point $a_\xi \in K$ depending only on $\xi$.

2. Thus every mapping $\xi$ can be seen as an address of $\phi_{\mathcal{F}}(\xi) := a_\xi$.

3. The above-mentioned solution $K$ equals $\big\{ \phi_{\mathcal{F}}(\xi) : \xi \in \{0, \ldots, r-1\}^\omega \big\}$.

Mappings $\xi : \omega \to X$ where $X$ is a finite set (alphabet) are known in the theory of formal languages as $\omega$-words.

It also makes sense to consider IFS where not all mappings $\mathcal{F}(i)$ are contractive. Should this be the case, Item 1 from above is not fulfilled for all $\xi$. Then one has to single out those $\xi$ for which the sequence (1) converges independently of the starting point $a \in \mathcal{X}$. In practice, this condition is very intriguing and depends heavily on the mappings $\mathcal{F}(i)$.

In order to avoid this dependence one often confines to the consideration of the contraction coefficients $c_j$ of the mappings $\mathcal{F}(j)$. In most cases, this yields only sufficient conditions for the convergence of the sequence (1). Here, as usual, we denote the real number $c_i$ a *contraction coefficient* of the mapping $\mathcal{F}(i)$ provided

$$\rho(\mathcal{F}(i)(a), \mathcal{F}(i)(a')) \leq c_i \cdot \rho(a, a') \tag{2}$$

holds for all $a, a' \in \mathcal{X}$.

The advantage of confining to contraction coefficients is, on the one hand, a simpler analysis of the IFS and, on the other hand, it provides construction methods for controlling the IFS in order to obtain convergence of the sequence (1). This is done by attaching to the IFS a device $\mathfrak{D}_F$ controlling the sequences $\xi$, that is, replacing the set $K$ in Item 3 by

[2] Here $\omega$ denotes the set of positive (non-zero) natural numbers.

$\{\phi_{\mathcal{F}}(\xi) : \xi \in F\}$ for a suitably chosen subset $F \subseteq \{0, \ldots, r-1\}^\omega$. In the sequel, we shall consider such systems consisting of an IFS an a controlling device $\mathfrak{D}_F$.

A suitable means for both problems is the use of weighted automata over the classical semi-ring of nonnegative reals with usual addition and multiplication. Here in an automaton $\mathcal{A} = (X, (0, \infty), Z, z_0, f, g)$ with output monoid $((0, \infty), \cdot)$ to each transition the contraction coefficient of the mapping $\varphi_x$ corresponding to the input letter $x \in X$ is assigned. The resulting output $g(s, w)$ is an upper bound to the contraction coefficient of the mapping $\varphi_w := \varphi_{x_1} \circ \cdots \circ \varphi_{x_n}$ $(w = x_1 \cdots x_n)$.

In case the contraction coefficients of $\varphi_w$ for $w \to \xi \in X^\omega$ converge to zero our system "draws a point for $\xi \in X^\omega$". The set of all such $\xi$ can be described topologically by a suitable topology (depending on the automaton $\mathcal{A}$).

This yields as systems guaranteeing the convergence in (1) so-called graph-directed structures or Multiple Recursive Function Systems (these are combinations of finite automata with iterated function systems).

A further main problem encountered in fractal geometry is the determination of the Hausdorff dimension of fractals. Here in certain cases an exact calculation is possible using solely the contraction coefficients of the mappings and the properties of the control structure of our system.

# Technical Contributions

# Bounds for Tree Automata with Polynomial Costs

Björn Borchardt[1], Zoltán Fülöp[2], Zsolt Gazdag[2], Andreas Maletti[1]

[1] Department of Computer Science, Dresden University of Technology, D-01062 Dresden, Germany

[2] Department of Computer Science, University of Szeged, Árpád tér 2., H-6720 Szeged, Hungary

[1] `{borchard,maletti}@tcs.inf.tu-dresden.de` [2] `{fulop,gazdag}@inf.u-szeged.hu`

A tree automaton with cost function is a classical tree automaton $M = (Q, \Sigma, \delta, F)$ supplemented with a mapping $c : \delta \longrightarrow A\langle\{x_1, x_2, \ldots\}\rangle$ associating with each transition of $M$ a cost polynomial over a semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$, cf. [1]. In such a polynomial a variable $x_i$ refers to the cost of the $i$-th subcomputation, so the cost of a computation is obtained by substituting the cost of the subcomputations for the corresponding $x_i$'s in the cost of the top-most transition. A tree automaton with cost function is bounded (with respect to a partial order on $A$), if there exists a semiring element $a \in A$ such that the cost of each accepting computation (i.e., ending in a final state) is below $a$. We firstly observe that the problem of boundedness (with respect to the natural order) is equivalent to the cost-finiteness problem in finitely factorizing and naturally ordered semirings. A semiring $\mathcal{A}$ is finitely factorizing if there are only finitely many possibilities to decompose each element $a \in A$ into $a = a_1 \oplus a_2$ and $a = a_3 \odot a_4$ in a nontrivial way.

The cost-finiteness problem is to determine whether the set $c(M) \subseteq A$ of all costs generated by all accepting computations of $M$ is finite. We are interested in finding a relatively large class of semirings such that cost-finiteness is decidable for every tree automaton with cost function over a semiring belonging to that class. For that purpose, we introduce monotonic and finitely factorizing (mff, for short) semirings. Roughly speaking, a semiring $\mathcal{A}$ is monotonic if it is infinite and there is a partial order $\preceq$ over $A$ such that the operations $\oplus$ and $\odot$ are monotonic with respect to $\preceq$ and $\prec$, respectively. The class of mff semirings seems to be rather wide, because it contains some well-known semirings: the semiring of non-negative integers, the max-plus semiring, and the semiring of finite languages. We characterize the cost-finite tree automata with cost function over mff semirings by proving that cost-finiteness is equivalent to a decidable property imposed on the cost polynomials of the transitions. More precisely, we show that transitions in a strong component of the trace graph of $M$ might only be assigned an element of a certain class of polynomials, in which membership is also shown to be decidable. Hence we generalize results of [1].

## References

[1] H. Seidl. Finite tree automata with cost functions. *Theoretical Computer Science*, 126(1):113–142, 1994.

[1] These authors are financially supported by the German Research Foundation (DFG, GK 334/3).

[2] Research of these authors was supported by grants from the exchange program of DAAD and MÖB and the Herbert Quandt Foundation.

# On the Closure of Recognizable Tree Series under Tree Homomorphisms

Symeon Bozapalidis, George Rahonis

Department of Mathematics
Aristotle University of Thessaloniki
GR-54124 Thessaloniki
Greece

{bozapali,grahonis}@math.auth.gr

We deal with systems of equations whose right hand-side members are formal tree series, i.e.

$$x_i = S_i, \qquad S_i \in K\langle\langle T_\Gamma(X_n)\rangle\rangle, \qquad 1 \le i \le n. \tag{1}$$

Such a system admits various kinds of solutions according to the substitution operation we use to solve it. Two modes of this operation are defined. The first one is the well-known $OI$-, whereas the second is $IO$-substitution. Precisely for $t \in T_\Gamma(X_n)$ and $S_1, \dots, S_n \in K\langle\langle T_\Gamma(X_n)\rangle\rangle$

$$t[S_1, \dots, S_n]_{IO} = \sum_{t_1, \dots, t_n \in T_\Gamma(X_n)} (S_1, t_1) \dots (S_n, t_n)\, t[t_1, \dots, t_n].$$

If some $S_i$, $i = 1, \dots, n$ above equals to 0 the obtained series is 0 even if the variable $x_i$ does not appear in $t$. In order to surmount this technical situation we introduce a slight modification of this operation, named *[IO]*-substitution.

Let $x_{i_1}, \dots, x_{i_p}$, $1 \le i_1 < \dots < i_p \le n$ be all the variables that occur in $t \in T_\Gamma(X_n)$. Then

$$t[S_1, \dots, S_n]_{[IO]} = \sum_{t_{i_1}, \dots, t_{i_p} \in T_\Gamma(X_n)} (S_{i_1}, t_{i_1}) \dots (S_{i_p}, t_{i_p})\, t[t_{i_1}, \dots, t_{i_p}].$$

We prove that *[IO]*-substitution is preserved by non-deleting tree homomorphisms.

Our approach to obtain $OI$- and *[IO]*-solutions of (1) is by approximating sequences using Tarski's Theorem. This is exactly the motivation to use the *[IO]*- instead of the $IO$-substitution.

A tree series is termed *[IO]-equational* if it is a component of the least *[IO]*-solution of a system (1) above, whose right hand-side members are polynomials.

The main result of the paper states that the closure of the class of recognizable tree series under non-deleting tree homomorphisms is the class *[IO]-Equat* of *[IO]*-equational tree series. Further, we prove that the Hadamard product of a recognizable tree series with an *[IO]*-equational tree series is *[IO]*-equational.

Applications of our results to tree language theory are exposed. In fact, we achieve to determine an equational description of the closure of $REC$ (the class of recognizable tree languages) under arbitrary tree homomorphisms. It is just the class of *[IO]*-equational tree languages. Next, we show that the branches of an *[IO]*-equational language form a recognizable language.

# Cover $\mathcal{K}$-automata

Jean-Marc Champarnaud[1], Franck Guingne[1,2], Georges Hansel[1]

[1] Laboratoire d'Informatique Fondamentale et Appliquée (LIFAR), Universite de Rouen, F-76821 Mont-Saint-Aignan Cedex, France

[2] Xerox Research Center Europe (XRCE), F-38240 Meylan, France

jmc@dir.univ-rouen.fr franck.guingne@xrce.xerox.com georges.hansel@univ-rouen.fr

Our aim is to extend the notion of cover automata for finite languages to cover $\mathcal{K}$-automata for polynomials. A cover automaton for a finite language $L$ [1] of order $l$ (the length of a longest word in $L$) is a deterministic automaton $\mathcal{C}$ such that $L(\mathcal{C}) \cap \Sigma^{\leq l} = L$ ($\Sigma^{\leq l}$ is the subset of words of $\Sigma^*$ having a length not greater than $l$). Checking word membership over a cover automaton only requires an additional test on the length of the word. Since covering generally reduces the size of an automaton [6], it is of practical interest to be able to compute a minimal cover automaton (w.r.t. the number of states). In [4], Körner describes a Hopcroft-like algorithm to compute a minimal cover automaton from an $n$-state cover automaton with an $O(n \log n)$ time complexity.

We define the notion of a cover $\mathcal{K}$-automaton for a polynomial, as an extension of the notion of a cover automaton for a finite language. Given an alphabet $\Sigma$ and a semiring $\mathcal{K}$, a cover $\mathcal{K}$-automaton for a polynomial $P$ is any $\mathcal{K}$-automaton over $\Sigma$ which realizes the polynomial $P$ when its input is restricted to $\Sigma^{\leq l}$. Given a cover $\mathcal{K}$-automaton for a polynomial $P$ our aim is to computing a cover $\mathcal{K}$-automaton for $P$ with a reduced number of states.

We first discuss of the relative power of different state relations that can be used for reducing the size of the cover $\mathcal{K}$-automaton. We show that the scheme of the algorithm for minimizing a subsequential transducer [2] or a $\mathcal{K}$-automaton [5] can be adapted for computing a reduced cover $\mathcal{K}$-automaton from a given cover $\mathcal{K}$-automaton. We report experimental results obtained with our implementation of this technique using WFSC (Weighted Finite State Compiler), a Xerox tool [3].

We finally show that a cover $\mathcal{K}$-automaton can be reduced according to a relation whose underlying relation on $\Sigma^{\leq l}$ is not a similarity relation. Computing a minimal partition of the set of states according to this merging relation is therefore a more complex problem.

## References

[1] C. Câmpeanu, N. Sântean, and S. Yu. Minimal cover-automata for finite languages. *Theoretical Computer Science*, 267(1–2):3–16, 2001.

[2] C. Choffrut. Minimizing subsequential transducers: A survey. *Theoretical Computer Science*, 292(1):131–143, 2003.

[3] A. Kempe, C. Baeijs, T. Gaál, F. Guingne, and F. Nicart. WFSC — a new weighted finite state compiler. In Z. Dang O. H. Ibarra, editor, *8th International Conference*

*on Implementation and Application of Automata, CIAA 2003, Santa Barbara, California, USA, July 16–18, 2003, Proceedings*, volume 2759 of *Lecture Notes in Computer Science*, pages 108–119, Heidelberg, 2003. Springer.

[4] H. Körner. A time and space efficient algorithm for minimizing cover automata for finite languages. *International Journal of Foundations of Computer Science*, 14(6):1071–1086, 2003.

[5] M. Mohri. Minimization algorithms for sequential transducers. *Theoretical Computer Science*, 234(1–2):177–201, 2000.

[6] N. Sântean. Towards a minimal representation for finite languages: Theory and pratice. Master's thesis, The University of Western Ontario, Ontario, 2000.

# Dealing with Size Limits in a Hardware Encoding of Weighted Finite Automata

Mathieu Giraud[1], Dominique Lavenier[2]

IRISA / CNRS
Université de Rennes 1
F-35042 Rennes Cedex
France

[1] mathieu.giraud@irisa.fr [2] lavenier@irisa.fr

We use Weighted Finite Automata (WFA) [7] to parse protein or nucleic banks for finding specific patterns. The weights of WFA enable arbitrary error counts or substitution costs [5]. A WFA $\mathcal{A}$ over the semi-ring $(\mathbb{Z}, +, \max)$ assigns to every word $w$ a weight $P(w)$. With a threshold $s_0$, one can define the language recognized by $\mathcal{A}$ by $\mathcal{L} = \{w \mid P(w) \geq s_0\}$. Given a large word $w$ (the bank), we address the problem of *continuous pattern matching*: one must find all the terminating positions $j$ of matching subwords $w_i w_{i+1} \dots w_j \in \mathcal{L}$. We hardwire WFA in reconfigurable processors to accelerate this parsing.

**Linear Encoding Scheme.** Finite state machines with $q$ states can be encoded in hardware either by the *logarithmic scheme* (a bit vector of size $\log_2 q$ stores the current state) or by the *linear scheme* which uses a bit vector of size $q$. Usually only one bit is set (this scheme is also named *one-hot*), but one can have multiple states active at the same time and thus simulate indeterminism. The linear scheme has other advantages [3, 8], and we showed that it can be generalized to parse WFA [5]. In this case, each state is translated into one flip-flop, and each set of transitions between two states is translated into a weight generator, an adder and an optional maximum operator (Figures 1 and 3).

**Size and Speed.** The hardwired WFA has a surface area of $O(|\Sigma| \cdot p \cdot |\delta|)$, where $p$ is the number of bits representing the weight and $|\delta|$ the number of transitions. One character is parsed on every clock cycle. The cycle time is in $O(p \cdot \log d_{\max})$, where $d_{\max}$ is the maximum incoming degree.

**Prototype Implementation.** Our practical implementation uses the *R-disk prototype*, a parallel architecture designed for mass data filtering [6]. Data is distributed among several nodes linked by an Ethernet network, and each node houses a hard disk drive and a reconfigurable processor (a FPGA) which filters data in an on-the-fly way. A host computer send queries and collects results (Figure 2). With the current reconfigurable chip, the size constraint is $p \cdot |\delta| \leq 600$. Therefore, one can use WFA with 75 transitions and 8 bits weights. That covers common biological patterns like those of [1]. The speed constraint is less restrictive, as the clock runs at 40 MHz, and each board can filter data at 16 MB/s. That flow on a single board is more than *4 times faster* than software simulation of WFA [4] on a 2 GHz PC. Massive parallelism is achieved through parallelization of several boards.

**Additional Transitions.** The size limit becomes crucial in some applications, for example when WFA modelize insertions or deletions in patterns. The linear encoding scheme prevents chains of $\epsilon$-transitions from being arbitrarily long (Figure 4) because of the increasing critical path. Systolic techniques could realize $\epsilon$-transitions, but they are only suitable for linear-shaped WFA. One solution is to add *pseudo-deletion transitions* for each possible path (Figure 5). There are $O(|\delta|^2)$ such transitions in the general case, but only a subset of them is sufficient for practical implementation to fit into the available chip size.

We are currently assembling a prototype of this architecture with 48 boards, and we expect a maximum flow of 768 MB/s. Moreover, as the available size on reconfigurable processors evolves faster than the computing power of conventional sequential machines, this kind of space computing will be even more advantageous [2].

## References


[1] P. Bucher and A. Bairoch. A generalized profile syntax for biomolecular sequences motifs and its function in automatic sequence interpretation. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *2nd International Conference on Intelligent Systems for Molecular Biology, ISMB 1994, Stanford, California, USA, August 15–17, 1994, Proceedings*, pages 53–61, Menlo Park, 1994. AAAI Press.

[2] A. DeHon. Very large scale spatial computing. In C. Calude, M. J. Dinneen, and F. Peper, editors, *3rd International Conference on Unconventional Models of Computation, UMC 2002, Kobe, Japan, October 15-19, 2002, Proceedings*, volume 2509 of *Lecture Notes in Computer Science*, pages 27–37. Springer, 2002.

[3] J. Dunoyer, F. Pétrot, and L. Jacomme. Stratégies de codage des automates pour des applications basse consommation: expérimentation et interprétation. In *Journées d'étude Faible Tension et Faible Consommation (FTFC 97)*, 1997.

[4] M. G. Eramian. Efficient simulation of nondeterministic weighted finite automata. In *4th Workshop on Descriptional Complexity of Formal Systems, DCFS 2002, London, Ontario, Canada, August 22-24, 2002, Proceedings*. University of Western Ontario, 2002.

[5] M. Giraud and D. Lavenier. Réalisation matérielle d'automates pondérés pour la recherche de motifs génomiques. In *9ème Symposium en Architecture et Adéquation Algorithme Architecture, SympAAA 2003, La Colle sur Loup, France, October 15–17, 2003, Proceedings*, pages 345–352, 2003.

[6] D. Lavenier, S. Guyetant, S. Derrien, and S. Rubini. A reconfigurable parallel disk system for filtering genomic banks. In *International Conference on Engineering of Reconfigurable Systems and Algorithms, ERSA 2003, Las Vegas, Nevada, USA, June 23–26, 2003, Proceedings*, 2003.

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

[8] R. Sidhu and V. Prasanna. Fast regular expression matching using FPGAs. In *9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2001, Rohnert Park, California, USA, April 29 – May 2, 2001, Proceedings*, 2001.

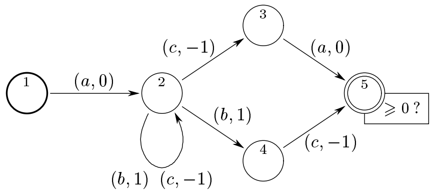



*Fig. 1 – The WFA $\mathcal{A}_1$*

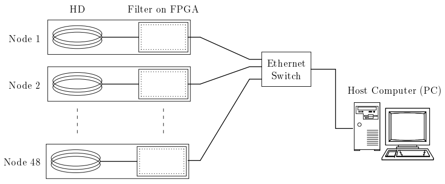



*Fig. 2 – The R-disk Architecture*

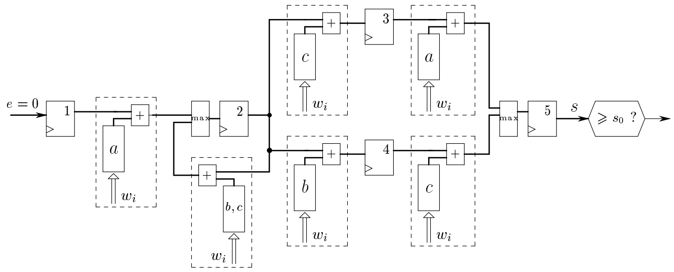



*Fig. 3 – Linear Encoding Scheme for $\mathcal{A}_1$*

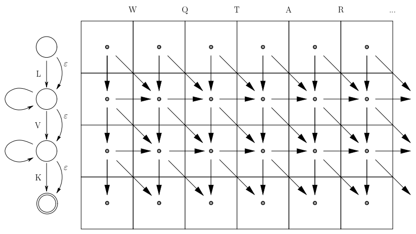



*Fig. 4 – Deletions with $\epsilon$-transitions*

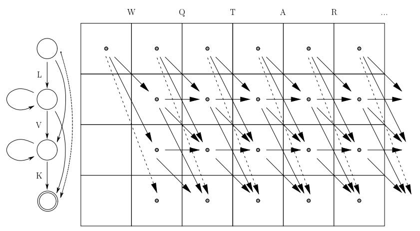



*Fig. 5 – Pseudo-deletion transitions*

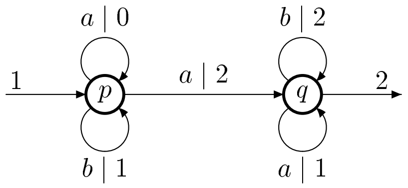
# Rational Series with Multiplicities in $(\min,+)$ and $(\max,+)$ Semirings and Unambiguity

Sylvain Lombardy

LIAFA
Université Paris 7
2, Place Jussieu, F-75251 Paris Cedex 5
France

lombardy@liafa.jussieu.fr

In the sequel, $A$ is a finite alphabet and $A^*$ is the free monoid over $A$. A $(\max,+)$-automaton $\mathcal{A}$ over $A^*$ is an automaton where every transition is not only *labelled* by a letter of $A$, but also *weighted* with a coefficient that is a real. More, every initial or terminal arrow has also a weight. The weight of a (successful) path (from an initial state to a terminal state) is the usual sum of the weights of every transition of the path and of the weights of the initial and terminal arrows. The *weight of a word* $w$ in $\mathcal{A}$ is the *maximal* weight of successful paths labelled by $w$. If $w$ labels no successful path, its weight is equal to $-\infty$.




| word | weight |
|---|---|
| $a$ | 5 |
| $b$ | $-\infty$ |
| $aa$ | $\max(5,6)=6$ |
| $ab$ | 7 |
| $ba$ | 6 |

The function (or series) that maps every word of $A^*$ onto its weight in $\mathcal{A}$ is the ($(\max,+)$-rational) series *realized* by $\mathcal{A}$. The support of the series realized by $\mathcal{A}$ is the set of words which have a weight different from $-\infty$.

A $(\min,+)$-automaton $\mathcal{I}$ on $A^*$ is defined the same way, except that the weight of a word $w$ in $\mathcal{I}$ is the *minimal* weight of successful paths labelled by $w$; if $w$ labels no successful path, its weight is equal to $+\infty$; the support of the series realized by $\mathcal{I}$ is the set of words which have a weight different from $+\infty$.

A $(\max,+)$-rational series $s$ and a $(\min,+)$-rational series $t$ are *equal* if they have the same support and if the weights of every word of this support are the same. In this case, we say that the $(\max,+)$-rational series $s$ *is* $(\min,+)$-rational.

An automaton $\mathcal{A}$ is *unambiguous* if, for every word $w$ of $A^*$, there is at most one successful path in $\mathcal{A}$ labelled by $w$. It is quite obvious that the $(\max,+)$-rational series realized by an unambiguous $(\max,+)$-automaton $\mathcal{A}$ is $(\min,+)$-rational. Actually, if $\mathcal{A}$ is considered as a $(\min,+)$-automaton, it still realizes the same series.

The converse result is presented here: the only rational series which are both $(\min,+)$-rational and $(\max,+)$-rational are exactly the series that can be realized by an unambiguous automaton.

**Theorem.** $\mathrm{Rat}_{\max}A^* \cap \mathrm{Rat}_{\min}A^* = \mathrm{NAmb}A^*$.

This result proves the equality of both families that naturally appear in the framework of $(\max, +)$- and $(\min, +)$-rational series. Rational series that belong to this family are rather nice. For instance, it is decidable [1], whether an unambiguous $(\max, +)$-rational series is sequential (*i.e.* whether it can be realized by a deterministic automaton). Nevertheless, as far as we know, the decidability of the unambiguity of $(\max, +)$- or $(\min, +)$-rational series is still open.

The author thanks Ines Klimann, Jean Mairesse and Christophe Prieur for their enriching remarks and discussions.

## References

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

# New Algorithms for Weighted Multi-tape Automata

Andre Kempe, Franck Guingne, Florent Nicart

Grenoble Laboratory
Xerox Research Centre Europe
Chemin de Maupertuis 6, F-38240 Meylan
France

{andre.kempe,franck.guingne,florent.nicart}@xrce.xerox.com

Finite state automata (FSAs) and weighted finite state automata (WFSAs) are well-known, mathematically well-defined, and offer many practical advantages [2, 1, 8]. A number of software systems have been designed to manipulate FSAs and WFSAs [5, 12, 10]. Both are widely used in language and speech processing [3, 9, 11]. Most systems and applications are dealing with *one-tape* and *two-tape automata*, also called acceptors and transducers.

*Multi-tape automata* (MTAs) [2, 4] offer additional advantages such as the possibility to represent different types of information used in NLP (e.g., surface word form, lemma, POS-tag, domain-specific information) on different tapes, or to preserve intermediate results of transduction cascades on different tapes. Operations can work on one, several, or all tapes. MTAs have been implemented and used, e.g., for the morphological analysis of Semitic languages, where the vowels, consonants, pattern, and surface form of words have been represented on different tapes of an MTA [6, 7].

This paper proposes three new operations for *weighted multi-tape automata* (WMTAs), which we call *auto-intersection*, *single-tape-intersection* and *multi-tape-intersection*. Furthermore, the paper proposes algorithms for these operations that have been implemented and illustrates them through an example of transduction cascades. Although all definitions and algorithms are given for WMTAs, they can be easily adapted to non-weighted MTAs since MTAs are equivalent to WMTAs over the Boolean semiring.

## References

[1] S. Eilenberg. *Automata, Languages, and Machines — Volume A*, volume 59 of *Pure and Applied Mathematics*. Academic Press, 1974.

[2] C. C. Elgot and J. E. Mezei. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68, 1965.

[3] R. M. Kaplan and M. Kay. Phonological rules and finite-state transducers. In *Annual Winter Meeting of the Linguistics Society of America*, New York, 1981.

[4] R. M. Kaplan and M. Kay. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378, 1994.

[5] L. Karttunen, J.-P. Chanod, G. Grefenstette, and A. Schiller. Regular expressions for language engineering. *Journal of Natural Language Engineering*, 2(4):307–330, 1997.

[6] G. A. Kiraz. Linearization of nonlinear lexical representations. In J. Coleman, editor, *3rd Meeting of the ACL Special Interest Group in Computational Phonology, SIGPHON 1997, Madrid, Spain, July 12, 1997, Proceedings*, pages 57–62, Somerset, New Jersey, 1997. Association for Computational Linguistics.

[7] G. A. Kiraz and E. Grimley-Evans. Multi-tape automata for speech and language systems: A prolog implementation. In D. Wood and S. Yu, editors, *2nd International Workshop on Implementing Automata, WIA 1997, London, Ontario, Canada, September 18–20, 1997, Revised Papers*, volume 1436 of *Lecture Notes in Computer Science*, pages 87–103, Heidelberg, 1998. Springer.

[8] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science.* Springer, 1986.

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

[10] M. Mohri, F. C. N. Pereira, and M. Riley. A rational design for a weighted finite-state transducer library. In D. Wood and S. Yu, editors, *2nd International Workshop on Implementing Automata, WIA 1997, London, Ontario, Canada, September 18–20, 1997, Revised Papers*, volume 1436 of *Lecture Notes in Computer Science*, pages 144–158, Heidelberg, 1998. Springer.

[11] E. Roche and Y. Schabes. *Finite-State Language Processing.* Language, Speech, and Communication. MIT Press, 1997.

[12] G. van Noord. FSA utilities: A toolbox to manipulate finite-state automata. In D. R. Raymond, D. Wood, and S. Yu, editors, *1st International Workshop on Implementing Automata, WIA 1996, London, Ontario, Canada, August 29–31, 1996, Revised Papers*, volume 1260 of *Lecture Notes in Computer Science*, pages 87–108, Heidelberg, 1997. springer.

# Weighted Automata and Quantum Finite Automata

M. V. Panduranga Rao, V. Vinay

Department of Computer Science and Automation
Indian Institute of Science
560012 Bangalore
India

{pandurang,vinay}@csa.iisc.ernet.in

Quantum finite automata (QFA) derive their strength by exploiting complex valued probability amplitudes and interference therein. This paper is a step towards finding the least powerful computing model which is purely classical and can capture the dynamics of quantum phase.

A variety of QFA models have been proposed. They range in power from the very weak one-way QFA [3, 4], which can recognize only a subset of regular languages, to Kondacs and Watrous' powerful two-way QFA [3], which is capable of recognizing some non-regular languages also. One such model is the two-way quantum finite automata with classical and quantum states (2QCFA) [1]. In this paper we give an efficient procedure for constructing a weighted automaton, given any 2QCFA, recognizing the same language.

We use a variant of the model of weighted automata as defined in [2]. Our model differs from the weighted automata model in that it consists of a tape on which the input resides, and a two-way head. Consider a weighted automaton $W$ defined over the semiring $\mathbb{K}$. $W$ recognizes an input $x$ in two steps [2]:

(I) Computation Phase: it associates a weight $w \in \mathbb{K}$ to $x$.

(II) Acceptance Phase: checks if the weight $w$ so computed belongs to a given fixed subring $\mathbb{J} \subseteq \mathbb{K}$.

The time taken for Step (I) is proportional to the number of tape cells scanned. Step (II) involves searching for $w$ in $\mathbb{J} \subseteq \mathbb{K}$. Our simulation employs weights from the semiring $\mathbb{K} = \mathbb{R}$. The weighted automaton obtained by our construction scans exactly the same cells as the 2QCFA being simulated and accomplishes Step (II) in constant time.

## References

[1] A. Ambainis and J. Watrous. Two-way finite automata with quantum and classical states. *Theoretical Computer Science*, 287(1):299–311, 2002.

[2] C. Cortes and M. Mohri. Context-free recognition with weighted automata. *Grammars*, 3(2–3):133–150, 2000.

[3] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *38th Annual Symposium on Foundations of Computer Science, FOCS 1997, Miami Beach, Florida, USA, October 19–22, 1997*, pages 66–75. IEEE Computer Society, 1997.

[4] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1–2):275–306, 2000.

# Properties and Applications of Parametric Weighted Finite Automata

German Tischler

Lehrstuhl für Informatik II
Universität Würzburg
Am Hubland, D-97074 Würzburg
Germany

tischler@informatik.uni-wuerzburg.de

Weighted finite automata are nondeterministic finite automata annotated with real weights on their edges and states. They have been used for about ten years in still-image and video compression. Parametric weighted finite automata (PWFA) are weighted finite automata parameterized with a natural number and thus are a generalization of weighted finite automata. As weighted finite automata compute real functions, parametric weighted finite automata compute functions on a multidimensional codomain. An interpretation function can be used to map points in this codomain into some space suitable for visualization, for example a bi-level image if the parameter is two. We show that the set of sets definable by parametric weighted finite automata is closed under set union, affine transformations and regular limitation of the domain language. Further we show that this set is not closed under the set complement operation and thus also not closed under set intersection. As PWFA can compute polynomials on finite intervals and are closed under affine transformation and set union, they can also display Bezier polynomials, B-splines and Catmull-Rom splines. Polynomials with real domain as well as some rational functions are computable by PWFA. Using the shown properties, we present a simple algorithm to infer parametric weighted finite automata from line-art style bi-level images, that for example can be obtained by applying an edge detection algorithm on some suitable source image. One particularly interesting class of input images are character shapes as defined by fonts. Having defined the letters a to z in basic and capital form and using the properties of closedness under set union and affine transformation, every English text can be rendered by a parametric weighted finite automaton. During an iterative build procedure of the automaton parts larger than one character can be reused for example in a scheme similar to the well-known Ziv-Lempel compression for text.

# Simplification of Intermediate Results during Intersection of Multiple Weighted Automata

Anssi Yli-Jyrä

Department of General Linguistics
P.O Box 9 (Siltavuorenpenger 20A)
FIN-00014 University of Helsinki
Finland

anssi.yli-jyra@helsinki.fi

This paper presents a new simplification method for weighted finite automata. The method suggests efficient approaches to some problems that are related to intersection of automata.

Minimal automata can be made smaller through simplifications such as *merging of states*. Although such simplifications change the recognized language, some other essential properties of the automaton may be preserved. For example, if we want to count, in a deterministic automaton, the strings whose length is below some limit, we can first merge some states and then count the number of strings using the simplified result that may be smaller than a minimal automaton [3]. The idea of various simplifications that preserve certain essential properties lends itself also to the case where the strings are described conjunctively, by an intersection of multiple automata. State merging when computing properties of an intersection can reduce the overall time complexity, and it resembles the projection operation of relation tables that is used in query optimization in modern database systems [5].

Our method assumes that an intersection of multiple minimal automata is carried out through pairwise intersections of automata. Some properties in the automata to be intersected indicate redundancy that leads to simple optimizations that are possible especially in unweighted automata, but also in some classes of weighted automata:

- The letters outside the intersection of the alphabets of the automata are useless.
- When we have already computed an intersection of a pair of automata, we can *save* one of the input automata as a so-called *reference automaton* and use it for further simplifications of the intersection result: states in the result can be merged as long as the simplified automaton rejects all strings of the reference automaton that are rejected by the intersection result.
- During a pairwise intersection, letters that never change the state of a (completely specified) input automaton, can be substituted with $\epsilon$ in both the automata. The substituted letters can be restored later by means of an inverse mapping from $\epsilon$ and the reference automaton.
- Trivial cycles on an input letter can be added to the simplified automaton, if this change does not add new strings to the intersection of the reference automaton and the simplified result.

These reductions are based on the structure of the intersected automata. Especially, we will merge only such states that are *connected* because this policy helps to preserve some properties throughout the simplifications. To enable more state merges, we *expand the common alphabet* of the automata in such a way that it is possible to determine the states of the reference automaton through local or piecewise testable properties of the strings. The representation for the intersection result comprises, thus, at least one reference automaton, the simplified automata and a mapping from the expanded alphabet to the original one. Our representation is related to automata decompositions using covers [9].

Efficient algorithms for the proposed method will be given. Some precursors of these algorithms have been known for some years [6, 7], but our new algorithms merge more states and they can be applied to a wider range of automata. Because merging of connected states resembles $\epsilon$-removal, there remain some restrictions on the generic applicability [2]. The new algorithms have been implemented as an extension (*fsiglibrary*) to AT&T's *fsmlibrary* collection of tools for manipulation of weighted finite transducers.

The original motivation for these methods comes from the framework of Finite-State Intersection Grammar (FSIG) [1] that is used for natural language parsing and disambiguation. The existing FSIG systems parse sentences by intersecting some 100 – 2600 deterministic finite automata, and the parsing result usually consists of only a small number of strings. Tapanainen [4] has investigated various strategies for constructing the final result by intersecting a pair of automata at a time in different orders and observed that some of the intermediate automata can become extremely large. In contrast to Tapanainen (ibid.), our representation for intermediate results involves a set of automata. Such a representation for the set $L$ of strings recognized by all the input automata can be exponentially smaller than the fully expanded intersection.

In some special cases, the emptiness of $L$ can be decided efficiently simply by computing the pairwise simplified intersections in a careful order, without any need to expand the simplified automata into the final intersection automaton. Such an efficient solution for the emptiness of the intersection is obtained if the automata describe strings that encode tree structures through balanced bracketing. This is true, in particular, for the automata of a *Bracketed FSIG* (B-FSIG). The automata of a B-FSIG can be divided into smaller automata each of which describes properties at different nesting levels [8]. When the automata for different levels are intersected in an appropriate order, the simplification method presented implements a kind of *structure sharing* through which the exponential growth of intermediate results is avoided until the compact representation is expanded to the final intersection result.

The structure sharing of balanced bracketings enables the following applications of our compact representation:

- backtrack-free search for a string belonging to $L$;
- constructing directly the final intersection result where all the states are both accessible and co-accessible;
- efficient enumeration of the strings of $L$ in a lexicographical order;
- (possibly) searching for the best weighted string in $L$ in a way that uses local ambiguity packing; and

- (possibly) computing the string count of $L$ (if $L$ is finite) without expanding the compact representation into a single automaton.

These possibilities remain to be studied later and they are not covered by our presentation.

## References


[1] K. Koskenniemi, P. Tapanainen, and A. Voutilainen. Compiling and using finite-state syntactic rules. In C. Boitet, editor, *14th International Conference on Computational Linguistics, COLING-92, Nantes, France, July 23–28, 1992, Proceedings*, volume 1, pages 156–162. Association for Computational Linguistics, 1992.

[2] M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.

[3] B. Ravikumar. Weak minimization of DFA — an algorithm and applications. In Z. Dang O. H. Ibarra, editor, *8th International Conference on Implementation and Application of Automata, CIAA 2003, Santa Barbara, California, USA, July 16–18, 2003, Proceedings*, volume 2759 of *Lecture Notes in Computer Science*, pages 226–238, Heidelberg, 2003. Springer.

[4] P. Tapanainen. *Applying a Finite-State Intersection Grammar*, chapter 10, pages 311–327. Language, Speech, and Communication. MIT Press, 1997.

[5] J. D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume 1–2. Computer Science Press, New York, 1988.

[6] A. Yli-Jyrä. Schematic finite-state intersection parsing. In K. Koskenniemi, editor, *10th Nordic Conference of Computational Linguistics, NODALIDA 1995, Helsinki, Finland, May 29–30, 1995, Short Papers*. Department of General Linguistics, University of Helsinki, 1995.

[7] A. Yli-Jyrä. Menetelmiä äärellisiin automaatteihin perustuvan lauseenjäsennyksen tehostamiseksi (methods for improving parsing efficiency in finite-state syntax). Master's thesis, Department of General Linguistics, University of Helsinki, 1997.

[8] A. Yli-Jyrä and K. Koskenniemi. Compiling contextual restrictions on strings into finite-state automata. unpublished, 2004.

[9] H. P. Zeiger. Cascade decomposition of automata using covers. In M. A. Arbib, editor, *Algebraic Theory of Machines, Languages, and Semigroups*, pages 55–80. Academic Press, 1968.

# Optimization Problems with Mixed $(\bigoplus, \bigotimes)$-linear Constraints

Karel Zimmermann[†]

Faculty of Mathematics and Physics
Charles University in Prague
Ke Karlovu 3, CZ-121 16 Praha 2
Czech Republic

karel.zimmermann@mff.cuni.cz

A considerable research effort was devoted to special algebraic structures, so called $(\bigoplus, \bigotimes)$-algebras in the past decades. In these structures, operations $\bigoplus$ and $\bigotimes$ are introduced, which play a similar role as the usual operations of addition and multiplication in classical linear algebra. As examples of such algebraic structures may serve the $(\max, +)$-algebra, $(\min, \max)$-algebra or $(\max, \cdot)$-algebra, which appeared under various names in some articles in the past (Cuninghame-Green, Gaubert, Gondran, Minoux, Vorobjov and others). The operations $\bigoplus$ and $\bigotimes$ can be extended to Cartesian products of a finite number of $(\bigoplus, \bigotimes)$-algebras; functions defined on such products with a range in a $(\bigoplus, \bigotimes)$-algebra, which are "$\bigoplus$-additive" and "$\bigotimes$-homogeneous" are called $(\bigoplus, \bigotimes)$-linear. Although optimization problems with $(\bigoplus, \bigotimes)$-linear objective functions and $(\bigoplus, \bigotimes)$-linear constraints are in general non-convex and non-differentiable optimization problems, special structural properties of such problems made it possible to develop efficient algorithms, which find their global optimal solution. Main features of such algorithms will be surveyed and other algorithms for solving some classes of optimization problems, in the constraints of which $(\bigoplus, \bigotimes)$-linear functions with respect to several pairs of operations $(\bigoplus, \bigotimes)$ occur (e.g. $(\max, +)$-linear, $(\max, \min)$-linear and usual linear functions). All $(\bigoplus, \bigotimes)$-algebras considered in this contribution are subsets of the real numbers. The complexity of the suggested algorithms and some illustrative examples from operations research will be briefly discussed.

[†]Under the support of GA ČR 201/01/343

# A Consistent and Efficient Estimator for Data-Oriented Parsing

Andreas Zollmann

Institute for Logic, Language, and Computation
University of Amsterdam
NL-1018 TV Amsterdam
The Netherlands

azollman@science.uva.nl

Data-Oriented Parsing (DOP) [2, 1] is a Statistical Parsing approach in which subtrees (called *fragments*) are extracted from the full-parse trees of the tree bank during the training process. These fragments are assigned weights between 0 and 1. Fragments can be recombined in a certain way to full-parse trees. This way, *new* parse trees—trees that not necessarily occur in the training corpus—can be obtained. The probability assigned to a full-parse tree under DOP is determined by the weights of the fragments from which that parse tree can be built up.

Although already achieving state-of-the-art performance, the commonly used model *DOP1* lacks a formal justification, such as, e.g., the *maximum-likelihood principle* common in statistical NLP. Furthermore, it has the disturbing property of *inconsistency* [4]: The probabilities that DOP assigns to parse trees do not converge to the relative frequencies of these trees in the training corpus when that corpus grows large. As we will demonstrate, the failure to fulfill this property causes DOP to incorrectly rank different possible parses for a given sentence based on the evidence in the training corpus. A related problem is the model's bias towards fragments of large parse trees.

Aside from estimation issues, the fact that a parse tree contains an exponential number of DOP fragments in terms of the size (i.e., the number of nodes) of the tree has consequences for DOP's computational properties. Whereas PCFG-based models have algorithms for computing the most probable parse in polynomial time w.r.t. sentence length, DOP's case is NP-complete [5] and is dominated by the huge size of the grammar.

**Contributions** We devise a new DOP model that satisfies the property of consistency. The difficulty lies in the problem that the maximum-likelihood method, according to which the joint probability of the parse trees in the training corpus should be maximized, an estimation method often leading to consistency, is not suitable for DOP: Standard maximum-likelihood estimation results in an *overfitted* (though consistent) estimator that only learns the parses directly occurring in the training data [3].

We employ a generalized maximum-likelihood approach leading to the first nontrivial consistent estimator for DOP. Our estimator DOP$^*$ applies held-out training: Grammar productions are extracted from one part of the training corpus, and their probabilistic weights are assigned based on their participation in derivations of the trees from a distinct, *held-out*, part of the corpus. While the formula for those weight assignments is derived from the optimization problem of maximizing the likelihood of the held-out corpus, a simplifying assumption in introduced during the solution process. This assumption leads to a computationally inexpensive closed-form solution and at the same time causes DOP* to prefer learning simpler derivations of parses during the training process.

As a side product of the estimation mechanism, the estimator learns only a number of fragments that is linear in the total number of nodes of the trees in the training corpus, thereby circumventing the inefficiency problems of original DOP without giving up on the idea of using arbitrary-size fragments.

Empirical results on the OVIS corpus demonstrate improvements in parsing accuracy over the original DOP estimator.

# References


[1] R. Bod. *Beyond Grammar: An Experience-Based Theory of Language.* CSLI Publications, California, 1998.

[2] R. Bod and R. Scha. Data-oriented language processing: An overview. Technical Report LP-96-13, Institute for Logic, Language, and Computation, University of Amsterdam, 1996.
available at: `http://www.essex.ac.uk/linguistics/clmt/papers/dop/bodscha.ps`.

[3] R. Bonnema. Data oriented semantics. Master's thesis, University of Amsterdam, 1997.

[4] M. Johnson. The DOP estimation method is biased and inconsistent. *Computational Linguistics*, 28(1):71–76, 2002.

[5] K. Sima'an. *Learning Efficient Disambiguation.* Institute for logic, language, and computation dissertation series 1999-02, University of Utrecht, 1999.

[6] K. Sima'an and A. Zollmann. A consistent and efficient estimator for data-oriented parsing.
available at: `http://staff.science.uva.nl/~azollman/publications.html`, 2004.

[7] A. Zollmann. A consistent and efficient estimator for the data-oriented parsing model. Master's thesis, Institute for Logic, Language, and Computation, University of Amsterdam, 2004.
available at: `http://staff.science.uva.nl/~azollman/publications.html`.

# Home-speakers

# Semiring Artificial Neural Networks and Weighted Automata

Sebastian Bader[1], Alexandre Scalzitti[2]

Department of Computer Science
Dresden University of Technology
D-01062 Dresden
Germany

[1] `sebastian.bader@inf.tu-dresden.de` [2] `scal@tcs.inf.tu-dresden.de`

In their seminal paper McCulloch and Pitts have shown the strong relationship between finite automata and so-called McCulloch-Pitts networks. Our goal is to extend this result to weighted automata. In other words, we want to integrate artificial neural networks and weighted automata. For this task, we introduce semiring artificial neural networks, that is, artificial neural networks which implement the addition and the multiplication of semirings. We present a construction of a semiring artificial neural network from a given weighted automaton, and back again. After that, we show how we can approach the problem of encoding an image into a weighted automaton by using a semiring artificial neural network in this process.

[1]Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft).

# The Theory of Recognizable Tree Series

Björn Borchardt[†]

Department of Computer Science
Dresden University of Technology
D-01062 Dresden
Germany

borchard@tcs.inf.tu-dresden.de

A tree series is a mapping $S : T_\Sigma \to A$ from the set $T_\Sigma$ of all trees over a given ranked alphabet $\Sigma$ into the set $A$. In automata theory tree series occur as semantics of weighted tree automata, which were introduced in [BR82] and investigated in, e.g., [AB87, Boz94, Sei94, Kui97, Boz01, ÉK03, BV03, Bor03, Bor04], where it is assumed that $A$ is the carrier set of some semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$. We consider bottom-up finite state weighted tree automata (for short: bu-w-fta), which generalize both, finite state weighted string automata and finite state tree automata. More precisely, the input of a bu-w-fta is a tree, and by letting such a device run on an input tree, weights (also: costs) coming from the semiring $\mathcal{A}$ are accumulated. Hence, trees are accepted by a bu-w-fta with a weight, i.e., the semantics of a bu-w-fta is a recognizable tree series.

In the talk we give an overview over the results which I have obtained during my research aiming for a PhD. In particular, we compare the power of deterministic and non-deterministic bu-w-fta and thereby provide a determinization of these devices. It turns out that deterministic and non-deterministic bu-w-fta are equally powerful, if the underlying semiring is a commutative semifield. We also characterize recognizable tree series in terms of the number of equivalence classes of a congruence relation. This result generalizes the well-known MYHILL-NERODE Theorem of classical theories. Finally, we prove decidability results of recognizable tree series, which requires us to derive a pumping lemma. It turns out that it is decidable, whether a tree series, which is accepted by a deterministic bu-w-fta, is constant, constant on its support, boolean, or polynomial.

## References

[AB87] A. Alexandrakis and S. Bozapalidis. Weighted grammars and KLEENE's theorem. *Information Processing Letters*, 24(1):1–4, 1987.

[Bor03] B. Borchardt. The MYHILL-NERODE theorem for recognizable tree series. In *7th International Conference on Developments in Language Theory, DLT 2003, Szeged, Hungary, July 7–11, 2003, Proceedings*, volume 2710 of *Lecture Notes in Computer Science*, pages 146–158. Springer, 2003.

[Bor04] B. Borchardt. A pumping lemma and decidability problems for recognizable tree series. *Acta Cybernetica*, 2004. to appear.

[†]Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft).

[Boz94] S. Bozapalidis. Representable tree series. *Fundamenta Informaticae*, 21(4):367–389, 1994.

[Boz01] S. Bozapalidis. Context-free series on trees. *Information and Computation*, 169(2):186–229, 2001.

[BR82] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18(2):115–148, 1982.

[BV03] B. Borchardt and H. Vogler. Determinization of finite state weighted tree automata. *Journal of Automata, Languages and Combinatorics*, 8(3):417–463, 2003.

[DV03] M. Droste and H. Vogler, editors. *1st Workshop on Weighted Automata: Theory and Applications, Dresden, Germany, March 4–8, 2002, Selected Papers*, volume 8 of *Journal of Automata, Languages and Combinatorics*, 2003.

[ÉK03] Z. Ésik and W. Kuich. Formal tree series. In Droste and Vogler [DV03], pages 219–285.

[Kui97] W. Kuich. Formal power series over trees. In S. Bozapalidis, editor, *3rd International Conference on Developments in Language Theory, DLT 1997, Thessaloniki, Greece, July 20–23, 1997, Proceedings*, pages 61–101. Aristotle University of Thessaloniki, 1997.

[Sei94] H. Seidl. Finite tree automata with cost functions. *Theoretical Computer Science*, 126(1):113–142, 1994.

# Weighted Büchi Automata

Manfred Droste[1], Ulrike Püschmann[2]

[1] Department of Computer Science, Universität Leipzig, PF 920, D-04009 Leipzig, Germany

[2] Institute of Algebra, Dresden University of Technology, D-01062 Dresden, Germany

[1] droste@informatik.uni-leipzig.de [2] up@math.tu-dresden.de

We will investigate weighted Büchi automata, i. e., weighted finite automata that run on infinite words. The behaviour of these automata can, similarly to the case of finite words, be described by formal power series, that is, functions that assign a value or weight taken from a semiring to every infinite word. Schützenberger's well-known result describes the behaviours of weighted automata on finite words precisely as the rational formal power series. An important result of Büchi characterizes the languages of infinite words recognized by Büchi automata as the $\omega$-rational languages.

Our aim here is to combine these two classical results and to show that the formal power series which are the behavior of some weighted Büchi automaton coincide with the $\omega$-rational formal power series, i. e., they can be obtained by natural operations from very simple ones.

In our situation, the words are infinite. This causes difficulties calculating their weight, since infinite sums and products might occur. In the case that the weights are taken from the tropical semiring over the nonnegative real numbers, M. Droste und D. Kuske ensured convergence of infinite sums and products by introduction of a deflation parameter. In very recent work, Ésik and Kuich overcame these difficulties by assuming certain completeness assumptions for the operations of the underlying semiring.

In our work, we are assuming the semiring to be ordered and to satisfy certain completeness assumptions for the order. Our automaton model also takes weights of final states into account which has natural intuitive interpretations and is similar to the case of classical weighted automata on finite words. Then we provide automata-theoretic constructions to establish our main result.

[2]Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft)

# On Subsequential, Path-neutral, and Unambiguous Weighted Automata†

Daniel Kirsten‡, Ina Mäurer§

Institute of Algebra
Dresden University of Technology
D-01062 Dresden
Germany

{kirsten,maeurer}@math.tu-dresden.de

In the talk, we consider several classes of weighted finite automata (WFA). The notion of an *unambiguous* WFA gives a proper extension of the strength of computability defined by *subsequential* WFA. We introduce *path-neutral* WFA and give characterizations of them in the case of the tropical semiring, e.g., the mapping of a path-neutral WFA is subsequential iff it has *bounded* variation. The latter case coincides with the condition that the given WFA has the *twins property*. The equivalent subsequential WFA then can be effectively constructed.

For the tropical semiring, it is decidable whether a WFA is path neutral and for a path neutral WFA whether it computes a subsequential function.

## References

[1] D. Kirsten and I. Mäurer. On the determinization of weighted automata. Manuscript, 2003.

†A complete version is available under http://www.math.tu-dresden.de/~kirsten/publications.

‡Supported by the research grant KI 822/1–1 of the German Research Foundation (Deutsche Forschungsgemeinschaft).

§Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft) and the Freistaat Sachsen.

# Weighted Asynchronous Cellular Automata


Dietrich Kuske[1], Rémi Morin[2]

[1] Institut für Algebra, Technische Universität Dresden, D-01062 Dresden, Germany

[2] Centre de Mathématiques et Informatique, Technopôle Château-Gombert, rue F. Joliot Curie 39, F-13453 Marseille Cedex 13, France

[1] `kuske@math.tu-dresden.de` [2] `morin@cmi.univ-mrs.fr`


Mazurkiewicz traces form a well-established model of concurrent systems. The idea is that certain basic actions are independent of each other while other pairs are dependent. For independent actions $a$ and $b$, the actual order in which they appear is irrevelant. Gastin and Droste introduced weighted trace automata that differ from classical automata in the additional assumption that the weight of all paths from state $p$ to state $q$ labeled $ab$ equals that of all paths from $p$ to $q$ labeled $ba$ — thereby reflecting that the order of these two independent actions is irrelevant. They succeeded in describing the possible behaviors (i.e., formal power series) of these systems using c-rational operations (a restriction of the well-known rational operations addition, multiplication, and iteration).

The usual source of independence of basic actions is the disjointness of resources used by these actions. This aspect, that is abstracted away in Droste and Gastin's global approach, is the main motivation of our work: can any behavior of a global automaton in the sense of Droste and Gastin be modeled by a "distributed one"? To answer this question, we provide the transitions of Zielonka's asynchronous cellular automata by weights in a natural way. Then we can indeed show that this distributed model allows precisely those behaviors that can be realized by global automata. Furthermore, we describe a class of global weighted trace automata whose behavior can even be realized by a deterministic weighted asynchronous cellular automaton.

# Relating Tree Transducers and Weighted Tree Automata

Andreas Maletti[†]

Department of Computer Science
Dresden University of Technology
D-01062 Dresden
Germany

maletti@tcs.inf.tu-dresden.de

A folklore result in automata theory relates (non-deterministic) generalized sequential machines (i.e., string-to-string-language transducers) and (non-deterministic) weighted string automata over the semiring of formal languages. Similarly, we show how this result might be generalized to bottom-up tree transducers (i.e., tree-to-tree-language transducers) and bottom-up weighted tree automata over a particular semiring. More precisely, this result can even be established for bottom-up tree series transducers (using pure substitution) and bottom-up weighted tree automata. Using this relation we can reprove the classical pumping lemma for bottom-up tree transducers as well as present a pumping lemma for bottom-up tree series transducers both derived from the pumping lemma for bottom-up weighted tree automata.

[†]Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft).

# Weighted Branching Automata — Combining Parallelism and Weights

Ingmar Meinecke[†]

Institute of Algebra
Dresden University of Technology
D-01062 Dresden
Germany

meinecke@math.tu-dresden.de

Extending work by Lodaya and Weil, we propose a model of weighted branching automata running on sequential-parallel posets in which the calculation of the weight of a parallel composition is handled differently from the calculation of the weight of a sequential composition.

Suppose we wanted to calculate the minimal duration of a run in a modularly constructed system build by nondeterministic choice, iteration, and sequential and parallel composition. The execution time of a sequential composition is the sum of the durations, and that of a parallel composition is the maximum of the durations of the arguments, possibly increased by some duration for the necessary fork and join operations at the beginning and end. A given sequential-parallel poset can be executed in different ways and we should consider the minimal duration of all possible executions. In order to accompany this situation, we introduce bisemirings, i.e. structures consisting of two semirings on a joint domain with the same additive operation. Weights of executions in our model of weighted branching automata are then evaluated in such bisemirings and the behavior of a weighted branching automaton is a function that associates with any sequential-parallel poset an element from the bisemiring.

In this talk we introduce the model of weighted branching automata, and characterize their behavior in the style of Kleene and Schützenberger stating the equivalence of regularity and rationality. Moreover, we discuss a slightly different notion of regularity, and present results about the closure of regular series under Hadamard product.

[†]Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft).

# Formal Series on Infinite Trees

Christian Pech†

Institute of Algebra
Dresden University of Technology
D-01062 Dresden
Germany

pech@math.tu-dresden.de

The aim of this talk will be to characterize recognizable formal power series on infinite trees by rational expressions. The main tool to achieve this goal will be languages of infinite weighted trees.

†Supported by the GK 334 of the German Research Foundation (Deutsche Forschungsgemeinschaft).