TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik

TUD-FI02-02 - March 2002

M. Droste[1] **H. Vogler**[2]

[1] Institute of Algebra

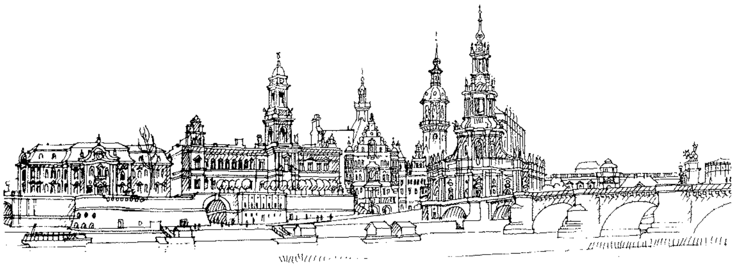
[2] Institute of Theoretical Computer Science

Weighted Automata: Theory and Applications

Technische Berichte

Technical Reports

ISSN 1430-211X



Technische Universität Dresden
Fakultät Informatik
D-01062 Dresden
Germany

URL: http://www.inf.tu-dresden.de/

Weighted Automata: Theory and Applications

Dresden, March 4 – 8, 2002

Preface

This technical report contains the program and the abstracts of lectures delivered at the workshop "Weighted Automata: Theory and Applications" which took place at Dresden University of Technology from March 4 – 8, 2002. 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 algebraic optimization of railway networks or natural language processing.

The workshop was attended by 64 participants from 15 countries.
Four tutorials were given by

S. Gaubert (INRIA, Paris)	W. Kuich (Wien)
D. Kozen (Cornell U., Ithaca)	M. Mohri (AT&T, New Jersey)

In addition, ten invited lectures were presented by

S. Bloom (Hoboken, New Jersey)	B. Khoussainov (Auckland)
Z. Ésik (Szeged)	G.J. Olsder (Delft)
Z. Fülöp (Szeged)	I. Petre (Turku)
B.F. Heidergott (Eindhoven)	E.W. Stark (Stony Brook, New York)
J. Karhumäki (Turku)	K. Zimmermann (Prague)

Furthermore, ten talks were selected as contributed communications.

The workshop was organized jointly by the Graduiertenkolleg *Specification of discrete processes and systems of processes by operational models and logics* and the *Institute of Algebra.* For further financial support we would like to thank *Sächsisches Staatsministerium für Wissenschaft und Kunst* and *Infineon Technologies Dresden GmbH.*

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 as usual. 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@math.tu-dresden.de` or `vogler@inf.tu-dresden.de`. Deadline for submissions is **April 20, 2002**. We intend to ensure a quick refereeing process.

Manfred Droste and Heiko Vogler

Program

<u>**Monday, March 4, 2002**</u>

9:00– 9:45	**W. Kuich:**	Formal power series and formal tree series (part 1)
9:50–10:35	**W. Kuich:**	Formal power series and formal tree series (part 2)
Break		
11:00–11:45	**D. Kozen:**	An axiomatic approach to weighted automata (part 1)
11:50–12:35	**D. Kozen:**	An axiomatic approach to weighted automata (part 2)
Lunch		
14:00–15:00	**S.L. Bloom, Z. Ésik:**	An extension theorem for grove theories
15:10–16:10	**Z. Ésik:**	Equational theories of exotic semirings, an overview
Break		
16:50–17:15	**J. Sakarovitch:**	On rational relations with multiplicity
17:15–17:40	**S. Lombardy, J. Sakarovitch:**	Derivatives of rational expressions with multiplicity
17:40–18:05	**J.J.M.M. Rutten:**	Coinductive counting with weighted automata

<u>**Tuesday, March 5, 2002**</u>

9:00– 9:45	**W. Kuich:**	Formal power series and formal tree series (part 3)
9:50–10:35	**W. Kuich:**	Formal power series and formal tree series (part 4)
Break		
11:00–11:45	**D. Kozen:**	An axiomatic approach to weighted automata (part 3)
11:50–12:35	**D. Kozen:**	An axiomatic approach to weighted automata (part 4)
Lunch		
14:00–15:00	**K. Zimmermann:**	Solution of $(max,+)$-linear inequality systems and some related topics
15:10–16:10	**J. Karhumäki:**	Weighted finite automata in image compression and manipulation
Break		
16:50–17:15	**H. Schulz:**	Computing real functions with weighted finite automata
17:15–17:40	**M. Droste, D. Kuske:**	Weighted automata on infinite words
17:40–18:05	**L. Staiger:**	Weighted finite automata and metrics in Cantor space

<u>**Wednesday, March 6, 2002**</u>

9:00– 9:45	**M. Mohri:**	Weighted automata algorithms (part 1)
9:50–10:35	**M. Mohri:**	Weighted automata algorithms (part 2)
Break		
11:00–11:45 **11:50–12:35**	**S. Gaubert:** **S. Gaubert:**	Max-plus linear dynamical systems: theory and applications (parts 1 & 2)
Lunch		
14:00–	**Excursion**	

Thursday, March 7, 2002

9:00– 9:45	**M. Mohri:**	Weighted automata algorithms (part 3)
9:50–10:35	**M. Mohri:**	Weighted automata algorithms (part 4)
Break		
11:00–11:45	**S. Gaubert:**	Max-plus linear dynamical systems:
11:50–12:35	**S. Gaubert:**	theory and applications (parts 3 & 4)
Lunch		
14:00–15:00	**I. Petre:**	Families of semilinear formal power series
15:10–16:10	**Z. Fülöp, H. Vogler:**	Generalizations of tree transducers to tree series transducers
Break		
16:50–17:15	**P. Buchholz, P. Kemper:**	Weak equivalence for (max/+) automata and related models
17:15–17:40	**C. Martin-Vide, V. Mitrana, R. Stiebe:**	Weighted grammars and automata with maximum interpretation
Discussion		
20:00–	**Conference Dinner**	

Friday, March 8, 2002

9:00–10:00	**G.J. Olsder:**	Max-plus algebra and its applications to railway systems
10:10–11:10	**B. Heidergott:**	Control and optimization of public transportation networks via (max,+) algebra
Break		
11:30–11:55	**P. Asveld:**	On grammatical errors and other imperfections
12:00–12:25	**C. Allauzen, M. Mohri:**	On the determinizability of weighted automata and transducers
Lunch		
14:00–15:00	**B. Khoussainov:**	Automatic algebraic systems
15:10–16:10	**E. W. Stark:**	Performance analysis for probabilistic input/output automata via linear representations

Abstracts

On the Determinizability of Weighted Automata and Transducers

Cyril Allauzen and Mehryar Mohri
{allauzen,mohri}@research.att.com
AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932, USA

Finite automata are classical computational devices used in a variety of large-scale applications [1]. Some applications in text, speech and image processing require more general devices, weighted automata, to account for the variability of the data and to rank various hypotheses [8, 7]. A weighted automaton is a finite automaton in which each transition is labeled with some weight in addition to the usual symbol. Finite-state transducers are automata whose transitions are additionally labeled with an output label.

Weighted automata and transducers provide a common representation for each component of a complex system in many applications and admit general algorithms such as composition to combine these components. The time efficiency of these systems is substantially increased when *deterministic* or *subsequential* machines are used [8] and the size of these machines can be further reduced using general minimization algorithms [8, 9]. A weighted automaton or transducer is *deterministic* or *subsequential* if it has a unique initial state and if no two transitions leaving the same state share the same input label.

A general determinization algorithm for weighted automata and transducers was given by [8]. The algorithm outputs a deterministic machine equivalent to the input weighted automaton or transducer and is an extension of the classical subset construction used for unweighted finite automata. But, unlike the case of unweighted automata, not all finite-state transducers or weighted automata and transducers can be determinized using this algorithm. In fact, some machines do not admit any equivalent deterministic one. Thus, it is important to design an algorithm for testing the determinizability of finite-state transducers and weighted automata.

A characterization of determinizable finite-state transducers based on a *twins property* was given by [5, 6]. The author also proved that the twins property is decidable, see [3] for a presentation of these results in English. A polynomial-time algorithm for deciding the twins property for functional transducers was given by [10], its time complexity is $O(|Q|^4(|Q|^2 + |E|^2)|\Delta|)$ where Q is the set of states of the input transducer, E its set of transitions and Δ the output alphabet.[1]

More recently, [2] proposed a similar polynomial-time algorithm for deciding the twins property for functional transducers, its complexity is $O(|Q|^4(|Q|^2 + |E|^2))$ and only differs from that of [10] by the fact that it doesn't depend on the output alphabet size.[2]

We present a new, conceptually much simpler and computationally much more efficient algorithm for testing the twins property for finite-state transducers. The complexity of our algorithm is $O(|Q|^2(|Q|^2 + |E|^2))$. It is based on a general algorithm of composition of finite-state transducers and a new characterization of the twins property in terms of combinatorics of words.

[1] This is based on our most favorable estimate of the complexity of that algorithm, the authors do not give a precise analysis of the complexity of their algorithm.

[2] The complexities given in [2] are wrong but they have been corrected in an extended version to appear in *Theoretical Computer Science*.

In the case of weighted automata, a similar twins property was introduced by [8] to characterize the determinizability of unambiguous weighted automata over a commutative semiring. The property is also a sufficient condition for the determinizability of ambiguous machines. [8] also gave an algorithm for testing the twins property for unambiguous weighted automata. The complexity of the best existing algorithm so far for testing this property was $O(|E|^2|Q|^6)$ [4]. We present a new and efficient algorithm for testing the twins property for unambiguous and cycle-unambiguous weighted automata whose complexity is $O(|Q|^2 + |E|^2)$. We further conjecture this complexity to be optimal.

The first step of both of our algorithms, transducer and weighted automata cases, consists of constructing the intersection or composition of the input machine M and its *inverse*, $M \circ M^{-1}$, which can be done in quadratic time $O(|Q|^2 + |E|^2)$. We also give an algorithm to test the *functionality* of a transducer T, that is to determine if T represents a function, using the same composed machine $T \circ T^{-1}$. The complexity of our algorithm is $O(|Q|^2 + |E|^2)$ since it only requires work linear in the size of $T \circ T^{-1}$.

References

[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers, Principles, Techniques and Tools*. Addison Wesley: Reading, MA, 1986.

[2] Marie-Pierre Béal, Olivier Carton, Christophe Prieur, and Jacques Sakarovitch. Squaring transducers: An efficient procedure for deciding functionality and sequentiality. In *Proceedings of LATIN'2000*, volume 1776 of *Lecture Notes in Computer Science*. Springer, 2000.

[3] Jean Berstel. *Transductions and Context-Free Languages*. Teubner Studienbucher: Stuttgart, 1979.

[4] Adam L. Buchsbaum, Raffaele Giancarlo, and Jeffery R. Westbrook. On the Determinization of Weighted Finite Automata. *SIAM Journal of Computing*, 30(5):1502–1531, 2000.

[5] Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoretical Computer Science*, 5:325–338, 1977.

[6] Christian Choffrut. *Contributions à l'étude de quelques familles remarquables de fonctions rationnelles*. PhD thesis, (thèse de doctorat d'Etat), Université Paris 7, LITP: Paris, France, 1978.

[7] Karel Culik II and Jarkko Kari. Digital Images and Formal Languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 599–616. Springer, 1997.

[8] Mehryar Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2), 1997.

[9] Mehryar Mohri. Minimization Algorithms for Sequential Transducers. *Theoretical Computer Science*, 234:177–201, 2000.

[10] Andreas Weber and Reinhard Klemm. Economy of Description for Single-Valued Transducers. *Information and Computation*, 118(2):327–340, 1995.

On Grammatical Errors and Other Imperfections

Peter R.J. Asveld

Department of Computer Science, Twente University of Technology
P.O. Box 217, 7500 AE Enschede, the Netherlands
e-mail: infprja@cs.utwente.nl

When we extend a grammar $G = (V, \Sigma, P, S)$ with additional productions to model grammatical errors we end up with some strings over Σ that do not belong completely to either $L(G)$ or to $\Sigma^\star - L(G)$. A similar situation occurs when an automaton or a recognition/parsing algorithm A has been augmented with certain transitions in order to deal with erroneous inputs: A processes strings over Σ that neither fully are in $L(A)$ nor in $\Sigma^\star - L(A)$.

To model these phenomena adequately the notions of fuzzy language, fuzzy grammar and fuzzy automaton turn out to be appropriate. A fuzzy language L over Σ is defined by its membership function $\mu_L : \Sigma^\star \to \mathcal{L}$ where $\mathcal{L}$ is a structure that is a bit more complex then the two-element set $\{0, 1\}$ used in case of ordinary or crisp languages. A fuzzy automaton [fuzzy grammar] is a weighted automaton [weighted grammar] where the weights are taken from $\mathcal{L}$.

Originally [10], $\mathcal{L}$ was taken equal to the real closed interval $[0, 1]$, but nowadays we use a structure somewhat more complex [9, 12, 8] than the semirings usually encountered in the study of weighted automata. More precisely, $\mathcal{L}$ is a completely distributive complete lattice provided with an additional operator $\star$, i.e., $\mathcal{L} = (\mathcal{L}, \wedge, \vee, 0, 1, \star)$ where $(\mathcal{L}, \wedge, \vee, 0, 1)$ is a completely distributive complete lattice and $(\mathcal{L}, \star, 1)$ is a commutative monoid; these two substructures are coupled by distributivity laws, and $0 \star a = 0$ holds for all a in $\mathcal{L}$; cf. [1, 2].

Imperfections can be small ("tiny mistakes": weights are close to, but unequal to 1) or big ("capital blunders": weights are close to, but unequal to 0). Multiple errors in a derivation or computation should be accumulated, the result of which determines the quality (the degree of perfection) of the string to be generated or accepted ultimately.

The three operations of $\mathcal{L}$ reflect several aspects of the generation or computation process:

- The $\vee$-operation plays an important rôle in the presence of nondeterminism or ambiguity.
- We need the $\wedge$-operation for imposing regular restrictions, in particular for "intersection with the set of words over a terminal alphabet".
- The $\star$-operation is essential in describing the accumulation of errors properly ("Making an error twice is worse than making it once." "A long sequence of tiny mistakes may result in something that looks like a capital blunder.")

The $\vee$-operation is needed for another reason; viz. the occurrence of an error may be compensated by another error to obtain a perfectly correct string w in the language $L(G)$ or $L(A)$. In order to yield $\mu_{L(G)}(w) = 1$ or $\mu_{L(A)}(w) = 1$, we need to take the supremum over all possible derivations or computations, respectively, including the completely correct ones.

In this framework some properties of context-free [3], regular [2, 6] and Lindenmayer [1, 6] languages have been generalized to their "fuzzified" counterparts. For instance, context-free grammars augmented with rules that model grammatical errors give rise to modifications of well-known parsing algorithms [4] which, apart from their usual job, are able to deal with erroneous inputs as well.

Although many of these generalizations hold for arbitrary $\mathcal{L}$, some results have been only established for the case in which $\mathcal{L}$ is linearly ordered (of which the real closed interval $[0, 1]$

with operations min, max and multiplication is the most frequently used, practical example).

As an illustration we consider the problem of (in)dependency of operations on fuzzy languages. An operation o from a class C of operations on fuzzy languages is called *independent* if o cannot be expressed as a polynomial over $C - \{o\}$. A major tool in studying independency is the set of algebraic closure operators that correspond to the subsets of C. This set turns out to be a bounded partially ordered monoid, the structure of which (i.e., its Cayley table and its partial order) answers many (in)dependency questions. In determining the structure of this monoid, it is crucial to establish (i) defining relations, and (ii) inequalities. Now (i) is usually possible for arbitrary $\mathcal{L}$, whereas (ii) relies on translating a proper inclusion between families of crisp languages to a proper inclusion between the corresponding families of fuzzy languages. In this translation process the linear order of $\mathcal{L}$ is probably indispensable [5, 7].

On the other hand, when $\mathcal{L}$ is not linearly ordered —even in the simplest case (i.e., $\mathcal{L}$ is the four-element distributive lattice that is not a chain, while $\star$ coincides with $\wedge$)— unexpected matters will occur: there exists two fuzzy context-free languages L_1 and L_2 over Σ with their crisp parts, defined by $c(L_i) = \{w \in \Sigma^\star \mid \mu_{L_i}(w) = 1\}$, being empty (i.e., $c(L_i) = \varnothing$ for $i = 1, 2$), whereas the crisp part of their union $c(L_1 \cup L_2)$ is not context-free; cf. [11, 4].

1. P.R.J. Asveld, Controlled fuzzy parallel rewriting, pp. 49–70 in: Gh. Păun & A. Salomaa (eds.), *New Trends in Formal Languages — Control, Cooperation, and Combinatorics* (1997), Lect. Notes. in Comp. Sci. **1218**, Springer, Berlin, etc.
2. P.R.J. Asveld, The non-self-embedding property for generalized fuzzy context-free grammars, *Publ. Math. Debrecen* **54** *Suppl.* (1999) 553–573.
3. P.R.J. Asveld, Fuzzy context-free languages — Part I: Generalized fuzzy context-free grammars (2000), CTIT Tech. Rep. No. 00-03, Enschede, the Netherlands.
4. P.R.J. Asveld, Fuzzy context-free languages — Part II: Recognition and parsing algorithms (2000), CTIT Techn. Rep. No. 00-25, Enschede, the Netherlands.
5. P.R.J. Asveld, Some operators on families of fuzzy languages and their monoids, pp. 1–14 in: M. Ito, Gh. Păun & Sh. Yu (eds.), *Words, Semigroups, and Translations* (2001), World Scientific, Singapore.
6. P.R.J. Asveld, Algebraic aspects of families of fuzzy languages, *Theor. Comp. Sci.* (to appear).
7. P.R.J. Asveld, Independency of operations on fuzzy languages, [in preparation].
8. G. Gerla, Fuzzy grammars and recursively enumerable fuzzy languages, *Inform. Sci.* **60** (1992) 137–143.
9. H.H. Kim, M. Mizumoto, J. Toyoda & K. Tanaka, L-fuzzy grammars, *Inform. Sci.* **8** (1975) 123–140.
10. E.T. Lee & L.A. Zadeh, Note on fuzzy languages, *Inform. Sci.* **1** (1969) 421–434.
11. M. Mizumoto, J. Toyoda & K. Tanaka, Examples of formal grammars with weights, *Inform. Process. Lett.* **2** (1973) 74–78.
12. W. Wechler: *The Concept of Fuzziness in Automata and Language Theory* (1978), Akademie-Verlag, Berlin.

An extension theorem for grove theories

Stephen L. Bloom
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ 07030

Zoltan Ésik
Institute for Informatics
University of Szeged
Szeged, Hungary

The talk will first survey some known results on the theory of the fixed point operation in several settings, recall the notion of Conway and iteration theories, and, in particular, review the authors' "matrix extension theorem". One of the corollaries of this theorem is that when $\mathbf{Mat}_S$ is a Conway or iteration theory, for a semiring S, then so is $\mathbf{Mat}_R$, where $R = S\langle\langle X\rangle\rangle$ is the semiring of formal power series with coefficients in S in the noncommuting variables X.

Then, we will discuss some models of fixed points related to weighted tree automata: formal power series of trees, and synchronization trees. We will prove a new normal form theorem and a new extension theorem for "grove theories", which implies in particular that if S is a Conway semiring or iteration semiring, then for any ranked alphabet Σ, the theory of formal tree series over Σ with coefficients in S is a Conway theory, or an iteration theory.

Weak Equivalence for (max /+) Automata and Related Models

Peter Buchholz*
Fakultät für Informatik
TU Dresden
D-01062 Dresden, Germany
p.buchholz@inf.tu-dresden.de

Peter Kemper†
Informatik IV
Universität Dortmund
D-44221 Dortmund, Germany
kemper@ls4.cs.uni-dortmund.de

Observation congruence as defined by R. Milner in the context of the process algebra CCS [6] has shown to be a very useful notion of process equivalence that helps to understand and compare complex behaviors and that can be exploited in (semi-)automatic verification due to its congruence property. In this contribution, a similar notion of observation congruence called weak equivalence is proposed for a different class of automata where transitions are labeled with labels and weights from an idempotent semiring. As semirings we consider $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$, $\mathbb{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +)$ and their completed versions restricted to non-negative values $\overline{\mathbb{R}}_{\max} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \max, +)$, $\overline{\mathbb{R}}_{\min} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +)$ [5]. For notational convenience we denote these semirings as $\mathbb{K}_{\hat{+}} = \{\mathbb{K}, \hat{+}, \hat{\cdot}, \mathbb{O}, \mathbb{1}\}$ where $\mathbb{K}$ is the set of values, $\hat{+}$, $\hat{\cdot}$ are the addition and multiplication operation and $\mathbb{O}, \mathbb{1}$ are the zero and one element of the semiring [2].

Definition 1 *[$\mathbb{K}_{\hat{+}}$-automaton] A finite $\mathbb{K}_{\hat{+}}$-automaton over a finite alphabet $\mathcal{L}$ including symbol τ is a quadruple $\mathcal{A} = (\mathcal{S}, \alpha, T, \beta)$ where $\mathcal{S}$ is a finite set of states and α, T, β are maps $\alpha : \mathcal{S} \to \mathbb{K}$, $T : \mathcal{S} \times \mathcal{L} \times T \to \mathbb{K}$, $\beta : \mathcal{S} \to \mathbb{K}$, denoted as initial-, transition- and final-weights, respectively.*

Definition 2 *[Paths in $\mathbb{K}_{\hat{+}}$-automata] A path π of length n in a $\mathbb{K}_{\hat{+}}$-automaton is a sequence $s_0, l_1, s_1, \ldots, l_n, s_n$ with $s_i \in \mathbb{K}$, $l_j \in \mathcal{L}$, $\alpha(s_0) \neq \mathbb{O}$, $\beta(s_0) \neq \mathbb{O}$ and $T(s_{i-1}, l_i, s_i) \neq \mathbb{O}$. $\mathcal{P}(\mathcal{A})$ is the set of all paths of automaton $\mathcal{A}$ and $\mathcal{P}(\mathcal{A}, (l_1, \ldots, l_n))$ is the set of all paths observing transition labels $l_1, \ldots, l_n$. The weight of a path π is defined as*

$$w(\pi) = \alpha(s_0) \hat{\cdot} \left(\widehat{\prod}_{i=1}^{n} T(s_{i-1}, l_i, s_i) \right) \hat{\cdot} \beta(s_n) \tag{1}$$

and the weight of sequence $l_1, \ldots, l_n$ equals

$$w(l_1, \ldots, l_n) = \widehat{\sum}_{\pi \in \mathcal{P}(\mathcal{A}, (l_1, \ldots, l_n))} w(\pi) \tag{2}$$

As usual empty products are $\mathbb{1}$ and empty sums are $\mathbb{O}$. Transitions labeled with τ are not visible by an observer. Following the idea proposed in [6, 1] we can transform our original transition system into a new one where τ-labeled transitions are skipped. Let $\mathcal{P}(\mathcal{A}, s, s', l)$ be the set of all paths which start in state s, end in state s' and one transition is labeled with l, all other transitions are labeled with τ. For each $\mathbb{K}_{\hat{+}}$ automaton $\mathcal{A} = (\mathcal{S}, \alpha, T, \beta)$ we can define an automaton $\mathcal{A}^* = (\mathcal{S}, \alpha, T^*, \beta)$ over alphabet $\mathcal{L}$ where $T^*(s, l, s') = \widehat{\sum}_{\pi \in \mathcal{P}(\mathcal{A}, s, s', l)} \widehat{\prod}_{i=1}^{|\pi|} T(s_{i-1}, l_i, s_i)$. By using T^* instead of T in (1) we obtain new weight functions $w^*(\pi)$ and $w^*(l_1, \ldots, l_n)$, which describe the weights of observable paths and sequences, respectively.

Proposition 1 *If $\mathcal{A}$ contains no cycles of τ-labeled transitions, then $\mathcal{A}^*$ can be computed with an effort in $O(n^3)$ where $n = |\mathcal{S}|$ the number of states.*

*This research is partially supported by DFG, SFB 358.
†This research is partially supported by DFG, SFB 559.

The proposition follows from standard results in max /+ (e.g., [5]). For automata with positive weights, a computation of shortest paths between states in case of $\overline{\mathbb{R}}_{\min}$ and a computation of $\mathcal{A}^*$ for $\overline{\mathbb{R}}_{\max}$ is possible in $O(n^3)$ as well. For the latter, cycles in the graph are considered and yield weight ∞. In the general case, treatment of cycles makes computation of $\mathcal{A}^*$ more complex.

Definition 3 *[Weak equivalence $\mathbb{K}_{\widehat{+}}$ -automata] An equivalence relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ is a weak equivalence relation, iff for all $(s_1, s_2) \in \mathcal{R}$, all $l \in \mathcal{L}$ and all equivalence classes $\mathcal{R}[\tilde{s}]$,*

$$\alpha(s_1) = \alpha(s_2),\ \beta(s_1) = \beta(s_2),\ \text{and} \ \widehat{\sum}_{s' \in \mathcal{R}[\tilde{s}]} T^*(s_1, l, s') = \widehat{\sum}_{s' \in \mathcal{R}[\tilde{s}]} T^*(s_2, l, s').$$

Two states s_1 and s_2 are weakly equivalent denoted as $s_1 \approx s_2$, if a weak equivalence relation $\mathcal{R}$ with $(s_1, s_2) \in \mathcal{R}$ exists.

Two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are weakly equivalent denoted as $\mathcal{A}_1 \approx \mathcal{A}_2$, if a weak equivalence relation $\mathcal{R} \subseteq (\mathcal{S}_1 \cup \mathcal{S}_2) \times (\mathcal{S}_1 \cup \mathcal{S}_2)$ exists such that for each $s_1 \in \mathcal{S}_1$ with $\alpha(s_1) \neq \mathbb{O} \vee \beta(s_1) \neq \mathbb{O}$ some $s_2 \in \mathcal{S}_2$ with $(s_1, s_2) \in \mathcal{R}$ exists and for each $s_2 \in \mathcal{S}_2$ with $\alpha(s_2) \neq \mathbb{O} \vee \beta(s_2) \neq \mathbb{O}$ some $s_1 \in \mathcal{S}_1$ with $(s_2, s_1) \in \mathcal{R}$ exists.

The proofs of the following theorems will be presented in the detailed version of this paper.

Theorem 1 *If $\mathcal{A}_1 \approx \mathcal{A}_2$, then $w_1^*(\mathbf{l}) = w_2^*(\mathbf{l})$ for all $\mathbf{l} \in \mathcal{L}^*$ ($\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$), where $w_i^*(\mathbf{l})$ is the weight of sequence $l_1, \ldots, l_n$ in automaton i according to transition relation T_i^*.*

The notion of weak equivalence is of great value, if it is a congruence. In that case, automata can be replaced by weakly equivalent ones that have smaller sets of states to achieve minimal representations. We first introduce some composition operations and show afterwards the congruence property.

Definition 4 *[Composition of $\mathbb{K}_{\widehat{+}}$ -automata] Let $\mathcal{A}_1 = (\mathcal{S}_1, \alpha_1, T_1, \beta_1)$ and $\mathcal{A}_2 = (\mathcal{S}_2, \alpha_2, T_2, \beta_2)$ be two finite $\mathbb{K}_{\widehat{+}}$ -automata.*

1. *The direct sum of automata[1] is defined as an automaton $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2$ with $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$, $\alpha(s) = \alpha_1(s)$ if $s \in \mathcal{S}_1$ and $\alpha(s) = \alpha_2(s)$ otherwise, $\beta(s) = \beta_1(s)$ if $s \in \mathcal{S}_1$ and $\beta(s) = \beta_2(s)$ otherwise, and $T(s, l, s') = T_1(s, l, s')$ if $s, s' \in \mathcal{S}_1$, $T(s, l, s') = T_2(s, l, s')$ if $s, s' \in \mathcal{S}_2$, $T(s, l, s') = \mathbb{O}$ otherwise.*

2. *The direct product of automata is defined as an automaton $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2$ with $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$, $\alpha(s) = \alpha_1(s)$ if $s \in \mathcal{S}_1$ and $\widehat{\sum}_{s' \in \mathcal{S}_1} \alpha_1(s') \,\hat{\cdot}\, \beta_1(s') \,\hat{\cdot}\, \alpha_2(s)$ if $s \in \mathcal{S}_2$, $\beta(s) = \beta_2(s)$ for $s \in \mathcal{S}_2$, $\beta(s) = \mathbb{O}$ otherwise, and $T(s, l, s) = T_1(s, l, s')$ if $s, s' \in \mathcal{S}_1$, $T(s, l, s') = T_2(s, l, s')$ if $s, s' \in \mathcal{S}_2$, $T(s, l, s') = \widehat{\sum}_{s_1 \in \mathcal{S}_1} T(s, l, s_1) \,\hat{\cdot}\, \beta(s_1) \,\hat{\cdot}\, \alpha(s')$ if $s \in \mathcal{S}_1$, $s' \in \mathcal{S}_2$, $T(s, l, s') = \mathbb{O}$ otherwise.*

3. *The synchronized product[2] over a set $\mathcal{L}_c$ with $\tau \notin \mathcal{L}_c$ is defined as an automaton $\mathcal{A} = \mathcal{A}_1 ||_{\mathcal{L}_c} \mathcal{A}_2$ over a label set $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ and state space $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$. Let $s = (s_1, s_2)$ and $s' = ({s'}_1, {s'}_2)$,*

[1] The terms direct sum and product are borrowed from [4] where they are introduced for stochastic automata.

[2] The terms free and synchronous product are borrowed from [1]. The free product results from a synchronous product with $\mathcal{L}_c = \emptyset$.

$\alpha(s) = \alpha(s_1) \,\widehat{\cdot}\, \alpha(s_2)$, $\beta(s) = \beta(s_1) \,\widehat{\cdot}\, \beta(s_2)$, *and*

$$T(s,l,s') = \begin{cases} T_1(s_1,l,{s'}_1) & \textit{if } l \in \mathcal{L}_1 \backslash \{\mathcal{L}_c \cup \tau\} \textit{ and } s_2 = {s'}_2 \\ T_2(s_2,l,{s'}_2) & \textit{if } l \in \mathcal{L}_2 \backslash \{\mathcal{L}_c \cup \tau\} \textit{ and } s_1 = {s'}_1 \\ T_1(s_1,l,{s'}_1) & \textit{if } l = \tau \textit{ and } s_1 \neq {s'}_1, s_2 = {s'}_2 \\ T_2(s_2,l,{s'}_2) & \textit{if } l = \tau \textit{ and } s_1 = {s'}_1, s_2 \neq {s'}_2 \\ T_1(s_1,l,{s'}_1) \,\widehat{+}\, T_2(s_2,l,{s'}_2) & \textit{if } l = \tau \textit{ and } s = s' \\ T_1(s_1,l,{s'}_1) \,\widehat{\cdot}\, T_2(s_2,l,{s'}_2) & \textit{if } l \in \mathcal{L}_c \\ \mathbb{O} & \textit{otherwise} \end{cases}$$

Other composition operations can be adopted from process algebras like GPA, a process algebra with a semantics corresponding to weighted automata in the finite state case [3].

Theorem 2 *If $\mathcal{A}_1 \approx \mathcal{A}_2$ and $\mathcal{A}_3$ are finite $\mathbb{K}_{\widehat{+}}$-automata, then*

1. $\mathcal{A}_1 + \mathcal{A}_3 \approx \mathcal{A}_2 + \mathcal{A}_3$,
2. $\mathcal{A}_1 \cdot \mathcal{A}_3 \approx \mathcal{A}_2 \cdot \mathcal{A}_3$,
3. $\mathcal{A}_3 \cdot \mathcal{A}_1 \approx \mathcal{A}_3 \cdot \mathcal{A}_1$,
4. $\mathcal{A}_1 ||_{\mathcal{L}_c} \mathcal{A}_3 \approx \mathcal{A}_2 ||_{\mathcal{L}_c} \mathcal{A}_3$ *for any finite set $\mathcal{L}_c$ with $\tau \notin \mathcal{L}_c$.*

References

[1] A. Arnold. *Finite Transition systems.* Prentice Hall, 1994.

[2] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat. *Synchronization and Linearity.* John Wiley and Sons, 1992.

[3] P. Buchholz and P. Kemper. Quantifying the dynamic behavior of process algebras. In L. de Alfaro and S. Gilmore, editors, *Process Algebras and Probabilistic Methods*, LNCS 2165, pages 184–199. Springer, 2001.

[4] R. G. Bukharaev. *Theorie der stochastischen Automaten.* Teubner, 1995.

[5] S. Gaubert and Max Plus. Methods and applications of (max,+) linear algebra. Research Report 3088, INRIA, 1997.

[6] R. Milner. *Communication and concurrency.* Prentice Hall, 1989.

Weighted automata on infinite words

Manfred Droste[1] and Dietrich Kuske[2]

We consider weighted automata over $(\mathbb{R}_{-\infty}, \max, +)$. The first new idea we deal with is that later costs incured by an execution of the system are not as pressing as immediate costs. Therefore, we introduce a nonnegative scaling factor q such that the nth step in a path does not incur the cost x, but the cost $q^n \cdot x$. This motivates a variation of the Cauchy product of formal power series now given by

$$(S +_q T, w) = \max_{uv=w}((S,u) + q^{|u|}(T,v))$$

for all finite words w and formal power series S and T. Together with the pointwise maximum, we obtain a semiring $(\mathbb{R}_{-\infty}^{\Sigma^*}, \max, +_q)$ which, for $q = 1$, is the semiring of formal power series. If $q \neq 1$, there is no base semiring K such that this structure is the semiring of formal power series over K. Nevertheless, it is possible to prove a Schützenberger-type characterization of the behaviors of weighted automata in the presence of a scaling factor $q \geq 0$.

We then consider the behavior at infinite words. The cost of an infinite path is the *limes superior* of the costs of its finite prefixes. The cost of an infinite word w is then the supremum of all the costs of paths labeled by this word w. This resembles the Büchi-acceptance condition for infinite words. Our main result in this context is a characterization of the behaviors of weighted Büchi-automata in the spirit of Schützenberger's result; the new operation extending the rational operations is the ω-power of a formal power series given by

$$(S^\omega, w) = \sup_{\substack{uv=w \\ u\in\Sigma^*}}((S,u) + q^{|u|}(T,v))$$

for any infinite word w.

The behavior of a weighted automaton at infinite words was also considered by Culik and Karhumäki. Although they work in the semiring $(\mathbb{R}, +, \cdot)$, their definition can readily be translated into our setting (for $q = 1$, i.e., without scaling): The cost of an infinite word is the *limes superior* of the costs of its finite prefixes. The cost of a finite word is then (as usual) determined by considering all possible paths. This definition is not as powerful as ours in the sense that there are functions generated by weighted automata according to our definition that cannot be generated according to Culik and Karhumäki's definition. We can show that for "almost deterministic" automata, the two definitions coincide: a weighted automaton is almost deterministic if any word gives rise to finitely many labels, only. This equivalence result resembles the well known fact $L^\omega(\mathcal{A}) = \overrightarrow{L^*(\mathcal{A})}$ for deterministic Büchi-automata $\mathcal{A}$.

[1]Institut ür Algebra, TU Dresden, D-01062 Dresden, Germany, `email:droste@math.tu-dresden.de`

[2]Department of Mathematics and Computer Science, University of Leicester, Leicester, LE1 7RH, UK, `email:d.kuske@mcs.le.ac.uk`

Equational theories of exotic semirings, an overview

Zoltan Ésik
Institute for Informatics
University of Szeged Szeged, Hungary

Exotic semirings are semirings whose underlying carrier set is some subset of the set of real numbers equipped with binary operations of minimum or maximum as sum, and addition as product. Two prime examples of such structures are the $(\max, +)$ *semiring* and the *tropical semiring*. It is shown that none of the exotic semirings commonly considered in the literature has a finite basis for its equations, and that similar results hold for the commutative idempotent weak semirings that underlie them. For each of these commutative idempotent weak semirings, the talk offers characterizations of the equations that hold in them, explicit descriptions of the free algebras in the varieties they generate, and relative axiomatization results. The results were obtained in collaboration with Luca Aceto and Anna Ingolfsdottir.

Generalizations of tree transducers to tree series transducers

Zoltán Fülöp
Institute for Informatics
University of Szeged
Szeged, Hungary

Heiko Vogler
Department of Computer Science
Dresden University of Technology
Dresden, Germany

Following the line of generalizing finite automata to automata with costs (or: multiplicity), [Sch62], [SS78], [KS86], tree transducers [Eng75] can be generalized to tree series transducers [EFV01] in such a way that, while a tree transducer computes a tree transformation, a tree series transducer computes a tree transformation together with the cost of its computation. The cost is computed in a semiring.

There are more possible ways of the generalization [Kui99], [EFV01], [FV02] however, the results which can be obtained are very sensitive of even small changes in the generalized model and also of the semiring in which the cost is computed.

In the talk we define tree series transducer models and show that some theorems concerning the classical tree transducers carry over to tree series transducers, while others only in restricted forms.

References

[Eng75] J. Engelfriet, Bottom–up and top–down tree transformations — a comparison, *Math. Systems Theory*, 9(3):198–231, 1975.

[EFV01] J. Engelfriet, Z. Fülöp and H. Vogler, Bottom-up and Top-down Tree Series Transformations, *Journal of Automata, Languages and Combinatorics*, accepted for publication.

[FV02] Z. Fülöp and H. Vogler, Tree Series Transformations that Respects Copying, manuscript.

[Kui99] W. Kuich, Tree transducers and formal tree series, *Acta Cybernetica*, 14, 135-149 1999.

[Sch62] M. P. Schützenberger, Certain elementary families of automata, In *Proceedings of Symposium on Mathematical Theory of Automata*, pages 139–153, 1962. Polytechnic Institute of Broocklyn.

[SS78] A. Salomaa and M. Soittola, *Automata-Theoretic Aspects of Formal Power Series.* Texts and Monographs in Computer Science, Springer Verlag, 1978.

[KS86] W. Kuich and A. Salomaa, *Semirings, Automata, Languages*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1986.

Max-plus linear dynamical systems: theory and applications

Stéphane Gaubert
INRIA, Domaine de Voluceau,
78153 Le Chesnay Cédex, FRANCE
Stephane.Gaubert@inria.fr
http://amadeus.inria.fr/gaubert

One of the key results in max-plus algebra is the max-plus spectral theorem, which describes precisely the asymptotic behavior of max-plus linear dynamical systems. The goal of this lecture is to present the max-plus spectral theorem, together with its applications to dynamic programming and to the performance analysis of discrete event systems. We shall also present some nonlinear extensions, as well as applications of theses ideas to max-plus rational series and max-plus linear semigroups.

Part 1. Max-plus dynamical systems: examples Models of discrete event systems and dynamic programming problems leading to max-plus linear systems. Dater representation of timed event graphs. Heaps of pieces models. Examples from flowshop and jobshop scheduling. Nonlinear extensions: monotone additively homogeneous maps as dynamic programming operators. Game examples.

Part 2. Max-plus spectral theorem Analogy with the Perron-Frobenius theorem. Characterization of the eigenvalues. Characterization of the eigenspace. Cyclicity theorem. Exercises (application to some of the previous examples.)

Part 3. Spectral theory of monotone homogeneous maps, algorithms, and related topics Cycle time and fixed point sets of monotone homogeneous maps. Policy iteration algorithms. Application: fast algorithm to compute max-plus spectral elements. Max-plus toolbox of scilab. Solving $Ax = Bx$. Computing invariant spaces.

Part 4. Max-plus rational series and max-plus linear semigroups Rational series version of the max-plus spectral theorem. Deciding the finiteness of finitely generated max-plus linear semigroups (Simon's theorem). Undecidability results (Krob's theorem). Projective max-plus linear semigroups. Max-plus Lyapunov exponents and asymptotic analysis of max-plus automata.

References

[ABG01] M. Akian, R. Bapat, and S. Gaubert. Generic asymptotics of eigenvalues using min-plus algebra. In *Proceedings of the Satellite Workshop on Max-Plus Algebras, IFAC SSSC'01*, Praha, 2001. Elsevier.

[AG01] M. Akian and S. Gaubert. Spectral theorem for convex monotone homogeneous maps, and ergodic control. 2001. Accepted for publication in *Nonlinear Analysis Theory, & Methods*, available on arXiv:math.SP/0110108, also INRIA Research Report 4273.

[Bap98] R. B. Bapat. A max version of the Perron-Frobenius theorem. *Linear Algebra Appl.*, 275/276:3–18, 1998.

[BCOQ92] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat. *Synchronization and Linearity.* Wiley, 1992. Available from http://www-rocq.inria.fr/scilab/cohen/SED/book-online.html.

[CG79] R. Cuninghame-Green. *Minimax Algebra.* Number 166 in Lecture notes in Economics and Mathematical Systems. Springer, 1979.

[CGQ99] G. Cohen, S. Gaubert, and J. Quadrat. Max-plus algebra and system theory: where we are and where to go now. *Annual Reviews in Control*, 23:207–219, 1999.

[Cho77] C. Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que fonctions rationnelles. *Theoret. Comput. Sci.*, 5:325–337, 1977.

[Gau95] S. Gaubert. Performance evaluation of (max,+) automata. *IEEE Trans. on Automatic Control*, 40(12):2014–2025, Dec 1995.

[Gau00] S. Gaubert. Introduction aux systèmes dynamiques à événements discrets. Polycopié de cours ENSTA et ENSMP/DEA ATS orsay, 1992–2000.

[GG98] S. Gaubert and J. Gunawardena. A non-linear hierarchy for discrete event dynamical systems. In *Proc. of the Fourth Workshop on Discrete Event Systems (WODES98)*, Cagliari, Italy, 1998. IEE.

[GG01] S. Gaubert and J. Gunawardena. The Perron-Frobenius theorem for homogeneous, monotone functions. Hewlett-Packard Technical Report 2001-12, HPL-BRIMS, 2001. Also arXiv:math.FA/0105091, submitted.

[GH00] S. Gaubert and D. Hong. Series expansions of lyapunov exponents and forgetful monoids. 2000. Accepted for publication in IJAC, available as INRIA Research Report 3971.

[GM77] M. Gondran and M. Minoux. Valeurs propres et vecteurs propres dans les dioïdes et leur interprétation en théorie des graphes. *EDF, Bulletin de la Direction des Etudes et Recherches, Serie C, Mathématiques Informatique*, 2:25–41, 1977.

[GM99] S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Automat. Control*, 44(4):683–697, 1999.

[GP97] S. Gaubert and M. Plus. Methods and applications of (max,+) linear algebra. In R. Reischuk and M. Morvan, editors, *STACS'97*, number 1200 in LNCS, Lübeck, March 1997. Springer.

[Kro93] D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Int. J. of Algebra and Comput.*, 3, 1993.

[Len01] C. Lenté. *Analyse max-plus de problèmes d'ordonnancement de type flowshop.* Thèse, Université de Tours, November 2001.

[Leu91] H. Leung. Limitedness theorem on finite automata with distance function: an algebraic proof. *Theoret. Comput. Sci*, 81:137–145, 1991.

[Plu99] M. Plus. The max-plus algebra toolbox of scilab. Freely available from http://www-rocq.inria.fr/scilab/, contrib section, 1999.

[Rom67] I. V. Romanovskiĭ. Optimization of stationary control of discrete deterministic process in dynamic programming. *Kibernetika*, 3(2):66–78, 1967.

[Sim94] I. Simon. On semigroups of matrices over the tropical semiring. *Theor. Infor. and Appl.*, 28(3-4):277–294, 1994.

[Vor67] N. Vorobyev. Extremal algebra of positive matrices. *Elektron. Informationsverarbeitung und Kybernetik*, 3, 1967. in russian.

Control and Optimization of Public Transportation Networks via (max,+) Algebra

Bernd Heidergott
Department of Mathematics and Computing Science
TU Eindhoven, the Netherlands

The (max,+) algebra provides a mathematical framework most suitable for the analysis of the time evolution of DESs that involve synchronization constrains. A generic example for DESs where synchronization plays a crucial role are public transportation networks (operating according to a timetable). In public transportation networks, such as railway or subway networks, synchronization constrains follow from (i) the planned departure times and (ii) the demand that trains should connect. In this lecture, we will provide an overview on recent results of applications of (max,+)-algebra to performance analysis of public transportation networks. First, we will give a simple characterization of the type of problems that can actually be modeled within the (max,+) framework (and thereby will explain why "trains" are a generic example of (max,+)-linear systems). Secondly, we discuss the application of (max,+)-techniques to capacity planning problems and give examples. Thirdly, a simple scheme for calculating propagation of delays in networks is presented. We conclude this lecture by posing some open problems.

Weighted Finite Automata in Image Compression and Manipulation

Juhani Karhumäki
Department of Mathematics
and TUCS
University of Turku
20014 Turku, Finland

Abstract: Finite automata with weights, or more precisely rational languages they define, can be used to describe natural two-dimensional images. As a consequence, the automaton can be seen as the compressed representation of the image it describes. An interesting point here is that many transformations of images can be done directly on the level of their compressed representations. This and other related questions form the topic of this survey talk.

Automatic Algebraic Systems

Bakhadyr Khoussainov
Department of Computer Science
The University of Auckland
Auckland, New Zealand

In this talk we introduce and motivate the study of automatic algebraic structures. We discuss and survey recent developments in this area. In particular, we will talk about automatic graphs, linear orders, and equivalence relations. We end the talk by discussing some open questions and possible future work in the area.

An axiomatic approach to weighted automata

Dexter Kozen
Cornell University
Ithaca, USA

In this talk I will discuss an axiomatic approach to weighted automata from the point of view of Kleene algebra. I will discuss such issues as composition of automata and complexity issues from this viewpoint. Our starting point is the characterization of weighted automata as matrices over an algebra of formal power series.

Formal Power Series and Formal Tree Series

Werner Kuich
Vienna University of Technology
Vienna, Austria

The purpose of these lectures is to develop some results on formal languages and automata, and formal tree languages and tree automata by an algebraic treatment using semirings, formal power series, formal tree series and matrices. The use of these mathematical constructs makes definitions, constructions and proofs more satisfactory from the mathematical point of view than the customary ones.

Contents:

1. Introduction to Semiring Theory

Continuous semirings, Formal power series, Matrices, Star - Sum - Equality, Matrix - Star - Equality.

2. Formal Power Series

Finite automata, Kleene's Theorem, Constructions for finite automata, Algebraic systems, Normal forms, Pushdown automata, Triple construction, Rational transducers, Abstract families of power series.

3. Formal Tree Series

Distributive algebras, Substitutions, Tree automata, Systems of equations, Recognizable tree series, Kleene Theorem, Tree series transducers, Abstract families of tree series, Applications to language theory.

Literature: W. Kuich, Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata. Handbook of Formal Languages (G. Rozenberg, A. Salomaa, Eds.), Springer 1997, 609 - 677.

Derivatives of rational expressions with multiplicity

Sylvain Lombardy Jacques Sakarovitch *

Abstract

The purpose of this work is to generalize the definition and constructions, due to V. Antimirov ([1]), of the so-called *partial derivatives* — and that we shall call here *derived terms* — from rational expressions and languages to rational expressions and languages *with multiplicity.* As it turns out, this is another example where taking multiplicities into account and considering series rather than languages yield simplification and better understanding of constructions and results on languages.

In 1964, J. Brzozowski defined the *derivatives* of a rational expression ([2]). He showed that, *modulo* the axioms of associativity, commutativity, and idempotency of the addition (on the set of languages) — the ACI properties — the set of derivatives of a given expression is finite, yielding both a new proof for (one direction of) Kleene's Theorem and an algorithm turning an expression into a deterministic finite automaton.

This problem (of turning an expression into a finite automaton) has attracted much attention since the beginning of the theory ([5]) and is an area of active research since then. In 1995, V. Antimirov made a fundamental contribution by defining the "*partial derivatives*" of an expression ([1]). Using his own words, "the idea behind [his] construction is that it allows to take into account the ACI-properties of only those occurrences of "+" in [rational] terms which appear at the very upper level".

This construction has a number of outcomes: the number of partial derivatives is not only finite but " small" : smaller than or equal to the number of occurences of letters in the expression; they are easier to compute than the classical derivatives and they yield a *non-deterministic* finite automaton with (almost) the same number of states; finally, the subset construction applied to that automaton gives back the deterministic one computed by Brzozowski's algorithm. The computation of Antimirov's automaton has been made really efficient by Champarnaud and Ziadi [4].

What is presented here is at the same time a *formalization* and a *generalization* of Antimirov's constructio to rational expressions with multiplicity in a semiring $\mathbb{K}$, or $\mathbb{K}$-expressions.

*Laboratoire Traitement et Communication de l'Information (C. N. R. S. URA 820), E. N. S. T., Paris.

Let us note that Caron and Flouret [3] have dealt with expressions with multiplicity to generalize the Glushkov construction, yielding a $\mathbb{K}$-automaton that plays the same role with respect to ours as the so-called *position automaton* do with respect to Antimirov's automaton in the Boolean case.

We define the derivation of a $\mathbb{K}$-expression with respect to a letter and then to a word. The main feature of our definition, that indeed realizes Antimirov's main idea, is that the result of the derivation of a $\mathbb{K}$-expression is not a $\mathbb{K}$-expression anymore but a linear combination of $\mathbb{K}$-expressions with coefficients in $\mathbb{K}$.

The generalization of Antimirov's results is then straightforward and gives our main result. For any $\mathbb{K}$-expression E, there exists a finite set of $\mathbb{K}$-expressions, called the *derived terms* of E, such that the derivation of E with respect to any word is a linear combination with coefficients in $\mathbb{K}$ of derived terms. The number of derived terms is smaller than or equal to the *litteral length* of E.

An automaton with multiplicity, and which recognizes the series denoted by E, can then be built, the states of which are the derived terms. In particular, let us note that this technique overcomes the problem that the addition is not idempotent anymore, and that Brzozowski's theorem does not hold in that setting.

A direct consequence, or corollary, of these results is that a rational series belongs to a finitely generated $\mathbb{K}$-module which is *stable* by derivatives (or residuals); but it should be stressed that this new proof of a well-known (and fundamental) result is *not* the main account, or purpose of this paper. Rather, what we show here is that, when a rational series s is defined by an expression E, a *symbolic computation* performed on E *effectively* gives —through their expressions— a finite set of generators of the stable module that contains this series s.

References

[1] V. Antimirov, Partial derivatives of regular expressions and finite automaton constructions. *Theoret. Comput. Sci. 155* (1996), 291–319.

[2] J. A. Brzozowski, Derivatives of regular expressions. *J. Assoc. Comput. Mach. 11* (1964), 481–494.

[3] P. Caron and M. Flouret, Glushkov construction for multiplicities. *Pre-Proceedings of CIAA'00*, M. Daley, M. Eramian and S. Yu, eds, Univ. of Western Ontario, (2000), 52–61.

[4] J.-M. Champarnaud and D. Ziadi, New finite automaton constructions based on canonical derivatives. *Pre-Proceedings of CIAA'00*, M. Daley, M. Eramian and S. Yu, eds, Univ. of Western Ontario, (2000), 36–43.

[5] V. Glushkov, The abstract theory of automata. *Russian Mathematical Surveys 16* (1961), 1–53.

Weighted Grammars and Automata with Maximum Interpretation

Carlos Martín-Vide[1] Victor Mitrana[2] Ralf Stiebe[3]

[1]Research Group in Mathematical Linguistics, Rovira i Virgili University,
Pça. Imperial Tárraco 1, 43005 Tarragona, Spain.
cmv@correu.urv.es

[2]Faculty of Mathematics, University of Bucharest,
Str. Academiei 14, 70109, Bucharest, Romania.
mitrana@funinf.cs.unibuc.ro

[3]Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg,
Postfach 4120, 39016 Magdeburg, Germany.
stiebe@iws.cs.uni-magdeburg.de

1 Introduction

Weighted grammars/automata are important generalizations of context-free grammars/finite automata widely and successfully used in many applications like text and speech processing [1, 2, 3].

In a very general approach, each derivation rule in a weighted grammar or transition in a weighted finite automaton has a weight which is an element of a semiring. An additional information is associated with each word, namely the sum of the leftmost derivation/accepting path-weights, the weight of a derivation/accepting path being the product of the weights of the applied rules. A weighted grammar defines a formal power set; if the semiring is ordered, it is natural to define a language with regard to a threshold t as the set of all words whose weight is at least t.

In this paper, we discuss weighted grammars and automata over the so-called *completed* $(\max, +)$ *semiring* $\overline{\mathbb{R}}_{\max} = (\mathbb{R} \cup \{-\infty, \infty\}, \max, +, -\infty, 0)$, and its sub-semirings $\overline{\mathbb{Z}}_{\max} = (\mathbb{Z} \cup \{-\infty, \infty\}, \max, +, -\infty, 0)$, and $\mathbb{Q}_{+}^{\max} = (\mathbb{Q}_{+} \cup \{0, \infty\}, \max, \cdot, 0, 1)$. The last approach was proposed already in 1969 by Salomaa [4] as *weighted grammars with maximum interpretation*. Here, the weight of a word w is the supremum of the weights of all derivations of w. A similar concept is that of a *valence grammar* over $\mathbb{Z} = (\mathbb{Z}, +, 0)$, where a word belongs to the language iff there is a derivation of weight 0.

In this paper, we discuss the theory of language families generated by weighted grammars and compare them with the families generated by valence grammars.

We use the following notations. The families of languages generated by weighted context-free grammars, weighted finite automata and weighted deterministic finite automata over $A \in \{\overline{\mathbb{R}}_{\max}, \overline{\mathbb{Z}}_{\max}, \mathbb{Q}_{+}^{\max}\}$, is denoted by $\mathcal{L}(\mathrm{wm}, \mathrm{CF}, A)$, $\mathcal{L}(\mathrm{wm}, \mathrm{REG}, A)$, and $\mathcal{L}(\mathrm{wm}, \mathrm{DREG}, A)$, respectively. The family of languages generated by valence grammars of type $X \in \{\mathrm{REG}, \mathrm{CF}, \mathrm{DREG}\}$ over $\mathbb{Z}$ is denoted by $\mathcal{L}(\mathrm{Val}, X, \mathbb{Z})$.

2 Results

Theorem 1 [Computational power]

1. For $X \in \{\mathrm{CF}, \mathrm{REG}, \mathrm{DREG}\}$, the families $\mathcal{L}(\mathrm{wm}, X, \overline{\mathbb{R}}_{\max})$ and $\mathcal{L}(\mathrm{Val}, X, \mathbb{Z})$ are incomparable.
2. For $X \in \{\mathrm{CF}, \mathrm{REG}\}$, $\mathcal{L}(\mathrm{wm}, X, \overline{\mathbb{Z}}_{\max}) \subset \mathcal{L}(\mathrm{Val}, X, \mathbb{Z})$.
3. For $A \in \{\overline{\mathbb{R}}_{\max}, \overline{\mathbb{Z}}_{\max}, \mathbb{Q}_{+}^{\max}\}$, $\mathcal{L}(\mathrm{wm}, \mathrm{DREG}, A) \subset \mathcal{L}(\mathrm{wm}, \mathrm{REG}, A) \subset \mathcal{L}(\mathrm{wm}, \mathrm{CF}, A)$.

Theorem 2 [Closure properties]

1. For $A \in \{\overline{\mathbb{R}}_{\max}, \overline{\mathbb{Z}}_{\max}, \mathbb{Q}_{+}^{\max}\}$, $\mathcal{L}(\mathrm{wm}, \mathrm{CF}, A)$ and $\mathcal{L}(\mathrm{wm}, \mathrm{REG}, A)$ are closed under rational transductions and under union, but fail to be closed under concatenation, intersection, complementation, and Kleene closure.
2. For $A \in \{\overline{\mathbb{R}}_{\max}, \overline{\mathbb{Z}}_{\max}, \mathbb{Q}_{+}^{\max}\}$, $\mathcal{L}(\mathrm{wm}, \mathrm{DREG}, A)$ is closed under intersection with regular sets and inverse homomorphisms, but fails to be closed under complementation, concatenation, union, homomorphisms, and Kleene closure.

Theorem 3 [Decidability properties]

1. The membership problem can be decided in time $O(n^4 \log n \log\log n)$ for weighted context-free grammars over $\mathbb{Q}_{+}^{\max}$.
2. The emptiness and the finiteness problems are decidable for weighted context-free grammars over $\mathbb{Q}_{+}^{\max}$.
3. The inclusion and equivalence problems are decidable for weighted deterministic finite automata over $\mathbb{Q}_{+}^{\max}$.

References

[1] C. Cortes, M. Mohri. Context-Free Recognition with Weighted Automata. *Grammars*, 3(2-3):133–150, 2000.

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

[3] M. Mohri, F.C.N. Pereira, M. Riley. The Design Principles of a Weighted Finite-State Transducer Library. *Theoretical Computer Science*, 231(1): 17–31, 2000.

[4] A. Salomaa. Probabilistic and weighted grammars. *Information and Control*, 15:529–544, 1969.

Weighted Automata Algorithms

Mehryar Mohri
AT&T Labs - Research
Florham Park, New Jersey

Finite automata have been used to create efficient Unix command tools such as grep or to build some of the components of a compiler for programming languages. But the high variability of various data sources in modern information systems and the use of complex statistical techniques to model them require the use of more general devices: Weighted Automata.

Weighted automata are now used in a variety of applications ranging from computational biology to speech and image processing. The weights are used in those applications to account for the variability of the data and to rank alternative hypotheses.

General and efficient weighted automata algorithms can be used to create, modify, and optimize weighted automata with hundreds of millions of states and transitions in some of these applications. We describe and analyze some of these algorithms including the composition, epsilon-removal, determinization, minimization, and synchronization of weighted automata.

Max-Plus Algebra and its Applications to Railway Systems

Geert Jan Olsder
Faculty of Information Technology and Systems
Delft University of Technology
P.O.Box 5031, 2600 GA Delft, the Netherlands
E-mail: g.j.olsder@math.tudelft.nl

One of the key features that characterize Discrete Event Systems (DESs) is that their dynamics are *event-driven* as opposed to *time-driven*, i.e., the behavior of a DES is governed only by occurrences of different types of events over time rather than by ticks of a clock. We will concentrate on modelling issues of DESs within the max-plus algebra setting. The main application to be discussed is the design of timetables for trains on an existing net and related questions with respect to capacity issues, propagation of delays. A discrete event is the departure of a train from a station. These events are linked to each other. Obviously a train cannot arrive before its departure time at the previous station to which the travelling time has been added. Trains at stations must also wait for each other in order to allow changeovers of passengers.

We will briefly dwell upon some other theoretical extensions such as

- relation with graph theory and Petri nets;
- numerical procedures, especially important when the dimension of the state is large (say > 1000);
- imbedding in min-max-plus systems;
- continuous flow systems with synchronization constraints;

With respect to applications, the Dutch railway system will be considered. Another application to be dealt with briefly is the design of the high speed connection between Amsterdam and the Belgian/Dutch border as part of the high speed line between Amsterdam and Paris (known as HSL South).

Families of semilinear formal power series

Ion Petre
Turku Centre for Computer Science
Turku, Finland

The semilinearity is a central notion in Automata Theory that only recently has been considered in details for formal power series. It is well known that the semilinearity can be defined for commutative monoids in at least two equivalent ways. It turns out however that for formal power series those definitions are not equivalent. We thus consider two families of semilinear formal power series in commuting variables, both natural extensions to series of the classical notion of semilinearity. We discuss the relationships among these two families of semilinear series and the algebraic, rational, and recognizable power series in commuting variables. We also investigate the behaviour of the two families of semilinear series under rational operations, Hadamard product, and difference. The fact that the closure properties of the semilinear sets of words and those of the semilinear sets of vectors of nonnegative integers, under the corresponding operations, extend only in part to formal power series and in general, not to both families of semilinear formal power series, gives a new insight on the notion of semilinearity.

Coinductive Counting with Weighted Automata

J.J.M.M. Rutten
CWI & Vrije Universiteit, Amsterdam, The Netherlands
Email: `janr@cwi.nl`, URL: `www.cwi.nl/~janr`

Based on the presence of a final coalgebra structure on the set of streams (infinite sequences of real numbers), [Rut01b] presents a coinductive calculus of streams together with applications to various problems in mathematics. Here we apply stream calculus to basic counting problems from enumerative combinatorics, using weighted automata (over one formal variable and with coefficients in the reals) to represent streams. The notion of *stream derivative*, which plays a key role in stream calculus, is used to go from streams to (generally infinite) weighted automata, and vice versa. Our counting method consists of three steps:

1. The objects to be counted are enumerated in the form of an infinite, tree-shaped weighted automaton, usually in such a way that nodes at level n correspond to an object consisting of n basic ingredients (e.g., a permutation of the set $\{1, \ldots, n\}$, or a word consisting of n left and right brackets).

2. Using the quantitative notion of *stream bisimulation* from stream calculus, the weighted automaton is simplified by identifying bisimilar states while keeping track of, that is, counting, multiplicities. The resulting automaton is generally much 'smaller' (but often still infinite) and much better structured.

3. Exploiting the well-structuredness of the obtained automaton, an expression in stream calculus is calculated (similar to that of a generating function), representing the stream of counts of all objects consisting of 0, 1, and so on basic building ingredients.

This elementary method, which can be summarized by: 'enumerate what you want to count and next reason formally in the calculus of weighted automata and streams', is surprisingly general and can deal with most of the at least basic examples of enumerative combinatorics, such as counting surjections, compositions, well-bracketed words, lattice paths, many kinds of permutations, set partitions, and the like.

Independent of the above application, stream calculus seems to make a contribution to the general theory of weighted automata as well, namely, the use of (quantitative) stream bisimulation and coinduction for reasoning about potentially infinite (and in the full version of the story also multi-variate) weighted automata and the streams they represent. As an illustration, consider the following two automata (states that are underlined have weight 1, the others have weight 0):

$$\begin{tikzcd} \underline{q_0} \arrow[r, "1"] & \underline{q_1} \arrow[r, "2"] & \underline{q_2} \arrow[r, "3"] & \cdots \end{tikzcd} \qquad \begin{tikzcd} \underline{r_0} \arrow[loop above, "1"] \arrow[r, bend left, "1"] & r_1 \arrow[l, bend left, "1"] \arrow[loop above, "3"] \arrow[r, bend left, "2"] & r_2 \arrow[l, bend left, "2"] \arrow[loop above, "5"] \arrow[r, bend left, "3"] & \cdots \arrow[l, bend left, "3"] \end{tikzcd}$$

Using coinduction, one can prove that both q_0 and r_0 represent the same stream $(0!, 1!, 2!, \ldots)$.

For an extended version of the present abstract, see [Rut01a].

References

[Rut01a] J.J.M.M. Rutten. Coinductive counting: bisimulation in enumerative combinatorics (extended abstract). Report SEN-R0129, CWI, 2001. Available at URL: www.cwi.nl.

[Rut01b] J.J.M.M. Rutten. Elements of stream calculus (an extensive exercise in coinduction). Report SEN-R0120, CWI, 2001. Available at URL: www.cwi.nl. Also in the Proceedings of MFPS 17, S. Brookes and M. Mislove (Eds.), ENTCS Vol. 45, Elsevier, 2001.

Computing Real Functions with Weighted Finite Automata
— Abstract —

Holger Schulz
Universität Gesamthochschule Siegen
Mathematik · Theoretische Informatik
Hölderlinstr. 3 · D-57068 Siegen
kuddel@informatik.uni-siegen.de

Already in [7] the question is discussed which real numbers and functions can be computed by a machine. Since then a large number of models, varying from machine models [8] to domain theoretic [2] and axiomatic approaches [4], to compute real functions has been developed. Nowadays most of them are known to be equivalent.

All of them have in common that the notion of computability depends on how the real numbers are represented. The ordinary digit representation is known to be not very useful. So other represenatations, e.g. negative digit and interval representation, are used.

Studying subclasses of computable real functions can be done by studying machine models and their computational power. The aspects of real number computations have to be reminded when we derive those machine models from the classical ones.

In [1] the use of WFAs to compute real functions is examined. Since that paper studies the theoretical background of image compression with WFAs it has a specific point of view. That leads to some disadvantages from the view of real number computability.

One point is the use of the common binary digit represenatation and the choise of standard representations for real numbers with multiple representations. This makes WFAs compute discontinuous functions. In [8] and equivalent approaches it is a central result, that computable real functions must be continuous.

Another problem in [1] is, that real numbers are chosen as weights for the WFAs. But the set of real numbers is not countable. We cannot represent real numbers as entities.

We will consider more adequate weights like rational numbers, dyadic and rational intervals and representations to study the WFAs in the context of the known approaches to real number computability.

References

[1] Karel Culik and Juhani Karhumäki. Finite automata computing real functions. *SIAM Journal for Computing*, 23 No. 4:789–814, 1994.

[2] M.H. Escardó. PCF extended with real numbers. *Theoretical Computer Science*, 162(1):79–115, August 1996.

[3] Michal Konecny. Real functions computable by finite transducers using affine ifs representations. Technical Report CSR-00-10, University of Birmingham, School of Computer Science, July 2000.

[4] M.B. Pour-El and I. Richards. *Computability in Analysis and Physics.* Springer Verlag, Berlin, 1989.

[5] Holger Schulz. Some aspects of computing real functions with weighted finite automata. Technical report, `www.informatik.uni-siegen.de/~kuddel`, 2002.

[6] Holger Schulz and Dieter Spreen. On the equivalence of some approaches to computability on the real line. In Klaus Keimel etal., editor, *Domains and Processes, Proc. 1st Intern. Symp. on Domain Theory, Shanghai, China, 1999*, 2000.

[7] Alan Turing. On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 42:230–265, 1937.

[8] Klaus Weihrauch. *Computable Analysis: an Introduction.* Texts in theoretical computer science. Springer Berlin, 2000.

Weighted finite automata and Metrics in Cantor space

Ludwig Staiger[*]
Martin-Luther-Universität Halle-Wittenberg,
Institut für Informatik,
D–06099 Halle

Weighted finite automata are used to describe fractal images (cf. [CD93, CK93]). In particular, they play some rôle in the computation of the contraction coefficients for so-called Multiple Recursive Function Systems (MRFS, they are a combination of finite automata with iterated function systems). This is explained in more detail in [FS01].
Here in a finite automaton $\mathfrak{A} = [X, (0, \infty), S, s_0, f, g]$ with output monoid $((0, \infty), \cdot)$ to each transition the contraction coefficient of the mapping φ_x corresponding to the input letter $x \in X$ is assigned. Consequently, the resulting output $g(s, w)$ is an upper bound to the contraction coefficient of the mapping $\varphi_w := \varphi_{x_1} \circ \cdots \circ \varphi_{x_n}$ $(w = x_1 \cdots x_2)$.
In case the contraction coefficients of φ_w for $w \to \xi \in X^\omega$ converge to zero the corresponding MRFS "draws a point for $\xi \in X^\omega$". The set of all such ξ can be described topologically by a suitable topology (depending on the automaton $\mathfrak{A}$ (cf. [FS01]).
Another kind of topology on X^ω are the U-δ-topologies introduced in [St87] (cf. also [DN92]). Here the distance between two ω-words $\xi, \eta \in X^\omega$ depends of the number of their common prefixes in the given language $U \subseteq X^\omega$.
In the present pager we give a relationship between both kind of topologies. It turns out that every automaton-definable topology is a U-δ-topology for a suitable $U \subseteq X^*$.
An algorithm constructing $U \subseteq X^*$ from a given automaton is described. Conversely, we derive properties of automaton-definable topologies which prove that not every U-δ-topology can be defined by a weighted automata.

References

[CD93] K. Čulik II and S. Dube, Rational and affine expressions for image description, *Discrete Appl. Math.* **41** (1993), pp. 85 – 120.

[CK93] K. Čulik II and J. Kari, Compressiing images using weighted finite automata, *Computer and Graphics* **17** (1993), pp. 305 – 313.

[DN92] Ph. Darondeau, D. Nolte, L. Priese and S. Yoccoz, Fairness, Distances and Degrees. *Theoret. Comput. Sci.* **97** (1992), 131–142.

[FS01] Fernau, H. and Staiger, L., Iterated Function Systems and Control Languages, *Inform. and Comput.*, Vol. **168** (2001), pp. 125 – 143.

[St87] Staiger, L., Sequential mappings of ω-languages. RAIRO–*Inform. Théor.*, Vol. **21** (1987), pp. 147 – 173.

[*]Electronic mail: `staiger@informatik.uni-halle.de`

Performance Analysis for Probabilistic Input/Output Automata via Linear Representations

Eugene W. Stark
Department of Computer Science,
State University of New York at Stony Brook,
Stony Brook, NY 11794 USA

A *probabilistic input/output automaton* (PIOA) is a finite automaton whose alphabet Σ is partitioned into *input actions* and *output actions*, which has a rational *probability* assigned to each input transition and a nonnegative rational *rate* assigned to each output action, and which satisfies certain *stochastic conditions.* Input transitions correspond to stimuli applied to a PIOA by its environment and output transitions correspond to responses made by the automaton. The rates assigned to the output transitions from a state are regarded as the parameters of a set of independent, exponentially distributed random variables whose values give the times remaining until the corresponding transitions will be taken. The sum of the rates of all the output transitions from a state is called the *total rate* for that state. PIOAs admit an operation of *composition*, which takes "compatible" PIOAs A_1 and A_2 and produces a kind of product automaton $A_1|A_2$, in which outputs of one component may synchronize with inputs of the other. *Closed* PIOAs, which are those with an empty set of input actions, describe continuous-time Markov chains. The composition operation thus permits one to give economical descriptions of CTMCs with very large state spaces.

Certain sets of trajectories for a PIOA may be described by specifying a finite sequence of actions that must occur, along with the total rate associated with each of states through which the trajectory must pass. Such a specification, which we call a *rated trace*, may be viewed as a finite word over the infinite alphabet $\Sigma \times \mathcal{Q}$, where $\mathcal{Q}$ denotes the set of rationals. A formal power series with rational coefficients over this alphabet amounts to a function from $(\Sigma \times \mathcal{Q})^*$ to $\mathcal{Q}$. We call such functions *observables.* Each PIOA A determines in a certain way a transformation of observables, which we call the *behavior* of A. It turns out that the mapping from PIOAs to their behaviors respects composition, and that interesting performance measures for a closed PIOA can be expressed in terms of the application of its behavior to particular observables. Moreover, it is sufficient to consider those observables that are *representable* in the classical sense for power series.. These considerations lead to a *compositional* algorithm for computing performance measures for PIOAs expressed as compositions of simpler components. The algorithm proceeds by starting with a representation of an initial observable, successively calculating the result of applying the behavior of each component to the representation obtained in the preceding step, and then reducing the new representation to one that is of minimal dimension. When all components have been applied, the final representation is evaluated on the empty word to obtain the result.

This talk will present a survey of the ideas outlined above and issues associated with the implementation of the indicated compositional analysis algorithm. Emphasis will be on the relationship with formal power series and rational functions.

Solution of (max-+)-linear inequality systems and some related topics

Karel Zimmermann
Faculty of Mathematics and Physics, Charles University Prague
Department of Applied Mathematics,
Malostranske namesti 25, CZ-118 00 Praha 1
e-mail: zimm@kam.ms.mff.cuni.cz

A survey of properties of solution sets of the following systems of inequalities will be given:

(1) $\quad max_{j \in N}(a_{ij} + x_j) \leq max_{k \in K}(b_{ik} + y_k), i \in S_1,$

(2) $\quad max_{j \in N}(a_{ij} + x_j) \geq max_{k \in K}(b_{ik} + y_k), i \in S_2,$

(3) $\quad \underline{x_i} \leq x_j \leq \overline{x_j}$ for $j \in N$,

(4) $\quad \underline{y_k} \leq y_k \leq \overline{y_k}$ for $k \in K$,

where S_1, S_2, N, K are given finite index sets and a_{ij}, b_{ik} and lower and upper bounds posed on the variables are given real numbers. Such systems are systems of inequalities with $(max, +)$-linear functions on both sides of the inequalities. The connection with systems of inequalities with residuated functions will be investigated. Algorithms which make it possible to find the maximum solution of systems (1)-(4) will be described. Solution of some optimization problems with the set of feasible solutions given by (1)-(4) will be studied. Motivating examples from the machine-time scheduling will be given.

Participants

List of Participants

Cyril Allauzen AT&T, New Jersey, USA

Peter R. J. Asveld Twente University of Technology, The Netherlands

Stephen Bloom Stevens Institute of Technology, Hoboken, USA

Björn Borchardt Technische Universität Dresden, Germany

Ralf Borgschulze Technische Universität Dresden, Germany

Kai Brünnler Technische Universität Dresden, Germany

Peter Buchholz Technische Universität Dresden, Germany

Manfred Droste Technische Universität Dresden, Germany

Zoltan Ésik University of Szeged, Hungary

Frank Felfe Technische Universität Dresden, Germany

Zoltan Fülöp University of Szeged, Hungary

Bernhard Ganter Technische Universität Dresden, Germany

Stephane Gaubert INRIA, Paris, France

Martin Gavalec University Hradec Králové, Czech Republic

Zsolt Gazdag University of Szeged, Hungary

Stefan Haar INRIA, Rennes, France

Bernd F. Heidergott EURANDOM, Eindhoven, The Netherlands

Benjamin Hiller Technische Universität Ilmenau, Germany

Steffen Hölldobler Technische Universität Dresden, Germany

Anna Ingólfsdóttir Aalborg University, Denmark

Claus Jürgensen Technische Universität Dresden, Germany

Juhani Karhumäki University of Turku, Finland

Peter Kemper Universität Dortmund, Germany

Bakhadyr Khoussainov University of Auckland, New Zealand

Daniel Kirsten Technische Universität Dresden, Germany

Sergey Kitaev Chalmers University of Technology, Sweden

Ines Klimann Université Paris, France

Jan Komenda CWI Amsterdam, The Netherlands

Dexter Kozen Cornell University, Ithaca, USA

Katsiaryna Krupchyk Belarus State University, Minsk, White Russia

Werner Kuich Technische Universität Wien, Austria

Dietrich Kuske University of Leicester, Great Britain

Leonard Kwuida Technische Universität Dresden, Germany

Eric Laugerotte Université Rouen, France

Sylvain Lombardy ENST, Paris, France

Grit Malik Technische Universität Dresden, Germany

Dragan Mašulović Technische Universität Dresden, Germany

Ingmar Meinecke Technische Universität Dresden, Germany

Victor Mitrana University of Bucharest, Romania

Mehryar Mohri AT&T, New Jersey, USA

G. J. Olsder Delft University of Technology, The Netherlands

Wilson Otasowie Kharkov State Academy of Food Technology and Management, Ukraine

Christian Pech Technische Universität Dresden, Germany

Ion Petre Turku Centre for Computer Science, Finland

Petr Podvesky Center for Computational Linguistics, Prague, Czech Republic

Reinhard Pöschel Technische Universität Dresden, Germany

Aliaksandr Puptsau Belarus State University, Minsk, White Russia

Bernd Reichel Otto-von-Guericke-Universität Magdeburg, Germany

Horst Reichel Technische Universität Dresden, Germany

Jan Rothe Technische Universität Dresden, Germany

Sebastian Rudolph Technische Universität Dresden, Germany

Jan Rutten CWI & Vrije Universiteit Amsterdam, The Netherlands

Jacques Sakarovitch ENST, Paris, France

Werner Schmidt Universität Greifswald, Germany

Holger Schulz Universität Gesamthochschule Siegen, Germany

Ludwig Staiger Martin-Luther-Universität Halle-Wittenberg, Germany

Eugene W. Stark State University of New York, USA

Ralf Stiebe Otto-von-Guericke-Universität Magdeburg, Germany

Lutz Straßburger Technische Universität Dresden, Germany

Igor Tarasyuk Technische Universität Dresden, Germany

Michael Thielscher Technische Universität Dresden, Germany

Heiko Vogler Technische Universität Dresden, Germany

Margit Voigt Technische Universität Ilmenau, Germany

Karel Zimmermann Charles University Prague, Czech Republic