5th International PhD School in Formal Languages and Applications

2006

2nd TERM

PROGRAMMES

FUZZY FORMAL LANGUAGES

Claudio Moraga, *University of Dortmund*
claudio.moraga@udo.edu

1. Introduction. Crisp formal languages, Chomsky hierarchy, Petri languages, Lindenmayer languages
2. Finite fuzzy automata, fuzzy grammars, fuzzy languages. Relationship between fuzzy automata and fuzzy grammars. Fuzzy languages as families of crisp languages
3. Crisp and fuzzy Petri nets. Crisp and fuzzy Petri languages. Peterson hierarchy of Petri languages. Evolution of fuzzy Petri languages
4. Crisp and fuzzy L-systems. The Lindenmayer-Chomsky hierarchy. Fuzzy L-systems. Calculating the membership degree of words

References

Hack, M. (1975), Petri net languages, Computation Structures Group Memo 124, Project MAC, MIT.
Lindenmayer, A. (1971), Developmental systems without cellular interactions, their languages and grammars, *Journal of Theoretical Biology* 30: 455-484.
Meyer zu Bexten, E., F. Sajadi & C. Moraga (1997), Properties of Lindenmayer fuzzy languages and α-driven Lindenmayer languages, in *Proceedings of the 27th International Symposium on Multiple–Valued Logic*: 195–200. IEEE/Computer Science Press.
Mitzumoto, M, J. Toyoda & K. Tanaja (1973), Examples of formal grammars with weights, *Information Processing Letters* 2(4): 311-336.
Moraga, C. (2000), Towards a fuzzy computability? *Mathware and Softcomputing* VI(2-3): 163-172.
Peterson, J.L. (1976), Petri net languages, *Journal of Computer and System Sciences* 13(1): 1-24.

ALGEBRAIC TECHNIQUES IN LANGUAGE THEORY

Zoltán Ésik, *Rovira i Virgili University*
zoltan.esik@urv.cat

We give a short introduction to fixed point theorems and the equational theory of fixed points. We define equational and rational subsets of an algebra and prove their equivalence. We apply this general result to word and tree languages. Then we show that, over linear functions on a semiring, the fixed point operation can be characterized by a star operation defining the semiring. We derive several identities of the star operation including the

Conway identities. We establish a general Kleene theorem for Conway semirings and show how this implies the classical theorems of Kleene and Schützenberger.

References

Bloom, S.L. & Z. Ésik (1993), *Iteration Theories*. Springer, Berlin.
Davey, B.A. & H.A. Priestley (1994), *Introduction to Lattices and Order*. Cambridge University Press, Cambridge.

STRING COMPLEXITY

Lucian Ilie, *University of Western Ontario*
ilie@csd.uwo.ca

The course will present a survey of some of the most important combinatorial complexity measures for strings, both finite and infinite. We focus on recent results and open problems.

We start with the most investigated complexity measure, namely, factor complexity. Lower and upper bounds are given and de Buijn strings are introduced, as the best known example for reaching the upper bounds. In the case of infinite strings, we present equivalent characterizations for ultimate periodicity. One of the most important classes of infinite strings, that of fixed points of morphisms (infinite strings obtained by iterating a morphism), is introduced and we present the factor complexity classes for these strings.

The second measure is Lempel-Ziv, which is at the core of the well known compression algorithms. We follow the same pattern as for factors. Lower and upper bounds are shown; de Bruijn strings are again an important example. An equivalent characterization for ultimate periodicity for infinite strings is given and the Lempel-Ziv complexity classes for fixed points of morphisms are presented.

Next main topic is repetitions. There are many types of repetitions which are thoroughly investigated, producing various complexity measures, such as the number of squares, the total number of repetitions, the number of runs (that is, maximal repetitions), etc. We shall present some of the latest results and open problems.

Several other complexity measures, less investigated but potentially interesting, are briefly surveyed.

Below are given the most important references. Quite a bit of the new material is not contained in any book; precise references will be given in class.

References

Allouche, J.-P. & J. Shallit (2003), *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, Cambridge.
Choffrut, C. & J. Karhumäki (1997), Combinatorics on words, in G.

Rozenberg & A. Salomaa, eds., *Handbook of Formal Languages*, vol. I: 329-438. Springer, Berlin.
Crochemore, M. & W. Rytter (1994), *Text Algorithms*. Oxford University Press, Oxford.
Lothaire, M. (1983), *Combinatorics on Words*. Addison-Wesley, Reading, MA. (2nd ed.: Cambridge University Press, Cambridge, 1997)
Lothaire, M. (2002), *Algebraic Combinatorics on Words*. Cambridge University Press, Cambridge.
Lothaire, M. (2005), *Applied Combinatorics on Words*. Cambridge University Press, Cambridge.
Smyth, W.F. (2003), *Computing Patterns in Strings*. Pearson Addison-Wesley, Reading, MA.

CONTEXT-FREE GRAMMAR PARSING

Giorgio Satta, *University of Padua*
satta@dei.unipd.it

1. Introduction to context-free grammar parsing
2. Tabular methods for context-free grammar parsing
3. Parsing through grammar transformations (covering grammars)
4. Parsing algorithms for mildly context-sensitive formalisms

References

Leermakers, R. (1989), How to cover a grammar, in *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*: 135-142, Vancouver.
Nederhof, M.-J. (1994), An optimal tabular parsing algorithm, in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*: 117-124, Las Cruces.
Nijholt, A. (1980), *Context-Free Grammars: Covers, Normal Forms, and Parsing*. Springer, Berlin.
Sippu, S. & E. Soisalon-Soininen (1988), *Parsing Theory: Languages and Parsing*. Springer, Berlin.

PARAMETERIZED COMPLEXITY

Jörg Flum, *University of Freiburg*
joerg.flum@math.uni-freiburg.de

1. Fixed-parameter tractability
2. Reductions
3. Parameterized intractability

References

Downey, R.G. & M.R. Fellows (1999), *Parameterized Complexity*. Springer, Berlin.
Papadimitriou, C.H. (1994), *Computational Complexity*. Addison-Wesley, Reading, MA.

PROBABILISTIC PARSING

Mark-Jan Nederhof, *Rovira i Virgili University*
markjan@let.rug.nl

1. Probabilistic context-free grammars (PCFGs) and their formal properties
2. Renormalization of PCFGs
3. Training PCFGs on the basis of corpora and simple smoothing methods
4. Probabilistic variants of context-free parsing algorithms
5. Agendas and Dijkstra's shortest-path algorithm extended by Knuth to PCFGs
6. Prefix probabilities
7. Probabilistic pushdown automata

References

Booth, T.L. & R.A. Thompson (1973), Applying probabilistic measures to abstract languages, *IEEE Transactions on Computers* C-22(5): 442-450.
Chi, Z. (1999), Statistical properties of probabilistic context-free grammars, *Computational Linguistics* 25(1): 131-160.
Goodman, J. (1999), Semiring parsing, *Computational Linguistics* 25(4): 573-605.
Jelinek, F. & J.D. Lafferty (1991), Computation of the probability of initial substring generation by stochastic context-free grammars, *Computational Linguistics* 17(3): 315-323.
Jelinek, F., J.D. Lafferty & R.L. Mercer (1992), Basic methods of probabilistic context-free grammars, in P. Laface & R. De Mori, eds., *Speech Recognition and Understanding: Recent Advances, Trends and Applications*: 345-360. Springer, Berlin.
Knuth, D.E. (1977), A generalization of Dijkstra's algorithm, *Information Processing Letters* 6(1): 1-5.
Nederhof, M.-J. (2003), Weighted deductive parsing and Knuth's algorithm, *Computational Linguistics* 29(1): 135-143.
Nederhof, M.-J. & G. Satta (2003), Probabilistic parsing as intersection, in *Proceedings of the 8th International Workshop on Parsing Technologies*: 137-148, LORIA, Nancy.
Ney, H. (1992), Stochastic grammars and pattern recognition, in P. Laface & R. De Mori, eds., *Speech Recognition and Understanding: Recent Advances, Trends and Applications*: 319-344. Springer, Berlin.
Santos, E.S. (1972), Probabilistic grammars and automata, *Information and Control* 21: 27-47.
Stolcke, A. (1995), An efficient probabilistic context-free parsing algorithm that computes prefix probabilities, *Computational Linguistics* 21(2): 167-201.

NATURAL LANGUAGE PROCESSING WITH SYMBOLIC NEURAL NETWORKS

Risto Miikkulainen, *University of Texas, Austin*
risto@cs.utexas.edu

1. Introduction to neural networks. Motivation and history. Neurons and networks. XOR example. NetTalk demo
2. Backpropagation method. Perceptrons, generalized perceptrons. Intuitive and mathematical derivation. Extensions. Character recognition demo
3. Learning sequences. Recurrent networks. Finding structure in time. Word sense disambiguation. Disambiguation demo
4. Cognitive language processing. Goals and method of cognitive science. Distributed representations. Subsymbolic parsing. SPEC model
5. Scaling up to real-world language. Grammars for natural language. Syntactic and semantic corpora. INSOMNET demo
6. Scaling up to discourse level. Parsing, generation, lexicon, memory, q/a models. Building large modular systems. DISCERN demo

Exercise: Finding structure in sentences

References

Elman, J.L. (1991), Distributed representations, simple recurrent networks, and grammatical structure, *Machine Learning* 7: 195-225. (optional)
Lawrence, S., C.L. Giles & S. Fong (2000), Can recurrent neural networks learn natural grammars?, in International Conference on Neural Networks (ICNN9): 1853-1858.
Mayberry, M.R. III & R. Miikkulainen (1994), Lexical disambiguation based on distributed representations of context frequency, in *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society*: 601-606. Lawrence Erlbaum, Hillsdale, NJ. (optional)
Mayberry, M.R. III & R. Miikkulainen (2003), Incremental nonmonotonic parsing through semantic self-organization, in *Proceedings of the 25th Annual Conference of the Cognitive Science Society*: 798-803. Lawrence Erlbaum, Hillsdale, NJ.
Miikkulainen, R. (1997), Natural language processing with subsymbolic neural networks, in A. Browne, ed., *Neural Network Perspectives on Cognition and Adaptive Robotics*: 120-139. Institute of Physics Press, Bristol.
Miikkulainen, R. (2000), Text and discourse understanding: the ISCERN system, in R. Dale, H. Moisl & H. Somers, eds., *Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*: 905-919. Marcel Dekker, New York.
Russell, S. & P. Norvig (2002), Neural networks, section 20.5 in *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ.

MATHEMATICAL EVOLUTIONARY GENOMICS

David Sankoff, *University of Ottawa*
sankoff@uottawa.ca

1. Combinatorial and algorithmic approaches: the Hannenhalli-Pevzner algorithms for genome rearrangement reconstruction and generalizations; genome doubling inference; rearrangement phylogeny.
2. Probabilistic models and statistical analyses: the Nadeau-Taylor theory of breakpoint distribution and reformulations; estimating translocations and inversions; tests for gene clustering.

References

Sankoff, D. (2005), Conserved segment statistics and rearrangement inferences in comparative genomics, in O. Gascuel, ed., *Mathematics of Evolution and Phylogeny*. Oxford University Press, Oxford, in press.
Sankoff, D. & N. El-Mabrouk (2002), Genome rearrangement, in T. Jiang, T. Smith, Y. Xu & M. Zhang, eds., *Current Topics in Computational Biology*: 135-155. MIT Press, Cambridge, MA.

PARALLEL GRAMMARS

Henning Fernau, *University of Trier*
henningfernau@yahoo.de

1. Lindenmayer systems
2. Indian parallel grammars
3. (Uniformly) limited Lindenmayer systems
4. Interfaces with regulated rewriting and grammar systems: scattered context grammars and parallel communicating grammar systems

References

Herman, G.T. & G. Rozenberg (1975), *Developmental Systems and Languages*. North-Holland, Amsterdam.
Prusinkiewicz, P. & A. Lindenmayer (1990), *The Algorithmic Beauty of Plants*. Springer, New York.
Rozenberg, G. & A. Salomaa (1980), *The Mathematical Theory of L Systems*. Academic Press, New York.

TOPICS IN ASYNCHRONOUS CIRCUIT THEORY

Janusz Brzozowski, *University of Waterloo*
brzozo@uwaterloo.ca

1. Motivation for studying asynchronous circuits
2. Mathematical models of gates and gate circuits
3. Binary analysis of gate circuits
4. Hazards, hazard algebras, and simulation of circuits
5. Ternary algebra and ternary simulation
6. Relation between simulation in hazard algebras and binary analysis
7. Delay-insensitivity in asynchronous circuits

References

Brzozowski, J.A. (2005), Topics in asynchronous circuits, survey article written for this course; contains suggestions for further reading.
Brzozowski, J.A. & Z. Ésik (2003), Hazard algebras, *Formal Methods in System Design* 23(3): 223-256.
Brzozowski, J.A. & M. Gheorghiu (2005), Gate circuits in the algebra of transients, *Theoretical Informatics and Applications* 39: 67-91.
Brzozowski, J.A. & C.-J.H. Seger (1995), *Asynchronous Circuits*. Springer, Berlin.
Brzozowski, J.A. & H. Zhang (2000), Delay-insensitivity and semi-modularity, *Formal Methods in System Design* 16(2): 187-214.
Sutherland, I.E. & J. Ebergen (2002), Computers without clocks, *Scientific American*: 62-69.
Ye, Y. & J.A. Brzozowski (2005), Covering of transient simulation of feedback-free circuits by binary analysis, Maveric Res. Rept. 2005-01, University of Waterloo, ON, Canada, June.
http://maveric.uwaterloo.ca/publication.html

DNA COMPUTING: THEORY AND EXPERIMENTS

Natasha Jonoska, *University of South Florida*
jonoska@math.usf.edu

To be determined.

IMAGE COMPRESSION

Jarkko Kari, *University of Turku*
jkari@utu.fi

1. Bilevel images, regular languages and fractals
2. Gray-scale images and weighted finite automata (WFA)
3. Decoding WFA images
4. Encoding images in WFA, minimum state WFA
5. Image compression using weighted finite automata
6. Weighted finite transducers (WFT)

References

Culik, K. & J. Kari (1993), Image compression using weighted finite automata, *Computers & Graphics*, 17: 305-313.
Culik, K. & J. Kari (1996), Finite state transformations of images, *Computers & Graphics*, 20: 125-135.
Culik, K. & J. Kari (1997), Digital images and formal languages, in G. Rozenberg & A. Salomaa, eds., *Handbook of Formal Languages*, vol. 3: 599-616. Springer, Berlin.
Kari, J. & P. Fränti (1994), Arithmetic coding of weighted finite automata, *RAIRO-Informatique Théorique et Applications*, 28: 343-360.

BIOMOLECULAR NANOTECHNOLOGY

Max Garzon, *University of Memphis*
mgarzon@memphis.edu

1. Basics of DNA and bionanotechnology
2. DNA-based memories and computers
3. Self-assembly
4. Microfluidics
5. Selected applications (semantic retrieval and bio-complexity)

References

To be determined.

WEIGHTED AUTOMATA

Manfred Droste, *University of Leipzig*
droste@informatik.uni-leipzig.de

Weighted automata form an exciting field using methods from theoretical computer science, algebra, and combinatorics. They have recently received much interest due to their applications in digital image compression and in natural language processing. A weighted automaton consists of a finite number of states. Actions from an alphabet can induce a change of the current state into another one (a 'transition'), and each transition carries a weight describing e.g. the resources used for its execution, the length of time needed, its reliability, or an award associated with it. Thus a weighted automaton is simply a classical non-deterministic automaton with weights associated to the transitions.

In our lectures, we will first describe how to define the behaviour of a weighted automaton. The weights are typically taken from a semiring, and the behaviour can often be described by suitable homomorphisms from the free monoid of all words over the alphabet of actions into a monoid of matrices over the given semiring –this is how algebra and combinatorics enter the scene. Then we will describe a fundamental characterization of the possible behaviours of weighted automata by rational formal power series: the Kleene-Schützenberger theorem. Afterwards, we will come to more recent research results on transformations of such behaviours, to descriptions of infinite behaviours and/or on aperiodic and star-free behaviours of weighted automata.

(Standard) References

Berstel, J. & C. Reutenauer (1988), *Rational Series and Their Languages*. Springer, Berlin.
Kuich, W. & A. Salomaa (1986), *Semirings, Automata, Languages*. Springer, Berlin.
Salomaa, A. & M. Soittola (1978), *Automata-Theoretic Aspects of Formal Power Series*. Springer, Berlin.

(More particular) References

Droste, M. & P. Gastin (2000), Aperiodic and star-free formal power series in partially commuting variables, in D. Krob, A.A. Mikhalev & A.V. Mikhalev, eds., *Formal Power Series and Algebraic Combinatorics*: 158-169. Springer, Berlin.
Droste, M. & D. Kuske (2003), Skew and infinitary formal power series, in J.C.M. Baeten, J.K. Lenstra, J. Parrow & G.J. Woeginger, eds., *Automata, Languages and Programming*, Lecture Notes in Computer Science 2719: 426-438. Springer, Berlin.
Droste, M. & G.-Q. Zhang (2003), On transformations of formal power series, *Information and Computation* 184: 369-383.

CATEGORIAL GRAMMARS

Michael Moortgat, *Utrecht University*
michael.moortgat@let.uu.nl

The framework. Type-logical categorial grammar goes back historically to the logical perspective on linguistic structure developed by Jim Lambek in the late 1950ies. Current research distinguishes itself from this early work by considering a richer vocabulary of logic constants (type-forming operations) and by the use of more refined notions of structural reasoning, inspired by the modalities of linear logic. In this course, we study categorial grammar from an interdisciplinary perspective, combining themes from Logic (prooftheory and modeltheory of grammatical derivability), Computer Science (parsing-as-deduction, grammar induction from corpora), and Linguistics (the logic of language variation).

Course outline. The central aim of the categorial approach is to identify the invariants of grammatical composition, and to develop a logical perspective on the structural rules which, in combination with these invariants, give rise to the actual diversity of language.

PART ONE. In categorial grammar, the assembly of form and meaning is analysed in terms of n-ary type-forming operations, with a Kripke-style interpretation as n+1-ary composition relations. The grammatical invariants are the laws that do not put any restriction on the interpreting composition relations. Languages differ in allowing certain structural deformations under which the basic form-meaning correspondences are preserved. These deformations are captured in terms of structural postulates, i.e. non-logical axioms that constrain the interpreting composition relations to have particular properties. The general theory of structural control has been developed by Kurtonina and Moortgat, and is now applied to the actual study of cross-linguistic variation.

PART TWO. For an optimal division of labour between the invariant base logic and the structural module, it is important to squeeze as much explanatory power as possible out of the structurally-free core system. Current research efforts are concentrated on enriching the logical vocabulary of the grammatical base logic, thus making it possible to recognize as grammatical invariants certain principles of linguistic organization that before were thought to require the introduction of non-logical axioms. A case in point is the analysis of scope construal of generalized quantifier expressions. Scope construal can be accounted for in terms of dual residuated pairs (fission and its co-implications) next to the familiar fusion/composition operation and its residuals; the interaction between these dual perspectives on linguistic structure is regulated in terms of weak distributivity principles originally studied by V.N. Grishin. Reducing the role of the structural component opens perspectives on lowering the computational complexity of the grammar logic.

Background material. To prepare for the course, participants can read Moortgat (2002). To prepare for the exam, Moortgat (1997). For the lab session, we will use the on-line version of Richard Moot's (2002) categorial theorem prover (available at http://grail.let.uu.nl/).

References

Moortgat, M. (1997), Categorial type logics, in J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, ch. 2, pp. 93-177. Elsevier/MIT Press, Amsterdam/Cambridge, MA.
Moortgat, M. (2002), Categorial grammar and formal semantics, in L. Nagel, ed., *Encyclopedia of Cognitive Science*, vol. 1, pp. 435-447. Nature, London.
Moot, R. (2002), Proof Nets for Linguistic Analysis, PhD dissertation, Utrecht Institute of Linguistics OTS, Utrecht University.

UNIFICATION GRAMMARS

Shuly Wintner, *University of Haifa*
shuly@cs.haifa.ac.il

The course introduces the foundations of some of the major formalisms used in computational linguistics nowadays, providing both the linguistic motivation and the necessary mathematical infrastructure.

1. Context-free grammars. Basics: strings, grammars, derivations, languages, trees. Properties of CFGs. The (in)adequacy of CFGs for describing natural languages
2. Extending CFGs: feature structures. Motivation. Properties: features, values, variables, paths, reentrancy. Subsumption and unification. Representing lists, trees and graphs
3. Unification grammars. Adding features to rules. Multi-AVMs, forms, derivations, languages, trees. Internalizing categories
4. Linguistic examples. Imposing subject-verb agreement. Case control. Subcategorization. Unbounded dependencies. Coordination
5. The expressiveness of unification grammars. Grammars for trans-context-free languages. Turing equivalence
6. The mathematics of feature structures

References

The course is based on a textbook that the instructor is coauthoring (with Nissim Francez). Relevant chapters of the book will serve as the course's reader.

GRAMMATICAL INFERENCE

Colin de la Higuera, *University of Saint-Étienne*
colin.delahiguera@univ-st-etienne.fr

1. Inductive inference, polynomial issues
2. Learning finite state automata
3. Learning context-free grammars
4. Learning from trees and structured data
5. Combinatorial techniques, artificial intelligence techniques
6. Stochastic finite automata and grammars
7. Applications

References

Angluin, D. (1992), Computational learning theory: survey and selected bibliography, in *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*: 351-369. ACM Press, New York.
Ferri, F.J., J.M. Iniesta, A. Amin & P. Pudil, eds. (2000), *Advances in Pattern Recognition*, Lecture Notes in Computer Science 1876. Springer, Berlin.
Kearns, M.J. & U.V. Vazirani (1994), *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA.

SEQUENTIAL PATTERN MATCHING

Thierry Lecroq, *University of Rouen*
thierry.lecroq@univ-rouen.fr

Aims. The course is devoted to algorithms processing strings and texts efficiently. These types of algorithms are used for software design in the domains of operating systems utilities, search engines on the Internet, data retrieval systems, analysis of genetic sequences, and natural language processing, for example. The underlying methodology borrows elements from automata theory and combinatorics on words.

1. Text searching
2. Structures for indexing texts

References

Apostolico, A. & Z. Galil, eds. (1997), *Pattern Matching Algorithms*. Oxford University Press, Oxford.
Charras, C. & T. Lecroq (2004), *The Handbook of Exact String Matching Algorithms*. King's College, London.
Crochemore, M., C. Hancart & T. Lecroq (2001), *Algorithmique du Texte*. Vuibert, Paris.
Crochemore, M. & T. Lecroq (2004), Pattern matching and text compression algorithms, in A.B. Tucker Jr., ed., *The Computer Science and Engineering Handbook*: 13.1-13.48. CRC Press, Boca Raton, FL.
Crochemore, M. & T. Lecroq (2006), Text searching and indexing, in Z. Ésik, C. Martín-Vide & V. Mitrana, eds., *Recent Advances in Formal Languages and Applications*: 43-80. Springer, Berlin.

Crochemore, M. & W. Rytter (2002), *Jewels of Stringology*. World Scientific, Singapore.
Gusfield, D. (1997), *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, Cambridge.
Navarro, G. & M. Raffinot (2002), *Flexible Pattern Matching in Strings*. Cambridge University Press, Cambridge.
Smyth, W.F. (2003), *Computing Patterns in Strings*. Pearson Addison-Wesley, Reading, MA.
Stephen, G.A. (1994), *String Searching Algorithms*. World Scientific, Singapore.

CRYPTOGRAPHY

Valtteri Niemi, *Nokia Research Center*
valtteri.niemi@nokia.com

1. Block ciphers, stream ciphers and public-key cryptography
2. Security protocols
3. Industrial applications of cryptography

References

Menezes, A., P.C. van Oorschot & S.A. Vanstone (1996), *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL.
Schneier, B. (1996), *Applied Cryptography*. John Wiley, New York.

SPLICING SYSTEMS

Paola Bonizzoni, *University of Milan Bicocca*
bonizzoni@disco.unimib.it

1. Recombinant DNA, gene splicing and formal language theory
2. Finite (linear, circular) splicing systems and regular languages
3. Algorithms in splicing systems theory

References

Bonizzoni, P., C. de Felice, G. Mauri & R. Zizza (2004), Circular splicing and regularity, *Theoretical Informatics and Applications* 38: 189-228.
Bonizzoni, P., C. de Felice & R. Zizza (2005), The structure of reflexive regular splicing languages via Schützenberger constants, *Theoretical Computer Science* 334(1-3): 71-98.
Bonizzoni, P. & G. Mauri (2005), Regular splicing languages and subclasses, *Theoretical Computer Science* 340: 349-363.
Head, T., G. Păun & D. Pixton (1997), Language theory and molecular genetics: generative mechanisms suggested by DNA recombination, in G. Rozenberg & A. Salomaa, eds., *Handbook of Formal Languages*, vol. 2: 295-360. Springer, Berlin.
Kim, S.M. (1997), Computational modeling for genetic splicing systems, *SIAM Journal of Computing* 26: 1284-1309.
Păun, G., G. Rozenberg & A. Salomaa (1998), *DNA Computing: New Computing Paradigms*. Springer, Berlin.

MATHEMATICAL FOUNDATIONS OF LEARNING THEORY

Satoshi Kobayashi, *University of Electro-Communications*
satoshi@cs.uec.ac.jp

1. Preliminaries
2. Identification in the limit
3. Identification from positive data
4. Various sufficient conditions for identification from positive data
5. Approximate identification
6. Conclusions

References

Angluin, D. (1980), Inductive inference of formal languages from positive data, *Information and Control* 45: 117-135.
Gold, E.M. (1967), Language identification in the limit, *Information and Control* 10: 447-474.
Kobayashi, S. & T. Yokomori (1995), On approximately identifying concept classes in the limit, in K.P. Jantke, T. Shinohara & T. Zeugmann, eds., *Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence 997: 298-312. Springer, Berlin.
Kobayashi, S. & T. Yokomori (1997), Learning approximately regular languages with reversible languages, *Theoretical Computer Science* 174: 251-257.
Shinohara, T. (1994), Rich classes inferable from positive data: length-bounded elementary formal systems, *Information and Control* 108: 175-186.
Wright, K. (1989), Identification of unions of languages drawn from an identifiable class, in *Proceedings of the 2nd Annual Workshop on Computational Learning Theory*: 328-333. Morgan Kaufmann, San Francisco, CA.

AN INTRODUCTION TO THE THEORY OF FINITE TRANSDUCERS

Jacques Sakarovitch, *École Nationale Supérieure des Télécommunications*
sakarovitch@enst.fr

1. Finite transducers and rational relations. Examples. The Rabin-Scott model. The image and composition theorems
2. Theoretical bases. Rational and recognisable sets. Morphisms and covering of automata. Equivalence theorems
3. Synchronised relations. Bounded length difference relations. Resynchronisation algorithm. The Boolean algebra of synchronised relations
4. Functional transducers. Deciding functionality. Uniformisation of rational relations. Structure theorem for rational functions
5. Sequential transducers. Deciding sequentiality. Algebraic characterisation of sequential functions. Minimisation of sequential transducers

References

The instructor will follow the corresponding sections in his book *Éléments de Théorie des Automates*. Vuibert, Paris, 2003, whose translation in English should appear soon at Cambridge University Press. It will not be available by the time of the lectures, but some excerpts will be distributed to the students.

Among other classical references for transducers:

Berstel, J. (1979), *Transductions and Context-Free Languages*. Teubner, Stuttgart.
Eilenberg, S. (1974), *Automata, Languages and Machines*, Volume A, Academic Press, New York.
Roche, E. & Y. Schabes (1997), *Finite-State Language Processing*. MIT Press, Cambridge, MA.

DATA COMPRESSION

Wojciech Rytter, *Warsaw University*
rytter@duch.mimuw.edu.pl

To be determined.

AUTOMATA FOR VERIFICATION

Moshe Vardi, *Rice University*
vardi@cs.rice.edu

To be determined.

CELLULAR AUTOMATA

Martin Kutrib, *University of Giessen*
kutrib@informatik.uni-giessen.de

1. Basics on cellular automata
2. Stacks, queues, and counters
3. Synchronization and constructibility
4. Cellular automata as language acceptors
5. Relations to context-free languages
6. Hierarchies

References

Buchholz, T., A. Klein & M. Kutrib (2000), Iterative arrays with small time bounds, in M. Nielsen & B. Rovan, eds., *Mathematical Foundations of Computer Science 2000*, Lecture Notes in Computer Science 1893: 243-252. Springer, Berlin.
Buchholz, T., A. Klein & M. Kutrib (2002), On interacting automata with limited nondeterminism, *Fundamenta Informaticae* 52: 15-38.
Buchholz, T. & M. Kutrib (1997), Some relations between massively parallel arrays, *Parallel Computing* 23: 1643-1662.
Buchholz, T. & M. Kutrib (1998), On time computability of functions in one-way cellular automata, *Acta Informatica* 35: 329-352.

Burks, A.W. (1970), *Essays on Cellular Automata*. University of Illinois Press, Urbana, IL.
Choffrut, C. & K. Culik II (1984), On real-time cellular automata and trellis automata, *Acta Informatica* 21: 393-407.
Cole, S.N. (1969), Real-time computation by n-dimensional iterative arrays of finite-state machines, *IEEE Transactions on Computers* C-18: 349-365.
Dyer, C.R. (1980), One-way bounded cellular automata, *Information and Control* 44: 261-281.
Ibarra, O.H., S.M. Kim & S. Moran (1985), Sequential machine characterizations of trellis and cellular automata and applications, *SIAM Journal on Computing* 14: 426-447.
Ibarra, O.H. & M.A. Palis (1985), Some results concerning linear iterative (systolic) arrays, *Journal of Parallel and Distributed Computing* 2: 182-218.
Iwamoto, C., T. Hatsuyama, K. Morita & K. Imai (2002), Constructible functions in cellular automata and their applications to hierarchy results, *Theoretical Computer Science* 270: 797-809.
Klein, A. & M. Kutrib (2003), Fast one-way cellular automata, *Theoretical Computer Science* 295: 233-250.
Kutrib, M. (1999), Array automata and context-free languages, Report 9907, Institute of Informatics, University of Giessen.
Smith III, A.R. (1972), Real-time language recognition by one-dimensional cellular automata, *Journal of Computer and System Sciences* 6: 233-253.
Terrier, V. (1995), On real time one-way cellular array, *Theoretical Computer Science* 141: 331-335.
Terrier, V. (1996), Language not recognizable in real time by one-way cellular automata, *Theoretical Computer Science* 156: 281-287.
Umeo, H. & N. Kamikawa (2002), A design of real-time non-regular sequence generation algorithms and their implementations on cellular automata with 1-bit inter-cell communications, *Fundamenta Informaticae* 52: 255-275.
Umeo, H., K. Morita & K. Sugata (1982), Deterministic one-way simulation of two-way real-time cellular automata and its related problems, *Information Processing Letters* 14: 158-161.
Waksman, A. (1966), An optimum solution to the firing squad synchronization problem, *Information and Control* 9: 66-78.

QUANTUM AUTOMATA

Jozef Gruska, *Masaryk University, Brno*
gruska@informatics.muni.cz

To be determined.

AQUEOUS COMPUTING

Tom Head, *Binghamton University*
tom@math.binghamton.edu

1. The origin of the concept of aqueous computing
2. Laboratory aqueous computing in Leiden
3. Laboratory aqueous computing in Binghamton
4. The future

References

1:
Head, T. (2000), Circular suggestions for DNA computing, in A. Carbone, M. Gromov & P. Prusinkiewicz, eds., *Pattern Formation in Biology, Vision and Dynamics*: 325-335. World Scientific, Singapore.
Head, T. (2001), Splicing systems, aqueous computing, and beyond, in I. Antoniou, C.S. Calude & M.J. Dinneen, eds., *Unconventional Models of Computation*, UMC'2K: 68-84. Springer, Berlin.

2:
Head, T., G. Rozenberg, R. Bladergroen, C.K.D. Breek, P.H.M. Lommerse & H. Spaink (2000), Computing with DNA by operating on plasmids, *BioSystems* 57: 87-93.

3:
Head, T., X. Chen, M.J. Nichols, M. Yamamura & S. Gal (2001), Aqueous solutions of algorithmic problems: emphasizing knights on a 3X3, ms.
Head, T., M. Yamamura & S. Gal (1999), Aqueous computing: writing on molecules, in *Proceedings of the Congress on Evolutionary Computing 1999*: 1006-1010. IEEE Service Center, Piscataway, NJ.
Yamamura, M., T. Head & S. Gal (2000), Aqueous computing: mathematical principles of molecular memory and its biomolecular implementation, in H. Kitano, ed., *Genetic Algorithms 4*: 49-73 (in Japanese).

4:
Head, T. (2000), Aqueous simulations of membrane computations, Research Group on Mathematical Linguistics, *Reports GRLMC 13/00*, Universitat Rovira i Virgili, Tarragona, 10 pp.
Head, T. (2001), An aqueous algorithm for finding the bijections contained in a binary relation, ms.
Head, T. (2001), Splicing systems, aqueous computing, and beyond, in I. Antoniou, C.S. Calude & M.J. Dinneen, eds., *Unconventional Models of Computation*, UMC'2K: 68-84. Springer, Berlin.
Head, T. (2001), Writing by methylation proposed for aqueous computing, in C. Martín-Vide & V.Mitrana, eds., *Where Mathematics, Computer Science, Linguistics and Biology Meet*: 353-360. Kluwer, Dordrecht.
Head, T., X. Chen, M.J. Nichols, M. Yamamura & S. Gal (2001), Aqueous solutions of algorithmic problems: emphasizing knights on a 3X3, ms.