Editorial

The papers in this volume are revised and extended versions of communications presented at the Third International AMAST Workshop on Algebraic Methods in Language Processing (AMiLP-3), held at the University of Verona, Verona, Italy, 25–27 August 2003.

Like the previous two editions of this workshop, that were, respectively, held in December 1995 at the University of Twente, Enschede, The Netherlands, and in May 2000 at the University of Iowa, Iowa City, Iowa, USA, the workshop was organized in the framework provided by the Algebraic Methodology and Software Technology (AMAST) movement. In this framework large international conferences and specialized workshops are held. The latest AMiLP workshop considered algebraic methods in formal languages, programming languages and natural languages. Anton Nijholt (Enschede, Netherlands) and Giuseppe Scollo (Verona, Italy) were responsible for the scientific program of the workshop, chairing a Program Committee which also included Domenico Cantone (Catania), Roberto Giacobazzi (Verona), Dirk Heylen (Enschede), Aravind Joshi (Philadelphia), Geert-Jan Kruijff (Saarbrücken), Vincenzo Manca (Verona), Uwe Mönnich (Tübingen), Till Mossakowski (Bremen), Mark-Jan Nederhof (Groningen), Maurice Nivat (Paris), Teodor Rus (Iowa City), Fausto Spoto (Verona), and Martin Wirsing (München).

The papers presented at the workshop were selected out of those submitted in response to a public call for papers; in addition, knowledgeable researchers were invited to present an overview of relevant research of theirs. These lectures were given by:

- Joseph Goguen: Semiotic Morphisms, Representations, and Blending for Interface Design
- Giancarlo Mauri: Word Design for Molecular Computing
- Christian Retoré: Semantic Aspects of Minimalist Grammars
- Antonino Salibra: Lambda Calculus: Models and Theories
- Mark-Jan Nederhof and Giorgio Satta: Probabilistic Parsing Strategies

The workshop provided the participants with a stimulating opportunity to exchange their ideas and compare their latest results. Uwe Mönnich accepted to join the AMiLP-3 PC Chair in the task of acting as Guest Editors of this special issue. A preliminary selection of papers for further review was carried out soon after the conclusion of the meeting, whereby authors were invited to submit revised, extended versions of their contributions. These underwent a careful refereeing and further revision, eventually resulting in the contents of the present issue, which testify to the continuing process of cross-fertilization between several disciplines, where the use of algebraic methods appears to be the main catalyst. We hope the results presented in this issue, the problems still left open and the new questions which arise here, will foster further research into what appears to be a mature, yet largely unexplored and exciting field of investigation.

Papers in this volume

Foundations of semantics of formal languages is the wide area hosting the first three contributions to this special issue.

Chantal Berline and Antonino Salibra generalize Baeten and Boerboom's method of forcing to obtain novel results relating to the lattice of lambda theories. Three applications of two corollaries of their main result respectively prove the existence of: (1) a finitely axiomatized λ-theory $\mathcal{L}$ such that the interval lattice constituted by the λ-theories extending $\mathcal{L}$ is distributive; (2) a continuum of pairwise inconsistent graph theories ($= \lambda$-theories that can be realized

as theories of graph models); (3) a congruence distributive variety of combinatory algebras (lambda abstraction algebras, respectively).

A novel approach to integrating a process algebra, viz. CSP, within the algebraic specification language CASL is proposed by Markus Roggenbach. Novel aspects include the combination of denotational semantics of the process part and, in particular, loose semantics for the data types, while covering both concepts of partiality and subsorting. This integration involves the development of a new so-called data-logic formulated as an institution. This data-logic serves as a link between the institution underlying CASL and the alphabet of communications necessary for the CSP semantics. Besides being generic in the various denotational CSP semantics, this construction leads also to an appropriate notion of refinement, with clear relations to both data refinement in CASL and process refinement in CSP.

In Dana Harrington's paper a uniqueness logic is introduced. In functional programming languages it is important to distinguish values of a variable which are referenced once from values that are referenced more than once in a program. Modifying a shared value can invalidate occurrences that yet have to be evaluated. Knowing about these properties makes it possible to introduce useful optimizations. Uniqueness type systems are used to describe these properties. In Harrington's contribution to this special issue a variant of Girard's intuitionistic linear logic is defined to express properties of uniqueness type systems using the Curry–Howard–Lambek isomorphism between propositions from logic and types from programming languages. A novel type system with a categorical semantics and extensions motivated by categorial considerations is proposed. The so-called uniqueness logic that is introduced here allows reasoning about uniqueness types.

The next six papers find their motivation in computational linguistics.

In his contribution, Gerald Penn describes several alternative data structures and algorithms for multiplying sparse upper-triangular matrices over closed semirings, and evaluates their efficiency in computing transitive closures over matrices over the Boolean semiring. Starting point for introducing the data structures and algorithms are applications and available resources in computational linguistics. For example, these applications require computations related to feature structure unification in different variants of feature grammars, computations related to deriving all-path parse forests for context-free grammars and computations related to path-accessibility in semantic nets. Rather than looking at artificially generated data-sets the author looks at data-sets found in linguistic applications. Taking this approach may be considered as an important contribution by this paper. Other contributions are a new representation for sparse matrices that allows efficient matrix multiplication, new views on the application of Dijkstra's shortest-path algorithm and the introduction of a variant of this algorithm, the so-called Topological Dijkstra algorithm.

In the contribution by Stephan Kepser and Uwe Mönnich on closure properties of context-free tree languages with an application to optimality theory, we see how results from formal language theory, in particular on tree languages and tree automata, can be used to model optimality theory. The closure properties are obtained on the class of linear (non-duplicating) context-free tree languages. They are closed under linear bottom–up tree transductions and under intersection with regular tree languages. These results are applied to optimality theory. This theory provides a framework for using rules and filters (constraints) to map one level of linguistic representation to another. The results on closure properties obtained by Kepser and Mönnich show how tree transducers, regular tree languages and other finite-state techniques can be used to generate linguistic representations and their constraints.

Modeling of metaphoric expressions is the topic of the paper by Helmar Gust, Kai-Uwe Kühnberger and Ute Schmidt. This work, as has been done by many others, builds on an algebraic framework provided by Indurkhya. Metaphoric expressions are strings made up of elements of a source and target concept network, and both these networks and a metaphorical relation between them that allows the wording of the metaphor, are represented as algebras. In their contribution, continuing this formal approach, the authors discuss the problems that emerge when straightforward algebraic descriptions of the source and target domains do not suffice. That is, when for example full first-order logic needs to be used, standard (anti-unification) approaches to get to a metaphor cannot be used. Instead the authors introduce their own and original framework, allowing both more general domain descriptions and analogical reasoning. Using this framework different types of metaphors are discussed.

In his paper, Peter R.J. Asveld examines the complexity, in terms of number of non-terminals, number of productions and, briefly, number of left-most derivations, of a sequence of families of context-free grammars in Chomsky normal form that generate the (finite) language of all permutations of a given alphabet. He develops and analyzes a number of recursive approaches to obtaining such grammar families and compares their resulting complexity measures with respect to the criteria just mentioned. In a sense, the paper can be regarded as a taxonomy of these particular approaches. There is no claim that the corresponding grammar families exhaust the class of context-free grammars generating

permutations nor that the grammars obtained by means of any of the approaches are optimal in terms of the specific complexity measures under investigation. Some of the sequences of cardinalities of productions and non-terminals that are established are novel and have been added to Sloane's Database of Integer Sequences.

Marcus Kracht explores in technical detail the relationship between partial algebras, many sorted algebras and generalized many sorted algebras that allow for polymorphism. Based on this comparison it is argued that an approach to compositional semantics which is informed by the ideas of partial algebra is superior to the more familiar Montagovian type-theoretical approach. The author thus provides a rigorous model for metatheoretical analysis that puts recent work in formal linguistics and algebraic semantics in perspective.

Discussions about proper interaction between dependency and constituency structures have a long tradition in formal linguistics. Radu Gramatovici and Carlos Martín-Vide offer a new framework in which word-order is imposed by a contextual mechanism and the dependency relation is constructed by means of tree rewriting methods. This framework is based on a structured version of a grammar formalism originally proposed by Marcus and inspired by the operation of inserting words in given phrases according to certain contextual dependencies. Since the order of daughter nodes in a dependency tree, in line with this analysis, may be specified by category symbols, it is possible to impose word-order by a contextually restricted insertion of another dependency tree. Depending on the type of restriction imposed on the operation of insertion the generative power of the resulting grammars can vary to a considerable degree. The authors show that grammars with regular contextual restrictions are able to generate the context-sensitive string language $\{a^n b^n c^n | n \geqslant 0\}$ and they conjecture that the product of the mirror language with itself lies outside the generative spectrum of this type of contextual dependency grammar. The last part of the paper is devoted to an equivalence proof of these grammars with a particular kind of restarting automata.

Finally, language processing and translation is the main area of the contribution by Rohit Kundaji and R.K. Shyamasundar, who propose a generalized proof rule for establishing refinement of source and target programs for which one need not know the underlying program transformations. Their method comes equipped with a semi-automatic tool that uses a theorem prover for validating the verification conditions. They further show that the translation validation infrastructure provides an effective basis for deriving semantic debuggers and illustrate the development of a simple debugger for optimized programs using this approach.

Acknowledgments

Guest Editors
A. Nijholt, G. Scollo, U. Mönnich
Department of Computer Science,
University of Verona, Verona, Italy
E-mail address: giuseppe.scollo@univr.it (G. Scollo).