Theoretical Computer Science 293 (2003) 237–241
Theoretical Computer Science
www.elsevier.com/locate/tcs

Editorial

The papers in this issue are revised and extended versions of communications presented at the Second International AMAST Workshop on Algebraic Methods in Language Processing (AMiLP '00), held at the University of Iowa, Iowa City, USA, 20–22 May 2000. Like the first workshop, that was held in 1995 at the University of Twente, Enschede, the Netherlands, the workshop was organized in the framework provided by the Algebraic Methodology and Software Technology (AMAST) movement. Papers from this first workshop were collected in the issue of TCS 1–3, 1998. In the AMAST framework large international conferences and specialized workshops related to algebraic issues are held. The recent AMiLP workshop considered algebraic methods in formal languages, programming languages and natural languages. Responsible for the scientific program of the workshop were Anton Nijholt (program chair, Enschede, Netherlands), Dirk Heylen (Enschede, Netherlands), Giuseppe Scollo (Catania, Italy) and Teodor Rus (Iowa City, USA).

The presentations in the workshop were selected from the response to a call for papers and in addition some well known researchers were invited to present an overview of their research area. These lectures were given by

- Theo Janssen: An Algebraic Approach to Grammatical Theories for Natural Languages.
- Teodor Rus: Algebraic Definition of Programming Languages.
- Gheorghe Paun: Molecular computing and Formal Languages.

The workshop provided the participants with a stimulating opportunity to exchange their ideas and compare their latest results. A selection made at the workshop, followed by a careful refereeing of the invited contributions, resulted in the contents of the present issue, which testifies 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 issue

One of the recurrent themes in several of the papers in this issue is that of formally characterizing context-sensitive constructions in natural language. Linguists are interested in representing structural aspects of natural language expressions rather than

PII: S0304-3975(01)00346-2

determining which strings belong to some language; i.e. in strong rather than weak generative capacity. In his paper to this collection, Aravind Joshi takes as a motivational starting point the observation made by McCawley and others that the use of context-sensitive rules by linguists is primarily used to check structural descriptions and not characterizing strings. In this respect one of the goals that could be set is to try and squeeze more strong power out of a formal system without increasing the weak generative power. The paper provides a comment on past results and observations in light of recent work from Tiede on the characterization of proof trees in Lambek grammar (LG). Tiede shows that there are Lambek grammars whose proof tree language is not regular. He provides an example that characterizes crossing dependencies. Joshi considers the question whether this property can be used to represent certain cross-serial dependencies in natural language.

The topic of characterizing non-context freeness is also addressed in the paper by Hans-Peter Kolb, Jens Michaelis, Uwe Mönnich and Frank Morawietz. Structural phenomena, like cross-serial dependencies in Swiss German or Dutch, that cannot be captured by context-free string grammars or regular tree grammars, are analyzed by providing a description in terms of regular tree languages combined with a special type of tree transformation effected by a non-deleting macro tree transducer. This result is complemented by a second one, providing a logical description of cross-serial dependencies in terms of Monadic Second-Order (MSO) logic. It is shown how the structures underlying the dependencies can be specified as MSO definable relations.

In the past language-theoretic complexity classes have been characterized by definability in the (weak) monadic second-order theories of strings and trees. This work dates to the late fifties (J.R. Buchi and C.C. Elgot) and the late sixties (J. Doner, J.W. Thatcher, J.B. Wright and M.O. Rabin). Given an MSO formula a finite-state automaton can be produced that accepts exactly the set of structures that satisfy it, yielding characterizations of regular and context-free languages. More recently, by viewing the step from strings to trees as a step from one- to two-dimensional structures and then generalizing this to tree-like structures of arbitrary dimension, James Rogers has employed similar characterizations in the framework of standard Government and Binding (GB) theory and Generalized Phrase Structure Grammar (GPSG).

In his contribution to this issue, Rogers looks at similar, weak MSO characterizations, of tree-adjoining languages using three-dimensional tree-like structures, again showing that the generalization to arbitrary dimensions is not only a theoretical result in formal language theory and logic, but has different applications in computational linguistics. Rogers emphasizes this point by showing how aspects of Tree-Adjoining Grammars (TAGs) for English can be expressed in weak MSO and discusses the application of his results in the construction and maintenance of wide coverage TAGs for natural languages.

A language of tree descriptions is the main subject of the paper by Denys Duchier on dominance constraints with Boolean connectives. Tree descriptions are a widely used tool in computational linguistics for talking and reasoning about trees. The traditionally well-known conjunctive fragment is extended to an account of all Boolean connectives. While satisfiability is NP-complete already in the conjunctive fragment, it can be addressed very effectively by constraint propagation; the main result in the present paper is a treatment of disjunction that proves suitable for constraint propagation.

Edward Stabler and Edward Keenan introduce their programme of research that explores how the algebraic descriptions of natural languages can be used to characterize notions of the structural similarity between a whole range of linguistic structures within a particular language or among different languages. A minimalist grammar formalism is introduced and applied to several grammar fragments, illustrating how claims about the structural similarity of expressions are reflected in relations between these algebraic grammars.

Most parsers for context-free phrase structure grammars are based on tabular parsing algorithms. These come in different representations but share a common so-called dynamic programming approach, in which results of partial computations are stored and re-used for computation of subsequent results. As mentioned by Karl–Michael Schneider in his contribution to this special issue, algebraic descriptions of tabular parsing algorithms are of twofold interest. They allow the determination of the ontological status of the concepts that play a role in parsing and the characterization in terms of deduction systems and they allow the comparison of algorithms for different classes of grammars by means of algebraic transformations. In addition, one might hope to find new construction methods for tabular algorithms. Schneider presents a general, algebraic foundation of tabular parsing. A characterization of parsing is obtained by introducing two algebras, one with partial trees and one with tuples of strings as their elements. The parsing problem, then, is defined as the computation of the inverse image of an input string with respect to a homomorphism between these algebras. A parsing strategy is given by the choice of operations. The comparison of parsing algorithms using this framework is illustrated by a construction technique that transforms a correct tabular algorithm for context-free grammars (bottom-up head-corner algorithm) into a correct algorithm for linear indexed grammars (LIGs). This is an interesting result, not only in the light of previous attempts to show the relationships between tabular parsing algorithms for context-free and linear indexed grammars, but also in the light of attempts to compile tree-adjoining grammars into linear indexed grammars for the purpose of parsing.

Investigations in range concatenation grammars (RCGs) have concerned both mathematical properties of the formalism as the application of describing constructions in natural language. For instance, it has been shown how different classes of TAGs can be translated into the RCG formalism. Although the class of RCG languages strictly contains the mildly context-sensitive languages it nevertheless has parsing algorithms that perform in polynomial time and RCGs can act as a syntactic backbone upon which feature structures or other decorations can be grafted. Pierre Boullier, in his contribution, surveys properties of RCGs and their languages. In particular he focuses on the behavior of RCGs in 'counting' by showing example languages that can be specified by RCGs while encountering problems using other formalisms. Here counting refers to the possibility of grammar elements to act as counters, remembering part of the history in the derivation of sentences. It is shown how in RCGs basic and complex arithmetic operations can be performed on counters. These operations include multiplication, testing that the length of a string is a prime and square root or logarithm.

As we see in several contributions to this special issue, results and yet not completely elaborated ideas from the early history of formal language theory still play an important role in new developments concerned with the construction of algebraic frameworks that allow comparisons between, e.g., grammar classes and parsing approaches.

In his contribution, Peter R.J. Asveld returns to a well-known milestone in the algebraic approach to the definition of families of formal languages: the introduction of the notion of (full) abstract family of languages (full AFL), introduced in the late sixties by S. Ginsburg, S.A. Greibach and J.E. Hopcroft. AFLs are concerned with closure properties of families of formal languages. However, rather than considering families of languages fitting in the traditional Chomsky hierarchy, Asveld discusses closure properties of so-called fuzzy languages, languages that allow for elements that are not completely in or out of the language, using a membership function that maps strings on a lattice-ordered structure rather than on a two-element set 0,1. Fuzzy languages can play a role in modeling grammatical errors and concepts in robust parsing approaches. A main part of Asveld's study is devoted to the set of regular fuzzy languages. Its role can be compared to that of the ordinary regular languages in the study of crisp formal languages. The fuzzy analogue of full AFL is introduced, i.e., a non-trivial family of fuzzy languages closed under appropriate fuzzy equivalents of the standard AFL closure operations. Weaker and more powerful variants can then be introduced analogous as in the study of standard AFL theory, and with an appropriate membership function, again analogous to standard AFL theory, infinite sequences of fuzzy AFLs can be obtained.

Finally, the paper on Compiling Dyadic First-Order Specifications into Map Algebra by Domenico Cantone, Andrea Formisano, Eugenio Omodeo and Calogero Zarba, contributes algorithms to translate a class of first-order formulae into the language of Tarski's relational calculus. These algorithms are equipped with correctness proofs, their computational complexity is assessed, their usage is illustrated by a wealth of examples where a convenient diagrammatic notation for relational expressions is employed, and on-going work on their implementation is reported.

Acknowledgements

Our words of thanks are addressed to the institutions and people who made AMiLP '00 and this special issue possible; in particular to the authors and the referees for their efforts towards meeting stringent quality requirements. The following referees, besides the editors and a majority of the authors, have helped us to review the papers in this special issue: Elvina Riccobene, Don Pigozzi, Klaas Sikkel, John Mordeson, Mark-Jan Nederhof, Gheorghe Paun, Theo Janssen, Michael Moortgat, Thom Fruehwirth, Roger

Maddux, Laura Kallmeyer, Antonio Salibra, Andrea Formisano, Eugenio Omodeo and Alain Lecomte.

We also wish to thank Maurice Nivat, Editor-in-Chief of Theoretical Computer Science, for his support to bringing this special issue into existence.

A. Nijholt[a]
G. Scollo
D. Heylen
Guest Editors

[a] *Department of Computer Science,*
University of Twente, Twente, Netherlands
E-mail: anijholt@cs.utwente.nl (A. Nijholt)