Preface

Across the boundary between the sixties and the seventies, facing the challenge of the emerging "software crisis", computer scientists suggested the use of abstraction for mastering complexity in software development. Indeed abstraction, appropriately associated with modularity, is first of all a powerful conceptual tool that can be exploited in any phase of the development process, in any application area and at a varying level of formalization.

Soon after the beginning of the seventies, thus more than 20 years ago, various attempts have been made at providing more rigorous foundations for the concept of abstraction and for related techniques. Almost inevitably, the foundations were based on mathematical techniques and, among these, mainly techniques based on logic and universal algebra emerged. At the beginning, the research efforts concentrated on exploring the formalization of the notion of abstract data type (shortly ADT) and the various techniques for their specifications. Later on, while the concept of abstract data type, formalized essentially as a many-sorted algebra, was beginning to be adopted systematically in books and courses, new and more ambitious research directions emerged, aimed at providing a conceptual basis and practical tools for the stepwise development of correct systems, appropriately structured into components. Moreover, the whole development process has been under investigation, from the early requirements phase to the last implementation step.

While the use of algebraic techniques started in the USA, soon European researchers got involved in the area, which became, at the end of the seventies, a major research topics in Europe. In 1982, as a natural consequence of the growing interest in the field, a series of workshops was initiated, of course named ADT Workshops. They have been regularly organized, two in every three years, after the first event in Langscheid, in Passau, Bremen, Warberg, Edinburgh, Berlin, Wusterhausen, Dourdan, Caldes de Malavella, Santa Margherita Ligure and Oslo. Since the 7th edition, to witness the maturity of the research carried out in the field, a subseries of the Lecture Notes in Computer Science appeared, under the name Recent Trends in Data Type Specification, which collects selected and carefully refereed papers, presenting written versions of some talks given in the workshops. In order to celebrate the 10th edition of the Workshop (S. Margherita, Italy, 30 May–3 June 1994), in the related volume, LNCS no. 906, a complete list of the talks of the first ten Workshops and of the related papers has been included. Moreover, following a very friendly suggestion and offer by Maurice Nivat, the current special issue of TCS has been planned, to provide an insight into the activity of the area. The papers in this issue, carefully refereed as usual,
have been written in response to an invitation by the guest editor, who has followed two main guidelines: to invite some of the main contributors to the previous ADT Workshops and to highlight a broad spectrum of research in the field. Though, unfortunately, not all the invitees have been able to accept, especially for contemporary other commitments, I believe that the volume is quite representative of the current research themes and techniques in the area and can easily motivate why the future Workshops, though still keeping the acronym ADT, will be associated with a different name, Algebraic Development Techniques, which is much more adequate to the topics of interest.

The first paper, by M. Cerioli and J. Meseguer, deals with a topic, relating and transporting logics, which has recently attracted a lot of interest; it addresses the issue of reusing parts of a logic in another and is characterized by the generality of the approach, appropriately supported by the language and techniques of categorical constructions. The paper by Hennicker, Wirsing and Bidoit, is illustrative of a modern way of supporting the development of structured specifications and their implementations, namely by endowing a specification language with proof systems; the notion of semantics here takes into account observability conditions expressed by observability operators. Jouannaud and Okada's paper is at the crossroad of different significant research streams, by introducing ADTS, the Abstract Data Type Systems, which on the one side generalize inductive types as in the calculus of constructions and on the other generalize the first-order framework of abstract data types by providing function types and higher-order equations. A classical subject, namely a friendly overview and definition of a specification language, is the subject of the work by Kahrs, Sannella and Tarlecki; but the novelty is that EML, an algebraic extension of SML, can be used as a support for a development methodology covering the phases from the abstract specification to the executable program (in SML). The other two papers are representative of the extension of algebraic techniques for addressing themes typically investigated in other areas. Orejas, Pino and Ehrig apply general algebraic techniques for investigating the compositionality of logic programs; their analysis relies on the use of the concept of institution and of institution-independent results. Finally, Costa and Reggio propose an extension of the classical algebraic framework to deal also with dynamic-data types, i.e. data representing dynamic/reactive systems; their approach provides a temporal many-sorted branching-time logic for the specification of dynamic systems, with various semantics and support for structuring and implementation.

Egidio Astesiano
Genova