Preface

CASL, the *Common Algebraic Specification Language*, has been designed by CoFI, the *Common Framework Initiative* for algebraic specification and development. CASL is an expressive language for specifying requirements and design for conventional software. It is algebraic in the sense that models of CASL specifications are algebras; the axioms can be arbitrary first-order formulas.

CASL is a major new algebraic specification language. It has been carefully designed by a large group of experts as a general-purpose language for practical use in software development – in particular, for specifying both requirements and design. CASL includes carefully selected features from many previous specification languages, as well as some novel features that allow algebraic specifications to be written much more concisely and perspicuously than hitherto. It may ultimately replace most of the previous languages, and provide a common basis for future research and development.

CASL has already attracted widespread interest within the algebraic specification community, and is generally regarded as a de facto standard. Various sublanguages of CASL are available – primarily for use in connection with existing tools that were developed in connection with previous languages. Extensions of CASL provide languages oriented toward development of particular kinds of software (reactive, concurrent, etc.).

Major libraries of validated CASL specifications are freely available on the Internet, and the specifications can be reused simply by referring to their names. Tools are provided to support practical use of CASL: checking the correctness of specifications, proving facts about them, and managing the formal software development process.

This reference manual gives a detailed presentation of the CASL specification formalism. It reviews the main underlying concepts, and carefully summarizes the intended meaning of each construct of CASL. It formally defines both the syntax and semantics of CASL, and presents a logic for reasoning about CASL specifications. It also provides extensive libraries of CASL specifications of basic datatypes, and an annotated bibliography of CoFI publications.

The companion *CASL User Manual* (LNCS 2900) illustrates and discusses how to write CASL specifications, introducing the potential user to the features of CASL mainly by means of illustrative examples. The User Manual also reviews the background of CoFI and CASL, and the underlying concepts of algebraic specification languages, as well as introducing the reader to some of the currently available CASL support tools, and to a couple of the CASL libraries of basic datatypes. Finally, the User Manual includes a substantial case study of the practical use of CASL in an industrially relevant context, and a Quick Reference overview of the CASL syntax.

Structure

Part I offers a definitive *summary* of the entire CASL language: all the language constructs are listed there systematically, together with the syntax used to write them down and a detailed explanation of their intended meaning. However, although it tries to be precise and complete, the CASL Summary still relies on natural language to present CASL. This inherently leaves some room for interpretation and ambiguity in various corners of the language, for example where details of different constructs interact. Such potential ambiguities are eliminated by the following formal definitions, which also establish sound mathematical foundations.

Part II gives a formal definition of the *syntax* of CASL. Both concrete and abstract syntax are defined by means of context-free grammars, using a variant of the BNF notation.

The ultimate definition of the meaning of CASL specifications is provided by the *semantics* of CASL in Part III. The semantics first defines mathematical entities that formally model the intended meaning of various concepts underlying CASL, which were introduced and discussed throughout the summary. The semantics is given in the form of so-called *natural semantics*, with formal deduction rules to derive judgments concerning the meaning of each CASL phrase from the meanings of its constituent parts.

The semantics is also a necessary prerequisite for the development of mechanisms for formal reasoning about CASL specifications. This is dealt with in Part IV, where *proof calculi* that support reasoning about the various layers of CASL are presented. Soundness is proved and completeness discussed by reference to the formal semantics of CASL.

All this work on the mathematical underpinnings of CASL, as documented in this Reference Manual, should make the language exceptionally trustworthy – at least in the sense that it provides a formal point of reference against which claims may (and should) be checked.

Finally, Part V presents extensive libraries of CASL specifications of basic datatypes. These include specifications of numbers (both bounded and unbounded), relations and orders, simple and structured datatypes, graphs, and various mathematical structures.

The Reference Manual is concluded by an annotated bibliography, a list of cited references, an index of specification and library names (referring to Part V), a symbol index, and an index of concepts.

An accompanying CD-ROM contains a copy of the libraries of specifications of basic datatypes and a collection of CASL tools.

Organization

CASL consists of several major *levels*, which are quite independent and may be understood (and used) separately:

Basic specifications denote classes of partial first-order structures: algebras where the functions are partial or total, and where also predicates are allowed. Subsorts are interpreted as embeddings. Axioms are first-order formulas built from definedness assertions and both strong and existential equations. Sort generation constraints can be stated. Datatype declarations are provided for concise specification of sorts equipped with constructors and (optional) selectors, including enumerations and products.

Structured specifications allow translation, reduction, union, and extension of specifications. Extensions may be required to be free; initiality constraints are a special case. A simple form of generic (parametrized) specifications is provided, together with instantiation involving parameter-fitting translations.

Architectural specifications define how the specified software is to be composed from a given set of separately developed, reusable units with clear interfaces.

Libraries allow the distributed storage and retrieval of (particular versions of) named specifications.

The CASL Summary in Part I is organized accordingly: after an introductory chapter, each level of CASL is considered in turn. The grammars for the abstract and concrete syntax of CASL in Part II are structured similarly. The chapters and sections of the CASL Semantics in Part III and of the CASL Logic in Part IV correspond directly to those of Part I. Thus readers interested in all aspects of one particular level of CASL should have no difficulty in locating the relevant chapters in each part, and similarly for all the sections dealing with a particular CASL construct.

References to chapters within the same part give just the chapter number, possibly following it by section and subsection numbers, e.g., Chap. 4, Sect. 4.2.3. References to chapters in other parts are always preceded by the Roman numeral indicating the part, e.g., Chap. III:4, Sect. III:4.2.3. Similarly for references to propositions, etc.

Acknowledgement. The design of CASL and the preparation of this book have involved a large group of persons, and a considerable amount of effort. Specific acknowledgements to contributors are given in the introductions to the individual parts. Much of the material on which this book is based was developed in connection with activities of CoFI-WG (ESPRIT Working Group 29432) and IFIP WG 1.3 (Working Group on Foundations of System Specification). The final design of CASL version 1.0.1 was reviewed and approved by WG 1.3 in April 2001. The current version (1.0.2) was adopted in October 2003; it incorporates adjustments to some minor details of the concrete syntax and semantics. No further revisions of the CASL design are anticipated.

Public drafts of this book were released in July and December 2003. The many insightful comments from CoFI participants were very helpful during the preparation of the final version. Detailed comments on all or part of the public drafts were received from Michel Bidoit, Christian Maeder, and Lutz Schröder, as well as from those responsible for the various parts of the book.

Special thanks are due to those responsible for editing Parts III–V: Don Sannella and Andrzej Tarlecki integrated several large, independently authored chapters into a coherent Part III, and Till Mossakowski took excellent care of the production of Parts IV–V.

Peter Mosses gratefully acknowledges support from BRICS[1] and the Department of Computer Science, University of Aarhus.

Finally, special thanks to Springer, and in particular to Alfred Hofmann as Executive Editor, for their willingness to publish this book, and for helpful advice concerning its preparation.

January 2004 Peter D. Mosses

News of the latest developments concerning CoFI and CASL is available on the Internet at `http://www.cofi.info`.

[1] Basic Research in Computer Science (`www.brics.dk`), funded by the Danish National Research Foundation.