

## *Book review*

*Essentials of Constraint Programming* by Thom Frühwirth and Slim Abdennadher, Springer, 2003, hard cover: ISBN 3-540-67623-6, ix + 145 pages.

This book is intended as a short and concise presentation of constraint programming and constraint reasoning that covers several aspects of the domain such as: formal and theoretical foundations (languages and semantics), algorithms (design of constraint solvers), and applications (modeling and resolution of problems using constraint techniques). The book is intended for graduate students, researchers, and practitioners in diverse areas of computer science. It is complemented by a web-page containing research papers, teaching material such as tutorials, slides and exercises, software such as constraint solvers, and references.

The book consists of 17 chapters and an appendix, distributed over three parts: the first part presents classes of constraint programming languages, the second part introduces some types of constraints together with the algorithms to solve them, and the last part discusses three applications.

In the first part, classes of constraint programming languages are presented in a uniform way by presenting in turn a calculus (abstract syntax and operational semantics), declarative semantics, and the correspondence between operational and declarative semantics (soundness and completeness). Classes of languages are extended step by step: first logic programming, then constraint logic programming, concurrent constraint logic programming, and finally Constraint Handling Rules (CHR), which is used to specify, design, and implement constraint solvers presented in Part 2 and 3. The appendix gives the syntax and semantics of first-order logic that constitutes the formal basis of this book.

Part 2 is devoted to constraint solvers. The first chapter presents the notion of constraint systems to specify the syntax and semantics of the constraints of interest. Then, capabilities of constraints solvers are enumerated before discussing principles of constraint solving techniques. The remainder of the second part is divided into 5 chapters, each of them presenting in a uniform way a common constraint system together with a solver (and its specification and implementation in the CHR language), and an example of a typical application. The constraint systems that are discussed are Boolean algebra, rational trees, linear polynomial equations, finite domains, and non-linear equations.

After a very brief description of the commercial market of constraint programming, the last part of the book presents three applications representing three classes of use of constraint techniques: optimal placement, reasoning with incomplete information, and timetabling. The first application deals with optimal placement for wireless communication; the second application is the Munich rent advisor for estimating the maximum fair rent for a flat; and the last one is a university course timetabling problem.

The approach of this book is very interesting: it first tackles theoretical foundations in terms of languages and semantics, before describing some constraint systems and some specifications and implementations of associated solvers, and finally some applications using the same formalism. From beginning to end, the book goes very smoothly and gradually from theory to some real life applications. Since languages and semantics are presented in a uniform and incremental way starting from logic programming, it is very simple to understand what are the commonalities and differences of the languages with their respective extensions (constraint, concurrency). However, the reader is expected to be familiar with first-order logic although the appendix introduces first-order languages. In a way, the title of the book is perhaps a bit misleading and could mention the word “logic”: much research about constraints is not related to logic.

Another interesting feature of the book is that it is completed by a web page which is very useful to find some more information. For those considering the book as a textbook for graduate students, the web page provides slides, tutorials, and exercises that come very handy. For a practitioner, the web page provides numerous examples of solvers written in the CHR language, and a large number of links to some applications. For a researcher, the web page is full of links related to constraint programming, or more precisely, related to the CHR language.

All along the book, a panel of examples is given to illustrate solvers, techniques, and applications of constraint programming. Most of the examples are very short, simple, and very explanatory. Moreover, they are different and cover several application domains of constraint programming. However, the limits of the constraint systems (in terms of expressiveness) and the limits of solvers (in terms of possible and potential applications) are not discussed.

One could say that this book is about constraint programming with the CHR language. Indeed, all the examples and programs are given in the CHR language, and the first part of the book leads to the CHR language syntax and semantics. However, the CHR language produces concise, short, readable, and easily understandable programs.

CHR comes in handy not only to implement but also to design solvers at a high level of abstraction, and thus, CHR can be used as a specification language.

In my opinion, the few shortcomings mentioned above are very minor. The book was written by experts in constraint programming, and one can clearly see in the book the experience of the authors in teaching constraints and the validity and clarity of their approach. This is a very valuable book that I wholeheartedly recommend to students as well as researchers and practitioners interested in the fields of constraint programming, declarative programming, logic programming, rule-based programming, and problem solving.

Eric Monfroy

*IRIN, University of Nantes, France*