

## *Book review*

*A Gentle Guide to Constraint Logic Programming via ECL<sup>i</sup>PS<sup>e</sup>* by Antoni Niederliński, xiii + 509 pages, published by Jacek Skalmierski Computer Studio, Third Edition, 2014. Also available at <http://www.anc1p.pl/> as a free secured PDF file.

This book is a detailed and deep introduction to modeling (and solving) combinatorial and continuous decision problems (both for finding feasible solutions and for optimization) using constraint logic programming (CLP). The approach is neither formal nor theoretical but operational and intuitive; the book discusses numerous problems (puzzles, real-world problems such as OR problems, ...) and gives the corresponding models and solutions in the freely available ECL<sup>i</sup>PS<sup>e</sup> CLP platform. It is also complemented by numerous exercises. The keywords of this book are really **modeling** (defined by the author as "translating verbal problem statements into Prolog or CLP programs") and **operational**.

I feel this book is intended for beginners who would have to read the book linearly, as well as for intermediates who could skip some chapters to focus more on other ones, and also for advanced who could analyze more deeply some problems, and for teachers looking for exercises. But in none of the previous cases it is aimed for someone looking for definitions and abstractions: this book definitively gives an intuitive and operational understanding of the topic.

The book is organized in 7 chapters. The first chapter is an introduction to CLP and its context. It finishes with an interesting classification of problems (feasible or optimal solution, states or state trajectories) which is used in the rest of the book.

Since its basic concepts also appear in CLP, the second chapter presents Prolog. This is also a nice way to introduce exhaustive search.

The next chapters focus on CLP. Chapter 3 and 4 are a discussion about computing feasible solutions. Chapter 3 introduces the basics (notion of domains, backtracking strategies, constraint propagation, basic constraints, search, ...) while Chapter 4 presents the most often used global constraints (e.g., `alldifferent`, `occurrences`).

The next two chapters deal with optimal solutions. The pattern is the same as for Chapter 3 and 4: Chapter 5 introduces optimization problems and search based on branch-and-bound approach, together with the basic built-ins to model them in ECL<sup>i</sup>PS<sup>e</sup> while Chapter 6 goes deeper into optimization using global constraints such as `cumulative` and `disjunctive`.

The last chapter is devoted to the modeling of problems which require variables with continuous domains, either to compute feasible solutions or optimal solutions. In my opinion this chapter is a bit weaker and less successfully completed than the

other ones (mainly in terms of presented problems). But I feel this is mainly due to the capacities of ECL<sup>i</sup>PS<sup>e</sup> with respect to continuous variables.

The range of illustrating examples is very wide. It includes models for computing feasible and optimal states (solutions or optimal solutions) and state trajectories (sequences or optimal sequences of well-defined states). Problems range from puzzles (such as the perennial "farmer-wolf-goat-cabbage" for state trajectories problems, or the cryptarithmic "send+more=money" for feasible state) and classical problems (such as Hanoi tower, or N queen problem) to OR problems (such as TSP, car sequencing, or job-shop). For some problems, various models are given, either to improve the models themselves (e.g., introducing new built-ins), to treat variations of the problem (e.g., computing a feasible solution or an optimal solution), or to present different approaches for modeling (e.g., integer programming-like or CLP-like approaches). The progression of the book is constant and the problems are of increasing complexity. Numerous figures and tables illustrate and clarify several notions and examples.

I think that to capture all the content of this book, the reader must have at his/her disposal a pen and a paper to sketch the models, and the ECL<sup>i</sup>PS<sup>e</sup> system to run them.

I do not always use the same vocabulary as the author (e.g., when he writes about path-consistency, I would rather say generalized arc consistency, or sometimes when he talks about consistency techniques, I would have used search space reduction or constraint propagation). But although there is no formal definition, the given explanations are sufficient to make such slight vocabulary ambiguities vanish.

Although this book focuses on modeling and not really on solving, I would have expected some more information about the solving process, for example when presenting variants of the same problem or when introducing global constraints such as `alldifferent` (they are not only syntactic shorthands but also result in more efficient constraint propagation algorithms).

Antoni Niederliński, the author, has been teaching Prolog and CLP for about 30 years at Polish universities. His educational experience and his desire to share his knowledge and expertise with others make him a natural candidate for writing such a book about modeling.

I warmly recommend this book to beginners, and intermediate and advanced users who want to learn or to go deeper into modeling of decision problems using CLP. I also recommend this book to those wishing to learn CP (Constraint Programming). In fact, the ECL<sup>i</sup>PS<sup>e</sup> built-ins and programs that appear in this book are very similar to the ones of an imperative CP system.

Eric Monfroy

LINA, UMR CNRS 6241, TASC INRIA, Université de Nantes, France

(*e-mail*: `Eric.Monfroy@univ-nantes.fr`)