Book review


The book provides a comprehensive overview of several logic programming formalisms based on the language of normal logic programs and its extensions that allow disjunctions in the head and both strong and default negations. The most broadly accepted semantics for programs in such general syntax is the semantics of answer sets, which generalizes the stable-model semantics of normal logic programs. Recent research demonstrated that logic programs, when interpreted by the answer-set semantics, can be used in declarative programming in a way that abandons some major tenets of the standard logic programming as implemented and exemplified by Prolog. Namely, to solve a problem the programmer constructs a program so that problem solutions are represented by answer sets of the program, and not by variable substitutions. Computing models and answer sets, rather than resolution refutation proofs, becomes the fundamental computational task and a uniform control mechanism.

The book refers to logic programming formalisms that are so used by a single term AnsProlog (with some modifiers specifying the exact set of connectives that are allowed). It stands for Programming in logic with Answer sets and is meant to stress that one programs by writing theories in the language of logic and to emphasize a shift in focus from proofs and variable substitutions to answer sets (models).

In the text, the author introduces AnsProlog, presents basic theoretical results and addresses the issue of programming methodology. A substantial fragment of the book deals with applications of AnsProlog in solving combinatorial and constraint satisfaction problems, and in knowledge representation. In the latter case, the author focuses on applications to reasoning about action and planning.

The book consists of nine chapters and two appendices.

1. Declarative programming in AnsProlog*: introduction and preliminaries
2. Simple modules for declarative programming with answer sets
3. Principles and properties of declarative programming with answer sets
4. Declarative problem solving and reasoning in AnsProlog*
5. Reasoning about actions and planning in AnsProlog*
6. Complexity, expressiveness, and other properties of AnsProlog* programs
7. Answer set computing algorithms
8. Query answering and answer set computing systems
9. Further extensions of and alternatives to AnsProlog*
Appendix A: Ordinals, lattices, and fixpoint theory
Appendix B: Turing machines

Chapter 1 contrasts declarative and procedural programming and points to the importance of the former for applications in artificial intelligence. It provides an introduction to the syntax and the semantics of AnsProlog. Chapter 2 shows several small programs solving problems and subproblems frequently found in combinatorial and constraint satisfaction problems (generating enumerations and orderings, imposing cardinality and weight constraints, checking satisfiability of propositional formulas) and in knowledge representation (modeling normative statements, the frame problem). Chapter 3 discusses basic theory of AnsProlog presented from the perspective of systematic program development. Chapter 4 discusses applications of AnsProlog in solving combinatorial and constraint satisfaction problems. It also shows AnsProlog representations of prioritized default theories and inheritance hierarchies. Chapter 5 deals with reasoning about action and planning and develops AnsProlog representations for several versions of problems in these domains. The complexity of reasoning with several classes of AnsProlog programs and the expressive power of the corresponding languages is the subject of Chapter 6. This chapter also deals with the issues of compactness of representation and compilability, and relates AnsProlog to other knowledge representation formalism such as circumscriptive, autoepistemic logic and default logic. Chapter 7 presents algorithms to compute answer sets, as well as the well-founded semantics of AnsProlog programs (the latter is important as it approximates the answer-set semantics and is fast to compute). Chapter 8 presents most advanced implementations of AnsProlog: smodels and dlv. The last chapter deals with extensions of AnsProlog (for instance, allowing nested expressions) as well as with additional applications. The appendices provide material needed to make the book self-contained.

This book is an important milestone in the development of logic programming as a knowledge representation tool. It is the first book that treats seriously a possibility of using logic programming with answer-set semantics as a declarative programming tool and a knowledge representation formalism and does it in a systematic and comprehensive way. It presents all major results concerning that formalism. It comes with an impressive number of examples and example programs. Especially important from the perspective of applications and ultimate acceptance of the paradigm of answer-set programming is an elegant treatment of reasoning about action and planning problems.

The book is a very good reference text. Sections containing notes and comments provide good insights into the time line of the development of ideas and provide pointers to all relevant bibliography. It will be greatly appreciated by the researchers in the field.

This text is an excellent research monograph that will certainly stimulate and affect further research efforts in the area of answer-set programming. It is also a text that could be used in the classroom for teaching declarative problem solving and knowledge representation.

The book is not free of shortcomings, which is perhaps not surprising given
the novelty of the subject and the size of the text. I felt that the organization of the material might be improved. For instance, it is not clear why the material on default logic and inheritance networks was included in Chapter 4 and, similarly, why connections of AnsProlog with circumscription, default logic and autoepistemic logic were dealt with in Chapter 6. The well-founded semantics is referred to in Chapter 7 and, in fact, is needed there to put some of the algorithms in the proper context, yet its discussion is delayed to Chapter 9.

As I mentioned, earlier, the text contains numerous example programs. While intuitively, these programs are correct, I missed formal correctness proofs. They could easily have been provided at least for some of the programs (for instance, those presented in Chapter 4). It is even more important if one remembers that one of the selling points of declarative programming is that it facilitates program correctness verification.

The book does an excellent job demonstrating that AnsProlog is a powerful modeling language. Relatively little attention was devoted to matters of computational effectiveness. I would welcome comments indicating the size of problems that can be handled by current generation of AnsProlog implementations. In this context, I felt the book did not give enough justice to extensions of AnsProlog allowing explicit modeling of cardinality and weight constraints. After all, it is well known that they are critical for the success of the current smodels implementation and without them, the performance suffers.

Lastly, the title of the book is perhaps somewhat misleading. It suggests a general text on declarative programming and knowledge representation while the book is really about AnsProlog and its applications.

However, all these, overall minor, shortcomings take second place to what is really important here. The area of declarative programming with the focus on models and not proofs has its first monograph. The presentation is formal yet the text reads well. It is comprehensive and provides an exhaustive collection of references and examples. It also emphasizes the importance of programming methodology and builds its rudiments. I wholeheartedly recommend this book to researchers and students in the fields of logic programming, declarative programming and knowledge representation.

MIROŚLAW TRUSZCZYŃSKI
University of Kentucky, Lexington, USA