KNOWLEDGE REPRESENTATION, REASONING, AND THE DESIGN OF INTELLIGENT AGENTS

Knowledge representation and reasoning is the foundation of artificial intelligence, declarative programming, and the design of knowledge-intensive software systems capable of performing intelligent tasks. Using logical and probabilistic formalisms based on Answer Set Prolog (ASP) and action languages, this book shows how knowledge-intensive systems can be given knowledge about the world and how it can be used to solve non-trivial computational problems. The authors maintain a balance between mathematical analysis and practical design of intelligent agents. All the concepts, such as answering queries, planning, diagnostics, and probabilistic reasoning, are illustrated by programs of ASP and its extensions. The text can be used for AI-related undergraduate and graduate classes and by researchers who would like to learn more about ASP and knowledge representation.

Dr. Michael Gelfond is a Professor of Computer Science at Texas Tech University. He received his PhD from the Institute of Mathematics of the Academy of Sciences, St. Petersburg, Russia. He is an AAAI Fellow and serves as an Area Editor for the *International Journal of Theory and Practice of Logic Programming* and as an Executive Editor of the *Journal of Logic and Computation*. In 2004 and 2012 he was the recipient of the award for most influential paper in twenty years by the International Association of Logic Programming.

Yulia Kahl is a member of the Texas Action Group and the Knowledge Representation Lab at Texas Tech. She received her Master's in Computer Science, focusing on the use of Answer Set Prolog for planning. She has also worked as a programmer at IBM.

KNOWLEDGE REPRESENTATION, REASONING, AND THE DESIGN OF INTELLIGENT AGENTS

The Answer-Set Programming Approach

MICHAEL GELFOND
Texas Tech University

YULIA KAHL

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

32 Avenue of the Americas, New York, NY 10013-2473, USA

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107029569

First published 2014

Printed in the United States of America

A catalog record for this publication is available from the British Library.

Library of Congress Cataloging in Publication Data
Gelfond, Michael.
Knowledge representation, reasoning, and the design of intelligent agents : the answer-set programming approach / Michael Gelfond, Texas Tech University, Yulia Kahl.
pages cm
Includes bibliographical references and index.
ISBN 978-1-107-02956-9 (hardback : alk. paper)
1. Intelligent agents (Computer software) I. Kahl, Yulia, 1970– II. Title.
QA76.76.I58G45 2014
006.3–dc23 2013029651

ISBN 978-1-107-02956-9 Hardback

To Lara and Patrick, with love

Contents

Preface

This is a book about knowledge representation and reasoning (KRR) – a comparatively new branch of science that serves as the foundation of artificial intelligence, declarative programming, and the design of intelligent agents – knowledge-intensive software systems capable of exhibiting intelligent behavior. Our main goal is to show how a software system can be given knowledge about the world and itself and how this knowledge can be used to solve nontrivial computational problems. There are several approaches to KRR that both compete with and complement each other. The approaches differ primarily by the languages used to represent knowledge and by corresponding computational methods. This book is based on a knowledge representation language called Answer Set Prolog (ASP) and the answer-set programming paradigm – a comparatively recent branch of KRR with a well-developed theory, efficient reasoning systems, methodology of use, and a growing number of applications.

The text can be used for classes in knowledge representation, declarative programming, and artificial intelligence for advanced undergraduate or graduate students in computer science and related disciplines, including software engineering, logic, and cognitive science. It will also be useful to serious researchers in these fields who would like to learn more about the answer-set programming paradigm and its use for KRR. Finally, we hope that it will be of interest to anyone with a sense of wonder about the amazing ability of humans to derive volumes of knowledge from a collection of basic facts. Knowledge representation and reasoning, located at the intersection of mathematics, science and humanities, provides us with mathematical and computational models of human thought and gives some clues to the understanding of this ability. The reader is not required to know logic or to have previous experience with computational systems. However, some understanding of the mathematical method of thinking will be of substantial help.

We have attempted to maintain a proper balance between mathematical analysis of the subject and practical design of software systems capable

of using knowledge about their environment to perform intelligent tasks. Beginning with simple question-answering agents, we progress to more and more complex ones and explain how important problems of knowledge representation and reasoning such as commonsense (default) reasoning, planning, diagnostics, and probabilistic reasoning are solved with ASP and its extensions. The precise mathematical definitions of basic concepts are always accompanied by informal discussions and by examples of their use for modeling various computational tasks performed by humans. Readers are encouraged to run programs to test their agent's ability to perform these tasks using available, state-of-the-art ASP reasoning systems.

Of course the worth of a particular KRR theory is tested by the ability of software agents built on the basis of this theory to behave intelligently. If, given a certain amount of knowledge, the agent exhibits behavior that we believe reasonable for a human with exactly the same knowledge, we deem the theory to be a step in the right direction.

We hope that serious readers will learn to appreciate the interplay between mathematical modeling of a phenomenon and system design, and will better understand the view of programming as the *refinement of specifications*. Even though the book does not discuss large practical projects, we believe that the skills learned will be of substantial use in the marketplace.

We meant for this book to be a foundation for those interested in KRR. Our desire to limit the material to one semester forced us to skip many topics closely related to our approach, such as various constraint and abductive logic programming languages and algorithms, descriptions of important ASP reasoning methods, the methodology of transforming ASP programs to improve their efficiency, and a large body of useful mathematical knowledge. The book does not cover other important KRR topics such as descriptive logics and their use for the Semantic Web and other applications, natural-language processing, and the like. To help those interested in building on the foundation presented and learning more about these other topics, we have included a section on references and further reading in each chapter. (These lists of references, however, are not even close to complete – the goal is simply to start the readers on their learning adventure.)

The first two chapters introduce the reader to the logic-based approach to agent design and to the main tool we use to create the agent's knowledge base – Answer Set Prolog. The third chapter discusses the roots of ASP and briefly covers several other important knowledge representation formalisms. The next two chapters are concerned with the use of ASP for design and implementation of question-answering agents. Chapter 4 illustrates the use of (possibly recursive) ASP definitions of relations for

building simple knowledge bases that allow incompleteness of information and a hierarchical organization of knowledge. Chapter 5 covers ASP representation of defaults and their exceptions – one of the distinctive features of ASP that makes it suitable for formalization of defeasible, commonsense arguments. In Chapter 6 our agents become more powerful as we show how various problems related to nontrivial search can be reduced to ASP reasoning. Chapter 7 discusses implementations of ASP solvers used to perform such reasoning. Chapters 8–10 deal with agents capable of performing actions and functioning intelligently in dynamic (changing) environments. The material includes discussion of action languages – tools for concisely describing such environments – and ASP-based reasoning algorithms that allow the agents to perform complex reasoning tasks including planning and diagnostics. Chapter 11 introduces a dialect of ASP capable of representing and reasoning with probabilistic knowledge. The emphasis here is on combining logical and probabilistic reasoning and the use of formal language to clarify and expand some classical probabilistic notions, including that of the behavior of probability under updates of the agent's knowledge base. Finally, in Chapter 12, we give a brief introduction to a traditional logic programming language called Prolog, which can be viewed as a special case of ASP. The appendices are meant to help the reader run ASP code using available ASP solvers. Sections marked with an asterisk are optional.

Preliminary versions of the book were used by the first author for teaching graduate and undergraduate classes in AI and intelligent systems at Texas Tech University. In some cases, the constraint of a single semester made it necessary to omit or only present highlights of Chapters 3 and 12.

We would like to conclude by acknowledging the people who knowingly or unknowingly contributed to this book. The first author was fortunate to learn about declarative programming and the logical approach to AI from work and occasional conversations with such founders of these fields as Bob Kowalski, John McCarthy, Jack Minker, and Ray Reiter. Without them this book would not have been possible. His life-long friendship and collaboration with Vladimir Lifschitz played a crucial role in his development as a scientist. The second author is grateful to her teachers, especially her advisor, Chitta Baral, who allowed her to pursue her interest in KRR and ASP. Also, she feels truly blessed to have been allowed to be a part of the Knowledge Representation Lab, first at the University of Texas at El Paso and now at Texas Tech University. The seminars conducted there and all the great people she met helped her maintain contact with the field that she has found exciting and rewarding.

We thank Vladimir Lifschitz and other colleagues, collaborators, friends, and students, including Evgenii Balai, Marcello Balduccini, Chitta Baral, Matt Barry, Justin Blount, Piero Bonatti, Marc Denecker, Thomas Eiter, Alfredo Gabaldon, Gregory Gelfond, Georg Gottlob, Daniela Inclezan, Patrick Kahl, Nicola Leone, Jorge Lobo, Viktor Marek, Veena Mellarkord, Ricardo Morales, Ilkka Niemella, Monica Nogueira, Mauricio Osorio, Ramon Otero, David Pearce, Alessandro Provetti, Halina and Teodor Przymusinski, Nelson Rushton, Torsten Schaub, Richard Scherl, Tran Cao Son, Yana Todorova, Bonnie Trailor, Miroslaw Truszczynski, Richard Watson, Yuanlin Zhang, and Weijun Zhu; they helped us develop a better understanding of the topics covered. Many of them also provided valuable comments on the book. The authors owe a special thanks to Pedro Cabalar, Vinay K. Chaudhri, Sandeep Chintabathina, Mauricio Osorio, Tran Cao Son, Richard Watson, and Yuanlin Zhang who used the earlier drafts of this book as primary or supplementary material for AI-related courses. Their feedback is greatly appreciated.

Thanks also to the developers of ASPIDE – Onofrio Febbraro, Kristian Reale, and Francesco Ricca – for including the query interface and its epistemic mode so that their program could be used to simulate the kind of dialogue we wanted to have with the computer. Thanks also to Nicola Leone and his team at DLVSystems who supported the development of these features. Everyone has been very helpful!

We also thank the great people at Cambridge University Press for their support. Ada Brunstein, Lauren Cowles, and Dana Bricken have been very helpful throughout the publishing process. We also truly appreciate the great feedback from the anonymous reviewers who gave valuable time to help us improve the book. The people at Springer-Verlag have been kind in letting us use the parts assembly problem from Clocksin and Mellish's wonderful *Programming in Prolog*, and we thank them kindly.

Michael wishes to thank his wife, Larisa, for constant encouragement and support that helped him go back to research after emigrating from the Soviet Union and to enjoy his work ever since. Without her this book would never have been written.

Yulia wishes to thank her husband, Patrick, who has always given her love, support, and the freedom to do what she loves. Without him, her part in the writing of this book would have been nothing but a dream. She also wishes to thank her wonderful parents, Michael and Larisa, without whom she would not have had the joy of working on this book.