Preface

This volume is concerned with the *classical planning problem*, which can be informally defined as follows:

> Given a description of the current situation (an *initial state*) of an agent, the means by which the agent can alter this situation (a set of *actions*) and a description of desirable situations (a *goal*), find a sequence of actions (a *plan*) that leads from the current situation to one which is desirable.

Instances of the classical planning problem, called *planning tasks*, can model all kinds of abstract reasoning problems in areas as diverse as elevator control, the transportation of petroleum products through pipeline networks, or the solution of solitaire card games.

These applications have little in common apart from the fact that, at a suitable level of abstraction, they can be precisely modelled using the notions of initial states, actions, and goals. In the elevator example, the initial state is given by the current location of the elevator and the locations of passengers waiting at different floors to board the elevator. The set of actions comprises movements of the elevator between different floors along with the ensuing activities of passengers boarding and leaving the elevator. The goal specifies a destination floor for each passenger. In the pipeline example, the initial state describes the initial contents of the pipelines and of the areas they connect. Actions model the changes in the contents of a pipeline as products are pumped through it, and a typical goal requires a certain amount of petroleum product to be available in a certain area. In the card game example, the initial state is given by a randomly dealt card tableau. The set of actions models the different ways of moving cards between piles that are allowed by the rules of the game. The goal consists of achieving a certain arrangement of the cards.

Because planning is not limited to a particular application area, or indeed any finite set of application areas, it is an example of *general problem solving*, and in fact the planning problem was first introduced to the Artificial Intelligence community under that name. Classical planning has been an active research area for about half a century, with Newell and Simon's work on the General Problem Solver [94] usually seen as a starting point. A historical perspective on planning research is provided by a collection of classical papers edited by Allen et al. [1] and a more recent survey by Weld [112].

One of the well-established facts about the planning problem is that it is *hard*. In the very general variants commonly studied in the early days of planning research, such as the original STRIPS formalism [40, 84], it is known to be undecidable [39]. In the more restricted formalisms typically considered today, the problem is still at least PSPACE-hard [19].

How do we solve such a hard problem? In his 1945 classic *How to Solve It*, mathematician George Pólya describes a four-step strategy to problem solving. Here is the first and most important step:

> "First, you have to *understand* the problem." [99, p. 5]

This is solid general advice. To solve the planning problem, that is to design efficient planning algorithms, it is important to understand it. Is planning difficult? Can we *prove* that it is difficult? Are there relevant special cases which are easier to solve than others?

This volume contributes to the understanding of the planning problem by formally analyzing those special cases which have attracted most attention in the past decade, namely, the standard *benchmark domains* of the International Planning Competitions [8, 66, 86, 87, 91]. This is the topic of Part I, *Planning Benchmarks.*

Because planning is general *problem* solving, planning tasks are commonly called planning *problems* in the literature. This implies that Pólya's recommendation is equally applicable to *their* solution: To solve a planning task, one has to understand it. Without any kind of intuition of which actions are useful for achieving the goals in a certain situation, the problem solver is more or less limited to blindly exploring the space of possible solutions, which is usually a fruitless endeavour. Given that we are interested in *algorithmic* approaches to planning, this "understanding" must be arrived at algorithmically. One well-established approach to *informed* planning algorithms is the use of heuristic search techniques [16, 68, 90]. (Not entirely coincidentally, Pólya's work is also responsible for introducing the word *heuristic* into modern scientific discourse, although not quite with the meaning in which it is generally used in Artificial Intelligence these days.)

This volume contributes to the practice of solving planning tasks by presenting a new approach to heuristic planning based on two central ideas: reformulating planning tasks into a form in which its logical structure is more apparent than in the original specification, and exploiting the information encoded in the *causal graphs* and *domain transition graphs* of these reformulated tasks. This is the topic of Part II, *Fast Downward.*

This volume is a revised version of my doctoral thesis, *Solving Planning Tasks in Theory and Practice* [61], submitted to Albert-Ludwigs-Universität Freiburg in June 2006. It has been a long time in the making, with the first ideas conceived in 1998. Over the years, many people have contributed to the work in some way or other, and I would like to use this opportunity to thank them.

Throughout my PhD studies, which began in 2001, Bernhard Nebel served as my advisor. In addition to providing a perfect work environment, his advice has always been very helpful. While giving me lots of freedom to pursue the scientific topics I was interested in, he pushed me at the right times and with the right amount of force to actually get the work done, which is a crucial contribution.

Sylvie Thiébaux served as the second reviewer of the thesis and gave me some exceptional feedback. I very strongly doubt that anyone else will ever read the thesis as closely as her, myself included.

This work has greatly benefited from feedback and discussions with many people in the AI planning community and some people outside. In addition to Bernhard and Sylvie, I particularly want to thank Carmel Domshlak, Stefan Edelkamp, Maria Fox, Inge Li Gørtz, Jörg Hoffmann, Derek Long, Silvia Richter, Jussi Rintanen and Menkes van den Briel for their scientific input.

Many of the results in Chaps. 4 and 5 build on the work of former students whose work I supervised. I want to thank Michael Drescher, Robert Mattmüller and Gabi Röger for their contributions.

I already mentioned the perfect work environment, and I want to use the opportunity to thank all my former and current colleagues at the Foundations of Artificial Intelligence group at Albert-Ludwigs-Universität Freiburg for useful discussions and for having a lot of fun together. Special thanks go to my officemates throughout the years, in chronological order: Rudi Triebel, Yacine Zemali, Christian Köhler, Yuliya Lierler, Marco Ragni, Yannis Dimopoulos and Sebastian Kupferschmid.

For their help – and patience – in the preparation of this volume, I want to thank Ursula Barth and Frank Holzwarth from Springer. Extra special thanks go to Gabi Röger, who undertook the very laborious task of preparing the keyword index.

Last but certainly not least, for moral support my heartfelt thanks goes to Stefan Franck, Sebastian Kupferschmid, Silvia Richter, Anna Seesjärvi and Libor Valevsky and to my family Gundula, Michael and Volker Helmert.

Freiburg, November 2007 Malte Helmert