Lutz Frommberger

Qualitative Spatial Abstraction in Reinforcement Learning

Springer

Cognitive Technologies

For further volumes:
http://www.springer.com/series/5216

Lutz Frommberger

Qualitative Spatial Abstraction in Reinforcement Learning

Springer

Dr.-Ing. Lutz Frommberger
Cognitive Systems Group
Department of Mathematics and Informatics
University of Bremen
P.O. Box 330 440
28334 Bremen
Germany
lutz@informatik.uni-bremen.de

This thesis was accepted as doctoral dissertation by the Department of Mathematics and Informatics, University of Bremen, under the title "Qualitative Spatial Abstraction for Reinforcement Learning". Based on this work the author was granted the academic degree Dr.-Ing.

Date of oral examination: 28th August 2009

Reviewers:

Prof. Christian Freksa, Ph.D. (University of Bremen, Germany)
Prof. Ramon López de Mántaras, Ph.D. (Artificial Intelligence Research Institute, CSIC, Barcelona, Spain)

Cognitive Technologies ISSN 1611-2482
ISBN 978-3-642-16589-4 e-ISBN 978-3-642-16590-0
DOI 10.1007/978-3-642-16590-0
Springer Heidelberg Dordrecht London New York

ACM Computing Classification: I.2

Cover design: KünkelLopka GmbH, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Teaching and learning are difficult tasks not only when people are involved but also with regard to computer programs and machines: When the teaching/learning units are too small, we cannot express sufficient context to teach a differentiated lesson; when they are too large, the complexity of the learning task can increase dramatically such that it will take forever to teach and learn a lesson. Thus, the question arises, how we can teach and learn complex concepts and strategies, or more specifically: How can the lesson be structured and scaled such that efficient and effective learning can be achieved?

Reinforcement learning has developed as a successful learning approach for domains that are not fully understood and that are too complex to be described in closed form. However, reinforcement learning does not scale well to large and continuous problems; furthermore, knowledge acquired in one environment cannot be transferred to new environments. Although this latter phenomenon also has been observed in human learning situations to a certain extent, it is desirable to generalize suitable insights for application also in new situations.

In this book, Lutz Frommberger investigates whether deficiencies of reinforcement learning can be overcome by suitable abstraction methods. He discusses various forms of spatial abstraction, in particular qualitative abstraction, a form of representing knowledge that has been thoroughly investigated and successfully applied in spatial cognition research. With his approach, Lutz Frommberger exploits spatial structures and structural similarity to support the learning process by abstracting from less important features and stressing the essential ones. The author demonstrates his learning approach and the transferability of knowledge by having his system learn in a virtual robot simulation system and consequently transferring the acquired knowledge to a physical robot.

Lutz Frommberger's approach is influenced by findings from cognitive science. In this book, he focuses on the role of knowledge representation for the learning process: Not only is it important to consider *what* is represented, but also *how* it is represented. It is the appropriate representation of an agent's perception that enables generalization in the learning task and that allows for reusing learned policies in new contexts—without additional effort. Thus, the choice of spatial representation for the agent's state space is of critical importance; it must be well considered by the designer of the learning system. This book provides valuable help to support this design process.

Bremen, September 2010 *Christian Freksa*

Preface

Abstraction is one of the key capabilities of human cognition. It enables us to conceptualize the surrounding world, build categories, and derive reactions from these categories to cope with different situations. Complex and overly detailed circumstances can be reduced to much simpler concepts, and not until then does it become feasible to deliberate about conclusions to draw and actions to take.

Such capabilities, which come easily to a human being, can still be a big challenge for an artificial agent: In the past years of research I investigated how to employ such human concepts in a learning machine. In particular, my research focused on utilizing spatial abstraction techniques in agent control, using the machine learning paradigm of reinforcement learning. This led to results published in journals and conference proceedings over the years that are now integrated and significantly extended to a comprehensive study on spatial abstraction in reinforcement learning in this book. It is spans the whole range from formal aspects to empirical results.

Reinforcement learning allows us to learn successful strategies in domains that are too complex to be described in a closed model or in cases where the system dynamics are only partially known. It has been shown to be effectively applicable to a large number of tasks and applications. However, reinforcement learning in its "pure" form shows severe limitations in practical use. In particular, it does not scale well to large and continuous problems, and it does not allow for reuse of already gained knowledge within the learning task or in new tasks in unknown environments. Spatial abstraction is an appropriate way to tackle these problems.

When regarding the nature of abstraction, I believe that only a consistent formalization of abstraction allows for a thorough investigation of its properties and effects. Thus, I present formal definitions that distinguish between three different facets of abstraction: aspectualization, coarsening, and conceptual classification. Based on these definitions it can be shown that aspectualization and coarsening can be utilized to achieve the same effect. Hence, the process of aspectualization is to be preferred when using spatial abstraction in agent control processes, as it is computationally simple and its features are easily accessible. This allows for coping even with high-dimensional state spaces. The property of a representation being aspectualizable turns out to be central for agent control.

In order to use abstraction to control artificial agents, I argue for an action-centered view on abstraction that concentrates on the decisions being drawn at certain states. I derive criteria for efficient abstraction in agent control tasks and show that these criteria can most satisfactorily be matched by the use of qualitative representations, especially when they model important aspects in the state space such that they can be accessed by aspectualization.

In sequential decision problems we can distinguish between goal-directed and generally sensible behavior. The corresponding spatial features form task space and structure space. As it is of special importance to describe structural elements of the state space explicitly in an abstract spatial representation, I introduce the concept of structure space aspectualizable observation spaces. For this kind of state space, two methods are developed in this book: task space tile coding (TSTC) and a posteriori structure space transfer (APSST). They allow for reusing structural knowledge while learning to solve a task and also in different tasks in unknown environments. Furthermore, I introduce structure-induced task space aspectualization (SITSA), a mechanism for situation-dependent spatial abstraction based on knowledge gained from a structural analysis of learned policies in previous tasks.

We will study the effect of the proposed techniques on an instance of structure space aspectualizable state spaces, namely le-RLPR, an abstract spatial representation tailored for robot navigation in indoor environments. It describes the circular order of landmarks around the moving robot and the relative position of walls with regard to the agent's moving direction. Compared to coordinate-based metrical approaches, le-RLPR enables us to learn successful strategies for goal-directed navigation tasks considerably faster. Policies learned with le-RLPR also allow for generalization within the actual learning task as well as for transferring knowledge to new scenarios in unknown environments. As a final demonstration we will see that RLPR-based policies learned in a simulator can also be transferred to a real robotics system with little effort and allow for sensible navigation behavior of a robot in office environments.

Acknowledgments

At this point I want to express my gratitude to several people who helped me during my work on this book.

First of all, I thank Christian Freksa for advising my doctoral thesis and giving me the opportunity to work in the Cognitive Systems research group at the University of Bremen. He brings together people from various scientific fields for interdisciplinary research. This provides an inspiring and productive atmosphere, and I am thankful that I was involved there for so many years. Christian has been always available when I needed advice. Often, his comments and ideas made me look at my work from a different point of view and thus broadened my mind.

Furthermore, I want to express my gratitude to Ramon López de Mántaras for his willingness to be a reviewer of my doctoral thesis and especially for his enthusiasm and his detailed and encouraging remarks on my work.

Particularly in the early stages of my research it had been important to receive encouraging feedback on the ideas I had. In particular, I thank Reinhard Moratz for initially supporting my approach. Furthermore, I thank Joachim Hertzberg, Frank Kirchner, Martin Lauer, George Konidaris, and Stefan Wölfl for inspiring and encouraging discussions that helped me to focus my work. Also, several anonymous reviewers provided substantial feedback on papers emerging from ongoing work on this book that I submitted to workshops, conferences, and journals.

Martin Riedmiller sparked my interest in reinforcement learning when I was a student at the University of Karlsruhe. I thank him for repeatedly giving me the opportunity to extensively discuss my work with him and his Neuroinformatics group at the University of Osnabrück. I acknowledge especially Stephan Timmer's valuable comments and hints regarding my approach.

I notably enjoyed working with my colleagues at the Cognitive Systems group, who gave me lots of feedback over the years. Especially, the graduate seminar was a great opportunity for inspiring discussions. I thank Diedrich Wolter for constantly pushing me forward and his help in making the nasty robot move. Also, I thank Mehul Bhatt, Frank Dylla, Julia Gantenberg, Kai-Florian Richter, Jan Frederik Sima, and Jan Oliver Wallgrün for volunteering to proofread parts of this book. I also thank my student co-workers for their dedication: Fabian Sobotka provided valuable assistance on the implementation of the software and Jae Hee Lee assisted in mathematical formalizations.

Money is not everything, but when available, it helps a lot. I thank the German Research Foundation (DFG) for its financial support of the R3-[Q-Shape] project of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, within which this work was carried out.

Most importantly, I thank my family, Michaela, Mara, and Laila. For many months I dedicated much of my time to writing this book rather than to them. I am deeply grateful for their support, their patience, and their love, without which finalizing this book would have been impossible.

Bremen, September 2010 *Lutz Frommberger*

Contents

Symbols

$\mathcal{A}$	Action space, 10
c_i	Sampled color view, 104
cert	Decision certainty for a state, 89
Conf	Decision confidence for a policy, 85
conf	Decision confidence for a state, 85
$\mathcal{D}$	Arbitrary domain, 45
E	Expected value (in statistics), 13
e	Eligibility trace, 18
H	Horizon, 11
h	Hash function, 76
$\mathcal{I}$	Initiation set for options, 31
I^{κ}	Aspectualization index vector, 52
I^{κ}	Inverse aspectualization index vector, 52
L^*	Set of all detected landmarks, 100
L_i	Sector for landmark selection, 100
$l_{\max}$	Maximum number of landmarks allowed in a sector, 101
$\mathcal{O}$	Observation space, 15
$\mathcal{O}_{\text{NDesc}}$	Set of non-decision structures, 90
$\mathcal{O}_S$	Structure space, 71
$\mathcal{O}_T$	Task space, 71
$p^{\max}$	Vector of maximum feature values, 79
$p_i^{\max}$	Maximum feature value of dimension i, 79
Q	Action-value function, 14
Q^*	Optimal action-value function, 19
Q_S	Structure space Q-function, 83
Q_T	Task space Q-function, 87
q_{π^*}	π^*-preservation quota, 56
$\mathcal{R}$	RLPR grid, 110
R	Reward function, 12
$\mathcal{S}$	State space, 10
s	State, 10

ssd	Structure space descriptor, 72
T	State transition function, 12
tsd	Task space descriptor, 72
V	Value function, 13
V^*	Optimal value function, 13
α	Learning rate, 18
β_i	Landmark sample angle, 104
χ	Tiling function, 76
γ	Discount factor, 13
δ	Temporal difference error, 18
ε	Exploration probability, 17
ζ	Motion noise parameter, 124
Θ	SITSA abstraction function, 90
κ	Abstraction function, 45
λ	Trace decay parameter, 19
μ	Task/structure space weight, 87
ξ	SMDP waiting time parameter, 31
π	Policy, 11
π^*	Optimal policy, 11
π_S	Structure space policy, 72
ρ	Line detection distortion parameter, 140
σ	Landmark noise parameter, 139
τ	RLPR overlap status, 111
$\overline{\tau}$	RLPR overlap status (whole scene), 111
$\overline{\overline{\tau}}$	RLPR overlap status (whole scene, boolean), 111
τ'	RLPR adjacency status, 111
$\overline{\tau}'$	RLPR adjacency status (whole scene), 111
$\overline{\overline{\tau}}'$	RLPR adjacency status (whole scene, boolean), 111
ψ	Observation function, 15
ψ_S	Structure space observation function, 114
ψ_T	Task space observation function, 100
ω	Observation in an observation space, 15
$\overline{\varpi}$	Structure space observation, 83
$\perp$	Empty symbol, 90
$\equiv_2$	modulo 2, 15
$\sim$	Similarity operator, 55

Acronyms

A-CMAC	Averager cerebellar model articulator controller
APSST	A posteriori structure space transfer
CMAC	Cerebellar model articulator controller
MDP	Markov decision process
QSR	Qualitative spatial reasoning
POMDP	Partially observable Markov decision process
RL	Reinforcement learning
RLPR	Relative line position representation
SDALS	Structural-decision-aware landmark selection
SITSA	Structure-induced task space aspectualization
SMDP	Semi-Markov decision process
TSTC	Task space tile coding

Chapter 1
Introduction

One of the most essential properties of a cognitive being is its ability to learn. Learning is the "process of acquiring modifications in existing knowledge, skills, habits, or tendencies through experience, practice, or exercise" (Encyclopædia Britannica, 2007). These modifications lead to a performance improvement of the cognitive being (also called *cognitive agent*) in the tasks it has to solve in its daily routines. Learning provides the agent with a preferably good adaptation of its behavior to the situations it is confronted with.

While most of the learning efforts of human beings and animals are achieved in the early years, learning is generally a life-long process. Perceived situations may change over time, and even the perception abilities themselves may change, and the dynamics of the agent may vary due to age or abrasion. These changes require continuous adaptations of the acquired strategies and behaviors over a longer period of time. It is desirable to find this ability also in artificial cognitive agents, for example, in autonomous robots.

1.1 Learning Machines

Much effort has been spent in the field of artificial intelligence (AI) to investigate methods for machine learning. This field of research spans two distinct paradigms. *Supervised learning* requires external knowledge given by an expert, who supervises the learning process. The learning agent is supposed to find a mapping between its input values and the desired output that is given by the expert. In contrast, *unsupervised learning* autonomously constructs a classification of the input without intervention from outside. Many types of machine learning approaches exist somewhere between supervised and unsupervised learning.

This book concentrates on one of the most influential machine learning techniques: the learning paradigm of *reinforcement learning* (RL) (Sutton and Barto, 1998). In RL, learning does not take place by teaching or supervision, but by interaction with a dynamic and uncertain environment. It can be seen as a form of

L. Frommberger, *Qualitative Spatial Abstraction in Reinforcement Learning*,
Cognitive Technologies, DOI 10.1007/978-3-642-16590-0_1,

weakly supervised learning. The concept of reinforcement learning was addressed very early in psychology and cybernetics and has gained a still increasing popularity in machine learning research over the last two decades. Basically, it implements a mechanism of reinforcing tendencies that lead the system to a "positive" state. Reinforcement learning is trial-and-error learning. Positive reinforcement is only given when the system reaches a well-defined goal state. The aim of this mechanism is to find the optimal way to reach this goal state. This way is given by a sequence of actions, each usually performed after a decision at a given, discrete point in time. Reinforcement learning methods are mostly applied to operate on a special case of sequential decision problems, so-called *Markov decision processes* (MDPs).

Reinforcement learning is very valuable when the characteristics of the underlying system are not known and/or difficult to describe or when the environment of an acting agent is only partially known or completely unknown. Various applications have been realized with reinforcement learning approaches, mostly concerning game playing, robotics, and control problems.

1.1.1 An Agent Control Task

Autonomous agents are in continuous interaction with the world they are operating in. Navigation in space, which is an essential ability of such agents, is a complicated process of perceiving the environment with their sensory system and performing physical actions according to an adequate interpretation of the collected sensory data. What is adequate in this context depends on the problem the agent has to solve.

Example 1.1. Imagine a discrete grid world with 6×6 grid cells (Fig. 1.1). An agent is always within one of the grid cells and can go from there to the adjacent grid cells in cardinal directions. The world is unknown to the agent, and its task is to reach a specified goal location from any position within the grid. There are 36 different positions the robot can be in, the *system states* or, for short, the *states*. This problem

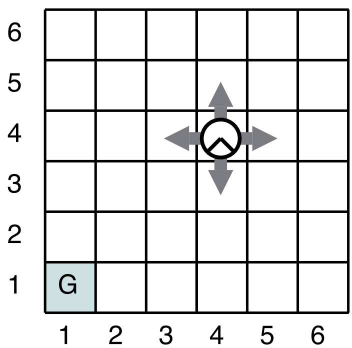

Fig. 1.1 A grid cell example. The robot is at position $(4,4)$. Its goal (G) is to reach the bottom left cell $(1,1)$. Its primitive actions are movements to neighbored cells to the left, right, top, and bottom of its position

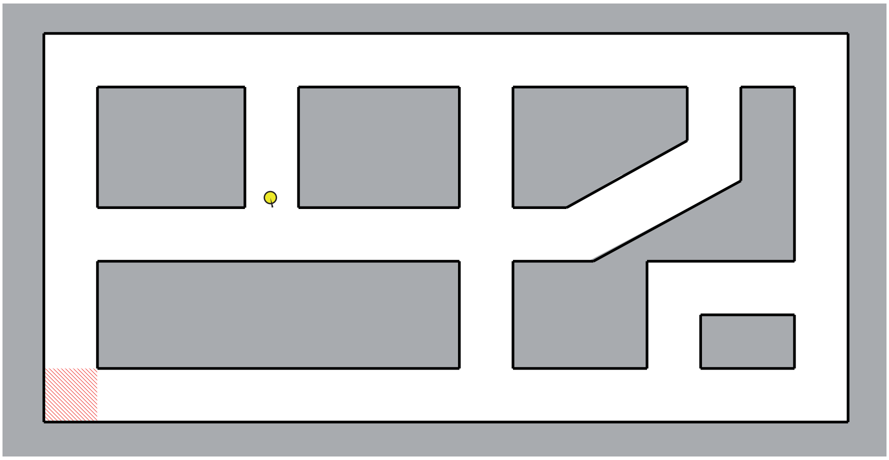

Fig. 1.2: A robot in a simulated office environment: the state space of this problem is continuous

can be formulated as a Markov decision process and can therefore be solved with reinforcement learning.

During the training process, the agent learns a *policy* that returns a particular action to execute for every state the agent is in. The policy is based on the *value function* that maintains an assessment of states with regard to solving the overall problem. For the simple problem in Example 1.1, a reinforcement learning algorithm is able to learn a solution after executing a few hundred actions. The complexity of RL scales linearly with the number of states: To give an impression, Kaelbling et al. (1996) report the need for 531,000 learning steps for a grid world with 3,277 states.

Long training times are a general problem of reinforcement learning. RL methods are proved to converge to an optimal solution, but the prerequisite is that *each* system state be continuously updated—which is practically impossible in larger state spaces. Even worse, most real-world state spaces are not discrete. Figure 1.2 shows a robot in an office environment—a continuous world with an infinite number of states.

An important question that arises here is how to describe a system state in a given context. Sensor readings of the robot are given in real numbers and form a continuous state space. That means that the value function has a continuous domain and cannot be stored easily in a table as in the case of a discrete one. To cope with continuous state spaces, some kind of *value function approximation* is used. Various approaches exist for this. What is common to all of them is that the incorporation of these methods introduces, besides a bunch of new parameters to cope with, uncertainty in the representation that may have unwanted effects. If the approximation is too rough, states may not be distinguished even if they had to be; if it is too fine, the training times will become unacceptably long. The choice of the right function approximation and the choice of its parameters usually requires solid expert knowl-