ulm university universität
uulm

Fakultät für Mathematik und Wirtschafts-wissenschaften
Institut für numerische Mathematik

C++: Solving Markov Decision Processes

Prüfungsleistung für die Universität Ulm

Vorgelegt von:
Adrian Lauber
adrian.lauber@uni-ulm.de
1042606

Gutachter:
Prof. Dr. Andreas Borchert

2020

Fassung 10. August 2020

Satz: PDF-LaTeX 2ε

Inhaltsverzeichnis

1 Introduction

The goal of this project is to implement a library to solve *Markov decision processes (MDPs)*. The focus is on extensibility, facilitating modern C++ features like smart pointers as well as following well-established C++ guidelines and best-practices.

The first chapter is a brief introduction into the problem class for which the library functionality is developed.

In the second chapter the focus is on the actual implementation. The class hierarchy is described as aspects of the build and test environment.

2 Markov Decision Process

Markov Decision Processes describe a time-discrete and stochastic process that can be used to model different decision-making problems. MDPs are used in all kinds of domains and multiple algorithms have been proposed to solve a particular MDP.

A MDP is defined by a state-transition-function $P_a(s, s')$ that gives the probability of transferring to a consecutive state s' when in state s and applying action a. For a given state s and action a the probability is independent from former states or actions. This property is referred to as the *Markov property*. The goal of optimization is a scalar value referred to as the *cost* or *reward* which is either minimized (in case a problem defined by costs) or maximized (for rewards). This optimization metric is a function of the current state and consecutive state given an action: $R_a(s, s')$

3 Implementation

3.1 Classes

3.2 Build

3.3 Testing

4 Extensibility