

## Übungen zum Modul WPM6: Algorithmisches Lernen

### SS 2011 Blatt 5

Ausgabe: 13.07.2011, Besprechung: 13.07.2011

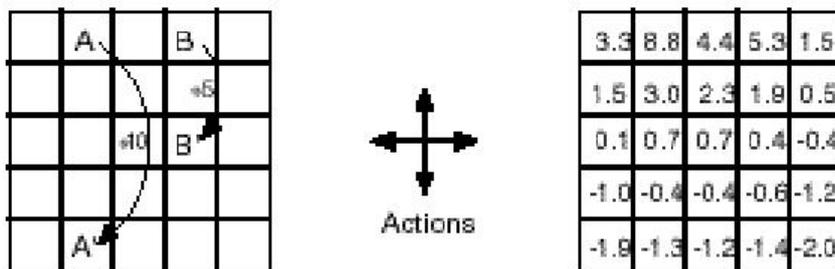
#### Folgende Aufgaben sind freiwillig zur Übung gedacht

**Aufgabe 5.1 [1 Punkte] Grid World Programmierung:** Implementieren Sie mit einem Werkzeug Ihrer Wahl, **eine** der folgenden Aufgaben mit Hilfe der DP Algorithmen.  $\gamma$  soll frei wählbar sein. Alle Zustände werden mit 0 initialisiert. Implementieren sie zunächst die Policy-Evaluation für die Zufalls-Policy. Definieren sie ein  $\Delta$ , bei dessen Unterschreitung die Evaluierung terminiert. Führen Sie eine Policy-Verbesserung durch und erweitern Sie Ihr Programm um eine Policy Iteration.

Empfohlene Vorgehensweise für alle, die einen Einstieg suchen:

- Laden sie die Datei gridworld.c von der Seite der Übungen herunter
- Kompilieraufwurf:  
gcc gridworld2.c -o gridworld2
- dieses Programm führt eine Policy Evaluation für das Beispiel Gridworld II durch

#### 5.1.1 [1]: Das Grid World Beispiel I aus der Vorlesung. (RL1 - S. 58)



- Eine kontinuierliche Aufgabe
- Aktionen, die den Agenten aus dem *Grid* nehmen würden, lassen den Zustand unverändert und geben einen Reward von -1 (Ausnahme: die beiden Sprünge, die stets mit 5 und 10 belohnt werden)
- Der *Reward* ist 0, außer wenn der Sprung A oder B ausgeführt wird

#### 5.1.2 [1]: Das Grid World Beispiel II aus der Vorlesung. (RL2 - S. 9)



- Eine episodische Aufgabe
- Nicht-terminale Zustände :1, 2, ..., 14;
- Ein terminaler Zustand (zweimal als schattiertes Quadrat dargestellt)
- Aktionen, die den Agenten aus dem *Grid* nehmen würden, lassen den Zustand unverändert



- Der *Reward* ist -1 bis der terminale Zustand erreicht ist

**5.1.3 [1]:** Ein Labyrinth Ihrer Wahl. Zeichnen Sie ein  $3 \times 3$ ,  $4 \times 4$  oder  $n \times m$  Labyrinth mit Wänden, Sackgassen etc.

- Eine episodische Aufgabe
- Eine Start- und ein Zielzustand
- Gegen Wände laufen läßt den Zustand unverändert.
- Der *Reward* ist -1 bis das Labyrinth gelöst ist