

Übung 2

Aufgabe 2.1

Definieren Sie mit eigenen Worten die Begriffe: Agent, Agentenfunktion, Agentenprogramm, Rationalität, Autonomie, Reflex-Agent, modellbasierter Agent, zielbasierter Agent, nutzenbasierter Agent, lernender Agent.

Aufgabe 2.2

Sowohl die Leistungsbewertung als auch die Nutzenfunktion messen, wie gut sich ein Agent verhält. Erklären Sie den Unterschied zwischen beiden.

Aufgabe 2.3

Betrachten Sie die Unterschiede zwischen Agentenfunktionen und Agentenprogrammen.

- Kann es mehrere Agentenprogramme geben, die eine bestimmte Agentenfunktion implementieren? Nennen Sie ein Beispiel oder zeigen Sie, warum es nicht möglich ist.
- Gibt es Agentenfunktionen, die von keinem Agentenprogramm implementiert werden können?
- Implementiert jedes Agentenprogramm für eine vorgegebene feste Maschinenarchitektur genau eine Agentenfunktion?
- Wie viele verschiedene mögliche Agentenprogramme gibt es für eine bestimmte Architektur mit n Bit Speicher?

Aufgabe 2.4

Betrachten Sie die Rationalität verschiedener Staubsauger-Agentenfunktionen.

- Zeigen Sie, dass die folgende einfache Staubsauger-Agentenfunktion (Tabelle) unter den Voraussetzungen für Rationalität tatsächlich rational ist.

Wahrnehmungsfolge	Aktion
[A, sauber]	rechts
[A, schmutzig]	saugen
[B, sauber]	links
[B, schmutzig]	saugen
[A, sauber], [A, sauber]	rechts
[A, sauber], [A, schmutzig]	saugen
...	...
[A, sauber], [A, sauber], [A, sauber]	rechts
[A, sauber], [A, sauber], [A, schmutzig]	saugen
...	...

- Leistungsbewertung: einen Punkt für jedes saubere Quadrat in jedem Zeitabschnitt über eine Lebensdauer von 1000 Zeitschritten
- Die „Geographie“ der Umgebung ist *a priori* bekannt, aber die Schmutzverteilung und die Ausgangsposition des Agenten nicht. Saubere Quadrate bleiben sauber, und durch das Saugen wird das aktuelle Quadrat gereinigt. Die Aktionen *Links* und *Rechts* bewegen den Agent nach links und rechts, außer wenn der Agent dadurch aus der Umgebung heraus geraten würde; in diesem Fall bleibt er dort, wo er ist.
- Die einzigen verfügbaren Funktionen sind *links*, *rechts*, *Saugen* und *NoOp* (nichts machen)
- Der Agent nimmt seine Position korrekt wahr und erkennt, ob sie schmutzig ist.

- Beschreiben Sie eine rationale Agentenfunktion für eine veränderte Leistungsbewertung, die für jede Bewegung einen Punkt abzieht. Braucht das zugehörige Agentenprogramm einen internen Zustand?
- Diskutieren Sie mögliche Agentenentwürfe für Fälle, wo saubere Quadrate wieder schmutzig werden können und die Geographie der Umgebung unbekannt ist. Ist es in

diesen Fällen sinnvoll, dass der Agent aus seiner Erfahrung lernt? Wenn dies der Fall ist, was soll er lernen?

Aufgabe 2.5

Entwickeln Sie für jeden der folgenden Agenten eine Beschreibung der Arbeitsumgebung:

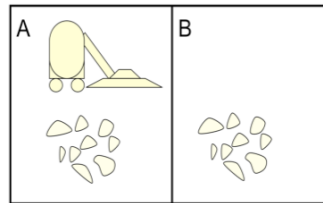
- a. Roboter-Fußballspieler
- b. Internet-Bucheinkaufsagent
- c. Autonomes Marsfahrzeug
- d. Beweisassistent für einen Mathematiker

Aufgabe 2.6

Charakterisieren Sie für jeden der in Übung 2.5 entworfenen Agententypen die Umgebung gemäß der in Kapitel 2.3 beschriebenen Eigenschaften und wählen Sie einen geeigneten Agentenentwurf aus.

Aufgabe 2.7 (Programmieraufgabe)

Implementieren Sie einen Leistungsbewertungs-Umgebungssimulator für die gezeigte Staubsaugerwelt.



Ihre Implementierung sollte modular sein, damit die Eigenschaften von Sensoren, Aktuatoren und Umgebung (Größe, Form, Schmutzplatzierung usw.) einfach geändert werden können. Die Java-Klassenbibliothek mit allen benötigten Klassen finden Sie unter <http://code.google.com/p/aima-java/downloads/list>; der Link steht auf meiner KI-Webseite unter „Links“.

Aufgabe 2.8 (Programmieraufgabe)

Implementieren Sie einen einfachen Reflex-Agenten für die Staubsaugerumgebung aus Aufgabe 2.7. Führen Sie den Umgebungssimulator mit diesem Agenten für alle möglichen anfänglichen Schmutzkonfigurationen und Agentenpositionen aus. Zeigen Sie die Leistungsbewertung des Agenten für jede Konfiguration auf ebenso wie den gesamten Durchschnittspunktebetrag.