

# Grundlagen der Künstlichen Intelligenz

Prof. Dr. J. Boedecker, Prof. Dr. W. Burgard, Prof. Dr. F. Hutter, Prof. Dr. B. Nebel,  
Dr. rer. nat. M. Tangermann  
M. Krawez, T. Schulte  
Sommersemester 2019

Universität Freiburg  
Institut für Informatik

## Übungsblatt 1 — Lösungen

### **Aufgabe 1.1** (Möglichkeiten und Grenzen der KI)

Recherchieren Sie in der KI-Literatur bzw. im Internet, inwiefern folgende Probleme heutzutage mittels Computer- bzw. Robotereinsatz gelöst werden können:

- (a) Spielen der Brettspiele Dame und Go.

**Lösung:**

Checkers is completely solved, optimal game guarantees a tie.

The game of Go has been regarded as a benchmark for AI for a long time. Since the number of possible games is much larger than, e.g., in chess, it often was claimed that the game requires human-like creativity. In 2016, Google's AlphaGo algorithm has beaten Lee Sedol, a world top Go player.

[http://en.wikipedia.org/wiki/Computer\\_Go](http://en.wikipedia.org/wiki/Computer_Go).

- (b) Verarbeiten natürlicher Sprache in Echtzeit.

**Lösung:**

Natural language processing is still an open field of research. A major milestone recently was IBM's Watson computer, which was able to win the first place in the popular quiz show Jeopardy! after competing against former human winners. This involved understanding questions posed in natural language and searching for the answer through a large database of information.

<http://www.research.ibm.com/deepqa/deepqa.shtml>

- (c) Autonomie unbemannter Fahr- und Flugzeuge (UGVs und UAVs).

**Lösung:**

In the context of the „DARPA Grand Challenge“ in 2005 the team of Sebastian Thrun from the Stanford University reached the goal of autonomous navigation of a vehicle through the desert at a route of 200 km.

By 2016, Google's Waymo cars have driven autonomously for over 2.7 millions kilometers on public roads, with only few accidents caused by the self-driving car.

More information:

[http://en.wikipedia.org/wiki/DARPA\\_Grand\\_Challenge](http://en.wikipedia.org/wiki/DARPA_Grand_Challenge) and  
[http://en.wikipedia.org/wiki/Google\\_driverless\\_car](http://en.wikipedia.org/wiki/Google_driverless_car).

Autonomous aircrafts are comparably easy to construct:

<http://www.ida.liu.se/ext/witas/> or  
[http://en.wikipedia.org/wiki/Unmanned\\_aerial\\_vehicle](http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle).

- (d) Automatische Gesichtserkennung.

**Lösung:**

In a crowd the rate of correctly recognized people is around 80 percent. In well-controlled settings this rate can be high enough to be used for, e.g., access control in banks, in military or scientific facilities. In these use cases the person have to face the camera frontally and don't move for some time.

- (e) Spielen von Computerspielen (z.B. klassische Atari-Spiele) wie ein Mensch.

**Lösung:**

The performance is of course dependent on the game at hand, but a big number of games can be approached by deep learning:

<http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html>

- (f) Komponieren von Musik.

**Lösung:**

Several approaches to automatic music generation exist, two most prominent algorithms are:

EMI

In a training phase the algorithm analyses several music pieces, then it tries to produce a new piece following the style of the input pieces.

<http://www.computerhistory.org/atc/m/algorithmic-music-david-cope-and-emi>

Melomics

Instead of copying a given style, Melomics generates a new music piece from some seed value, following general rules of music composition.

<http://geb.uma.es/melomics>

- (g) Turing-Test

**Lösung:**

Chat bots (programs which simulate a conversation by answering to user input) have a long history and have grown quite sophisticated in recent years. They typically use a large online data base, which is extended by analyzing chats with users. However, those bots are still far away from passing the Turing test, mainly because they cannot well understand the semantics and context of a conversation.

Schreiben Sie Ihre Erkenntnisse in jeweils 2–3 Sätzen auf.

**Aufgabe 1.2** (Performanz und Nutzen)

- (a) Was ist der Unterschied zwischen einer Performanzmessung und einer Nutzenfunktion?

**Lösung:**

Ein Performanzmessung bewertet die Leistung eines Agenten von außen, quasi aus der Sicht einer objektiven externen Instanz, während eine Nutzenfunktion dem Agenten selbst erlaubt, seine möglichen Aktionen zu bewerten.

- (b) Beschreiben Sie den Zusammenhang zwischen Performanzmessung und Nutzenfunktion bei einem lernenden Agenten.

**Lösung:**

Ein lernender Agent verändert seine utility function auf Grund der Rückmeldung des „Kritikers“, die wiederum von der Performanzmessung abhängt.

**Aufgabe 1.3** (Rationale Agenten)

- (a) Geben Sie für jeden der folgenden Agenten eine Beschreibung des Performanzmaßes, der Umgebung, der Aktuatoren und der Sensoren (**P**erformance **E**nvironment **A**ctuators **S**ensors) an:

- (i) Monopoly spielen
- (ii) Leichtathlet beim Weitsprung
- (iii) 2048 spielen (<http://gabrielecirulli.github.io/2048>)

**Lösung:**

Agent	P	E	A	S
Monopoly	Besitz Spielende	nach Spielbrett, an- dere Spieler	Würfeln, Grundstücke kaufen, ...	Würfelaugen, Handlungen anderer Spieler, ...
Weitsprung	gesprungene Weite in Meter	Sportstadion	Muskeln des Athleten	Augen des Athleten, Schiedsrich- terentscheidung (übertreten, Weite)
2048	höchstes Highscore	Tile, 4 x 4 grid	4 Bewegungen	Zustand des Spiels

- (b) Klassifizieren Sie die in (a) formalisierten Umgebungen der Agenten nach folgenden Kriterien:

- vollständig beobachtbar oder teilweise beobachtbar

- deterministisch oder stochastisch
- statisch oder dynamisch
- diskret oder kontinuierlich

**Lösung:**

- Monopoly: teilweise beobachtbar, stochastisch, statisch, diskret
- Weitsprung: vollständig beobachtbar, stochastisch, statisch (abgesehen vom Wind), kontinuierlich
- 2048: vollständig beobachtbar, stochastisch, statisch, diskret

**Aufgabe 1.4** (Problemformalisierung)

Geben Sie für die folgenden Problemstellungen jeweils eine möglichst präzise Formulierung an, die aus Anfangszustand, Zustandsraum, Aktionen, Zieltest und einer Pfadkostenfunktion besteht:

- Sie wollen den Rubiks Zauberwürfel lösen.  
<http://de.wikipedia.org/wiki/Zauberw%C3%BCrfel>

**Lösung:**

**Zustände:** Die Zustände sind  $9 \cdot 6 = 54$ -Tupel der Farbe der Seitenflächen.

**Anfangszustand:** “zufällige” Konfiguration der Farben

**Zieltest:** Haben alle Seitenflächen eine einheitliche Farbe?

**Aktionen:** Drehen einer der Seitenflächen um  $90^\circ$  (Oben, unten, links, rechts, vorne, hinten) eventuell auch Drehungen um  $180^\circ$  sowie der Mittelfläche

**Pfadkosten:** Je nach Metrik und verfügbaren Aktionen 1 pro Aktion (oder 2 pro  $180^\circ$  Drehung)

- Sie wollen eine Karte von Europa mit nur vier Farben einfärben. Damit man die Grenzen einzelner Länder erkennen kann, ist es erforderlich, dass es keine Nachbarländer mit gleicher Einfärbung gibt.

**Lösung:**

**Zustände:** Die Zustände sind 49-Tupel der Farben *rot, grün, blau, gelb* oder *ungefärbt* (z.B.). Jeder Eintrag repräsentiert ein Land auf der Karte. Der Kompaktheit halber kann man die Farben mit den Zahlen 0 (*rot*) bis 4 (*ungefärbt*) kodieren.

**Anfangszustand:** Alle Länder sind ungefärbt / alle Länder sind rot / jedes Land kann eine der  $4+1=5$  Farben [ungefärbt, r,g,b,y] haben.

**Zieltest:** Alle Länder sind gefärbt, benachbarte Länder haben unterschiedliche Farben?

**Aktionen:** Einem Land eine der 4 Farben zuweisen und vorher überprüfen ob ein Nachbarland schon die gleiche Farbe hat / ein Land umfärben, wenn ein Konflikt festgestellt wird.

**Pfadkosten:** Jede Aktion verursacht Einheitskosten.