1 Einleitung

Es gibt viele unterschiedliche Tätigkeiten, die wir als *Lernen* bezeichnen können, wie z.B. eine Fremdsprache erlernen, Schach spielen lernen, oder eine fremden Stadt kennenlernen. Die algorithmische Erlernbarkeitstheorie geht der Frage nach, ob und wie solche Lernprobleme durch Maschinen bzw. Algorithmen gelöst werden können. Hierzu befaßt sie sich u.a. mit

- der Entwicklung von mathematischen Lernmodellen für unterschiedliche Lernscenarien,
- dem Entwurf und Analyse von Algorithmen für unterschiedliche Lernprobleme, und
- dem Nachweis der Nicht-Erlernbarkeit von Lernproblemen für ein festgelegtes Modell

Die Spezifikation eines Lernmodells erfordert die Festlegung der folgenden Punkte:

1. *Was soll gelernt werden?* (Sprachen, Funktionen, Gewinnstrategien)
2. *Wie soll gelernt werden?* (Interaktion mit der Umwelt, Lehrer)
3. *Wodurch ist erfolgreiches Lernen charakterisiert?*

Zur Erläuterung dieser Punkte betrachten wir zunächst einmal die folgende Aufgabestellung.

Beispiel 1.1. Ein Roboter bewegt sich in einer unbekannten Umgebung. Dabei sind einige Orte verboten, alle anderen Orte sind erlaubt. In jedem Schritt kann sich der Roboter von einem Ort zu einem benachbarten Ort bewegen, indem er eine aus einer kleinen Anzahl von verschiedene Grundbewegungen ausführt (z.B. nach vorne gehen, nach rechts gehen, nach links gehen und zurückgehen). Ziel des Roboters ist es, sich ein vollständiges Bild seiner Umgebung zu machen.

Zunächst überlegen wir uns, wie wir eine solche Umgebung mathematisch modellieren können. Eine Möglichkeit besteht darin, die Umgebung durch einen *endlichen Automaten* zu repräsentieren. Die verschiedenen Orte entsprechen dann den Zuständen des Automaten, und die erlaubten Orte sind akzeptierenden Zustände. Mögliche Bewegungen zwischen den einzelnen Orten werden durch die Überführungsfunktion des Automaten dargestellt. Das Alphabet des Automaten ist also die Menge $\{V, R, L, Z\}$, und ein akzeptiertes Wort entspricht einer Wanderung des Roboters von einem festen Ausgangspunkt zu einem erlaubten Ort. (Eine solche Modellierung läßt eine sehr kompakte Beschreibungen der Umgebung zu. Zum Beispiel kann eine Umgebung, in der all Orte erreichbar und erlaubt sind, durch einen Automaten mit nur einem akzeptierenden Zustand beschrieben werden.)

Der Roboter soll in beliebigen Umgebungen eingesetzt werden können. Die Aufgabe ist also, den endlichen Automaten für die zunächst unbekannte Umgebung aus der Menge *aller* endlichen Automaten zu finden. Dabei können wir uns allerdings auf solche Automaten mit Alphabet $\{V, R, L, Z\}$ beschränken. Damit haben wir festgelegt, was gelernt werden soll, und das Lernziel ist die vollständige Kenntnis des zu erlernenden Automaten. Das weitere Vorgehen zur algorithmische Lösung dieser Aufgabe hängt nun entscheidend davon ab, in welcher Art wir Informationen über den in Frage stehenden Automaten erhalten. Einen Lernalgorithmus für endliche Automaten werden wir ausführlich in Abschnitt 6 behandeln.

Betrachten wir nun ein Beispiel für das Erlernen eines einfachen geometrischen Konzepts.

Beispiel 1.2. Es sollen achsenparallelen Rechtecken in $\mathbb{N}^2$, also Mengen der Form

$$\{(x, y) \in \mathbb{N}^2 \mid x_{min} \leq x \leq x_{max} \wedge y_{min} \leq y \leq y_{max}\}$$

erlernt werden. Der Lernalgorithmus erhält eine Folge von Punkten $p_1, p_2, \ldots$ mit $p_i \in \mathbb{N}^2$. Für jeden Punkt p_i muß er eine Vorhersage machen, ob p_i zu dem Rechteck R gehört, bevor er erfährt, ob p_i im Rechteck ist oder nicht. Der Lernalgorithmus soll möglichst wenige Vorhersagefehler machen.

Die Aufgabe ist also, ein zunächst unbekanntes Rechteck aus der Menge aller Rechtecke zu finden. Solche Rechtecke können durch die Koordinaten zweier

Eckpunkte beschrieben werden. Der Lernalgorithmus erhält Informationen über das in Frage stehende Rechtecke in Form von positiven und negativen *Beispielen*, also Punkte die in oder außerhalb des zu erlernenden Rechtecks liegen. Er hat aber keinen Einfluß auf die Auswahl der Punkte.

Betrachten wir nun den folgenden Lernalgorithmus. Zu Beginn des Lernvorgangs nehmen wir an, daß kein Punkt in dem Rechteck enthalten ist. Wir wählen also als Starthypothese $h = \emptyset$. Für jeden Punkt p_i treffen wir dann die Vorhersage gemäß unserer aktuellen Hypothese h. Im Fall einer falschen Vorhersage ersetzen wir h durch das kleinste Rechteck das h und p_i enthält. Die Hypothese h bleibt stets innerhalb des in Frage stehenden Rechtecks, und bei jeden Vorhersagefehler erhöhen wir die Höhe oder die Breite von h. Es folgt, daß wir höchstens $h + b - 1$ Vorhersagefehler für beliebige Folgen $p_1, p_2, \ldots$ und Rechtecke der Höhe $h = y_{max} - y_{min} + 1$ und Breite $b = x_{max} - x_{min} + 1$ machen.