



# 2

## Algorithmen

### Über die Kunst, Computer zu Problemlösern zu machen

Nicolas Berberich

Zuallererst möchten wir einen zentralen Begriff der Informatik klären, der Ihnen in diesem Buch immer wieder begegnen wird, und zwar den des *Algorithmus*. Wir könnten das kurz und schmerzlos machen, indem wir Ihnen einfach eine Definition präsentieren.

#### Definition Algorithmus

Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems. Eine Eingabe wird dabei in genau definierten Schritten zu einer Ausgabe umgewandelt.

---

N. Berberich (✉)

TU München und LMU München, München, Deutschland

E-Mail: [n.berberich@tum.de](mailto:n.berberich@tum.de)

Zum besseren Verständnis sind jedoch konkrete Beispiele hilfreich und deutlich interessanter als reine Definitionen. Zum Glück hat unsere Lisa ein abenteuerliches Leben, in dem es nur so von Algorithmen wimmelt. So wie an diesem Sonntag. Endlich ist es soweit: Lisas Brieffreundin Jana aus Frankreich kommt sie besuchen. Lisa hat Jana versprochen, mit ihr deutsche Pfannkuchen zu backen. Wenn Lisa ihre Freundin das nächste Mal in Frankreich besucht, wird Jana sie dafür in die hohe Kunst der Crêpe-Zubereitung einführen. Ärgerlicherweise kann sich Lisa aber nicht mehr erinnern, wie genau Pfannkuchen zubereitet werden, und muss deshalb auf ein Backrezept zurückgreifen. Im Grunde ist ein Backrezept ein Algorithmus, denn es liefert eine genaue Handlungsvorschrift, nach der die Zutaten (die Eingabe des Algorithmus) in mehrere Pfannkuchen (die Ausgabe des Algorithmus) umgewandelt werden. Hier ist das Backrezept für das sich Lisa entscheidet:

Backrezept für Pfannkuchen:

*Eingabe* (Zutaten): 400 g Mehl, 750 ml Milch, 3 Eier, 1 Prise Salz

*Handlungsvorschrift*:

1. Mehl und Milch in einer Schüssel verrühren.
2. Eier und eine Prise Salz hinzugeben und verrühren.
3. Pfanne erhitzen.
4. Mit einer Schöpfkelle Teig in die Pfanne geben und auf dem Pfannenboden verteilen.
5. Pfannkuchen bei mittlerer Hitze backen.
6. Wenn die Unterseite goldbraun ist, dann wende den Pfannkuchen mit einem Pfannenwender.

7. Wenn die zweite Seite goldbraun ist, dann nimm den Pfannkuchen mit dem Pfannenwender heraus.
8. Solange noch Teig übrig ist: Gehe zu Schritt 4.
9. Fertig! Guten Appetit!

*Ausgabe:* 12 Pfannkuchen

Theoretisch könnte man einen solchen Algorithmus auch einem fortgeschrittenen Roboter übergeben und diesen die Handlungsschritte durchführen lassen. Wichtig ist, dass ein Algorithmus, der von einem Roboter oder einem Computer durchgeführt werden soll, sehr *präzise formuliert ist und keine Mehrdeutigkeiten zulässt*. Im Gegensatz zu einem Menschen verfügt ein Roboter nämlich nicht über einen *gesunden Menschenverstand* und weiß deshalb nicht, was „eine Prise“ Salz bedeutet. Hier müsste stattdessen eine genaue Mengenangabe stehen. Genauso müsste auch exakt vorgeschrieben werden, was mit „mittlerer Hitze“ und was genau mit „goldbraun“ gemeint ist.

Möchte man einen Algorithmus von einem Computer durchführen lassen, dann müssen Eingabe und Ausgabe natürlich digital sein – zum Beispiel Zahlen, Wörter oder digitale Dokumente. Ein einfaches Beispiel ist ein Algorithmus, der zwei Zahlen durch wiederholte Addition miteinander multipliziert.<sup>1</sup>

---

<sup>1</sup>Das Prinzip, eine Multiplikation durch wiederholte Addition in einer Rechenmaschine darzustellen, geht auf den Universalgelehrten Gottfried Wilhelm Leibniz zurück. Schon im Jahr 1673 – mehrere Jahrhunderte vor der Entwicklung von elektronischen Computern – stellte Leibniz in London der Royal Society das Modell einer mechanischen Rechenmaschine vor, welche alle vier Grundrechenarten beherrschte. Leibniz führte übrigens auch das Binärsystem in der europäischen Wissenschaft ein, nach dem alle Zahlen durch 0en und 1en dargestellt werden können und welches die Basis für alle digitalen Computer bildet.

Dieser kann so aussehen (zum besseren Verständnis stehen die Beispielzahlen in eckigen Klammern):

*Eingabe:* zwei positive ganze Zahlen, die miteinander multipliziert werden sollen [z. B. 3 und 4]:

*Handlungsvorschrift:*

1. Setze die Variable  $x$  zu Beginn auf 0.
2. Zähle von 0 in Einserschritten bis zur ersten Eingabezahl [bis 3].
3. Bei jedem dieser Einserschritte addiere die zweite Zahl [4] auf  $x$  und speichere das *Ergebnis* erneut in  $x$  ab.
4. Nachdem du in Einserschritten bei der ersten Zahl [3] angekommen bist und das Ergebnis aktualisiert hast [12], bist du fertig.

*Ausgabe:* Das Ergebnis ist das Produkt der beiden Eingabezahlen.

Auch wenn sich diese Vorschrift vielleicht etwas kompliziert gelesen hat, entspricht sie doch genau der Art und Weise, wie wir alle in der Grundschule Multiplizieren gelernt haben: Man erhält das Ergebnis von „3 mal 4“, wenn man 4 drei mal addiert: „ $3 * 4 = 4 + 4 + 4$ “.

Diese logische Struktur des Algorithmus kann mithilfe einer Programmiersprache in Computercode aufgeschrieben werden. Effektiv funktioniert ein Algorithmus also wie eine Maschine, in die wir Dinge oben hineinwerfen (Eingabe) und aus der nach mehreren Verarbeitungsschritten unten ein Ergebnis herauskommt (Ausgabe). Was genau im Inneren des Algorithmus vor sich geht, ist häufig für Anwender gar nicht so wichtig. Deshalb werden Algorithmen manchmal als schwarze Boxen (engl. black boxes) betrachtet, in die

man nicht hineinsehen kann und von denen man nur die Eingabe und Ausgabe kennt.

Dieses *Blackbox-Denken* kann aber auch gefährlich sein, weil man dadurch leicht übersieht, in welchen Fällen man den Algorithmus nicht einsetzen kann. Der obige Multiplikationsalgorithmus funktioniert beispielsweise nur für positive ganze Zahlen und nicht für die Multiplikation von zwei negativen Zahlen oder zwei Kommazahlen. Wann genau (also für welche Eingabewerte) ein Algorithmus eingesetzt werden kann und für welche Eingabewerte er falsche Ergebnisse liefert, kann man nur herausfinden, indem man die Blackbox öffnet und versucht, die Funktionsweise des Algorithmus nachzuvollziehen. Es wird in diesem Buch darum gehen, die Blackboxes der wichtigsten Algorithmen des maschinellen Lernens und der künstlichen Intelligenz zu öffnen.

Die Eingabe und Ausgabe eines Algorithmus können auch Positionen und Richtungen in der realen Welt sein (z. B. die eigene Position in einem Labyrinth und der Weg nach draußen). Das erkennen Lisa und Jana, als sie nach dem Verzehr ihrer Pfannkuchen zu einem Maislabyrinth fahren:

„Wie sollen wir hier jemals wieder herausfinden?“ Leicht genervt schaut sich Jana um. „Das sieht in allen Richtungen gleich aus. Überall diese blöden Pflanzen!“

Lisa fühlt sich etwas schuldig, schließlich hat sie ihre Freundin zum Besuch des Maislabyrinths überredet. Es dauert nicht lange, bis sich die beiden heillos verlaufen haben. Bei jeder Kreuzung eine zufällige Richtung zu wählen, war kein guter Plan gewesen, um aus dem Labyrinth zu kommen...

„Wir brauchen eine richtige Strategie. Eine Vorschrift, nach der wir uns bei jeder Kreuzung für eine Richtung entscheiden können und damit nach draußen finden. Lass uns mal etwas nachdenken, bevor wir weiter planlos umherirren“, sagt Lisa. Obwohl Jana wenig überzeugt aussieht, nickt sie und setzt sich zum Nachdenken neben Lisa.

### Seneca über Irrgärten

„Das eben geschieht den Menschen, die in einem Irrgarten hastig werden: Eben die Eile führt immer tiefer in die Irre.“

Lucius Annaeus Seneca (4 n. Chr.–65 n. Chr.)

Nach einigen Minuten hat sich Lisa eine Strategie überlegt. Sie möchte einfach immer an der rechten Wand entlang laufen und an jeder Kreuzung den Weg nehmen, der direkt rechts von ihrer Laufrichtung liegt.

Diese Strategie heißt Tiefensuche und lässt sich wie folgt in Algorithmenform schreiben:

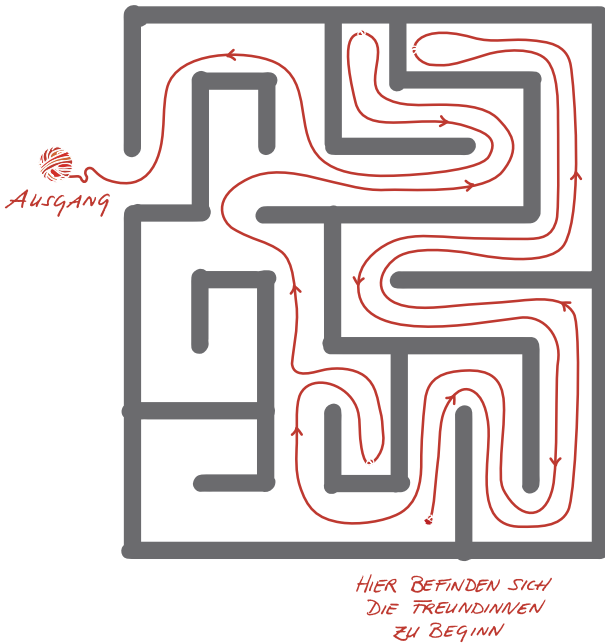
*Eingabe:* Lisa und Jana im Inneren des Labyrinths.

*Handlungsvorschrift:*

1. Wähle eine beliebige Richtung parallel zur Labyrinthwand.
2. Gehe an der rechten Labyrinthwand entlang, bis du am Ausgang des Labyrinths angekommen bist.

*Ausgabe:* Weg zum Ausgang des Labyrinths

Obwohl Jana der Zweifel an dieser Strategie ins Gesicht geschrieben steht, folgt sie ihrer Freundin Lisa. Und tatsächlich! Schon nach kurzer Zeit haben die beiden den Ausgang des Labyrinths gefunden. „War ja gar nicht so schwer, wie gedacht“, grinst Lisa zufrieden. In Abb. [2.1](#)

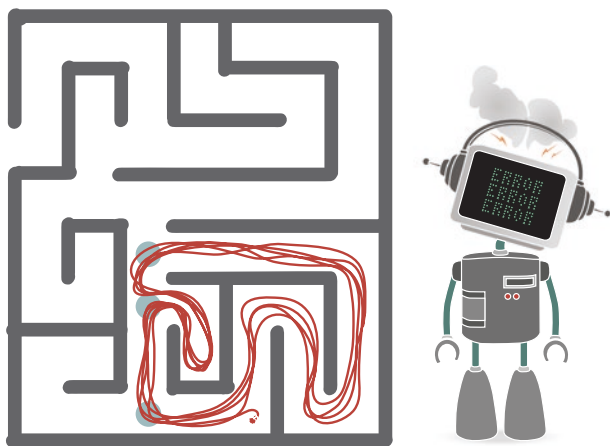


**Abb. 2.1** Der Weg aus dem Labyrinth mithilfe des Tiefensuche-Algorithmus

ist der Weg eingezeichnet, den die beiden zurückgelegt haben.

Algorithmen werden meist in einer allgemeinen Form und damit unabhängig vom konkreten Problemfeld definiert. Das hat den Vorteil, dass sie damit auf eine große Anzahl verschiedener Problemfelder angewendet werden können. Die allgemeine Form des Algorithmus, den Lisa und Jana verwendet haben, um aus dem Labyrinth herauszufinden, wird Tiefensuche-Algorithmus genannt. Wie der Name schon andeutet, gehört der Tiefensuche-Algorithmus zur Gruppe der Suchalgorithmen. Neben dem Suchen eines Weges durch

ein Labyrinth gehören Routenplaner in Navigationsgeräten und Suchmaschinen wie diejenige von Google zu den praktischen Anwendungen von Suchalgorithmen. Wie oben bereits erwähnt, ist es wichtig zu wissen, welche Algorithmen für eine bestimmte Aufgabe verwendet werden können und insbesondere, in welchen Situationen bestimmte Algorithmen *nicht* angewendet werden können. Der Tiefensuche-Algorithmus führt nach einiger Zeit immer zum Ziel, es sei denn, im Labyrinth gibt es *Zyklen*. Ein Zyklus ist ein Rundweg, bei dem man immer wieder und ohne Ende an bereits besuchte Kreuzungen kommt. Ein Mensch würde nach einigen Runden im Kreis stutzig werden, seinen gesunden Menschenverstand einschalten und die Strategie abwandeln. Ein Computer oder Roboter weicht hingegen von seinem Algorithmus nicht ab und würde bis in alle Ewigkeit seine Runden drehen (siehe Abb. 2.2).



**Abb. 2.2** Der Tiefensuche-Algorithmus kann auf manche Probleme angewendet zu Endlosschleifen führen. Deshalb muss auf Zyklen geprüft werden. Man sollte im Allgemeinen Algorithmen nicht blind anwenden, sondern deren Begrenzungen beachten



Albert Einstein hat angeblich einmal gesagt: „*Die Definition von Wahnsinn ist, immer wieder das Gleiche zu tun und andere Ergebnisse zu erwarten.*“ Für intelligentes Verhalten ist es notwendig, aus Erfahrung lernen zu können und das eigene Handeln entsprechend anzupassen. Auch dafür gibt es Algorithmen. Diese definieren eine Strategie, wie Erfahrungen in Form von Daten zum Lernen genutzt werden können. So kann die Leistungsfähigkeit eines Computers in Bezug auf eine bestimmte Problemstellung verbessert werden. Man gibt den Algorithmen also nicht vor, *wie* genau sie ein Problem zu lösen haben, sondern stattdessen, wie sie aus ihrer Erfahrung lernen können, das Problem *besser* zu lösen.

Der Teilbereich der künstlichen Intelligenz, der sich mit diesen Lernalgorithmen beschäftigt, wird maschinelles Lernen genannt und ist für die meisten der neuesten Erfolgsmeldungen im Bereich der künstlichen Intelligenz verantwortlich. Deshalb fokussieren wir uns in diesem Buch auf Lernalgorithmen (und nicht auf Back-, Multiplikations- oder Suchalgorithmen). Die Anwendung von maschinellem Lernen auf KI-Probleme läuft in zwei Schritten ab: Algorithmen der klassischen KI, wie zum Beispiel Suchalgorithmen, können direkt auf eine Problemstellung angewendet werden. Beim maschinellen Lernen hingegen wird zuerst mit Hilfe von Trainingsdaten (z. B. Erfahrungswerten) unter Anwendung eines Lernalgorithmus in der sogenannten Trainingsphase ein Modell gelernt. Dieses gelernte Modell kann dann in einem zweiten Schritt, der Test- oder Anwendungsphase, auf eine Problemstellung angewendet werden. Was unter dem Begriff *Modell* in diesem Zusammenhang zu verstehen ist, welche verschiedenen Modelle es im maschinellen Lernen gibt und wie diese mit Hilfe von Lernalgorithmen gelernt werden, darum geht es in diesem Buch.

Interessanterweise sind die meisten Lernalgorithmen gar nicht besonders neu, sondern wurden bereits vor Jahrzehnten entwickelt. Neu ist jedoch die gigantische Menge an Daten, dank günstiger Sensoren und dem Internet, sowie modernen Computerchips, auf denen die Lernalgorithmen besonders effizient angewendet werden können. Das bedeutet aber nicht, dass die Algorithmen selbst unwichtig geworden sind. Ganz im Gegenteil! Denn genau wie bei den Suchalgorithmen geben auch die Algorithmen des maschinellen Lernens vor, wofür man maschinelles Lernen verwenden kann und wo dessen Grenzen und Risiken liegen.

### **Treiber der Erfolgsgeschichte des maschinellen Lernens**

Das Zusammenspiel von Algorithmen, Daten und Computer-Hardware bildet das Herz der Erfolgsgeschichte des maschinellen Lernens.

Im nächsten Kapitel stellen wir das Feld des maschinellen Lernens noch genauer vor und zeigen, in welche drei Bereiche es sich einteilen lässt.