

## Die Mechanisierung des Denkens

---

Lange vor Alan Turing und Joseph Weizenbaum, nämlich schon im 17. Jahrhundert machte sich René Descartes Gedanken darüber, ob Maschinen je Sprache verstehen werden können. In seinem berühmten *Discours de la Méthode* versucht er an einer Stelle zu zeigen, dass das nicht möglich ist. Dabei geht es ihm allerdings nicht um die Fähigkeiten von Maschinen per se, sondern um die Abgrenzung des Menschen vom Tier. Für Descartes sind sowohl der menschliche als auch der tierische Körper nur Maschinen. In einem Gedankenexperiment stellt er sich vor, dass wir eines Tages eine Maschine bauen könnten, die sich innerlich und äußerlich nicht von einem Affen unterscheiden lässt. Dass das gehen könnte, scheint ihm plausibel. Dann stellt er sich vor, dass wir einen perfekten Androiden bauen könnten, also eine Maschine, die einem Menschen gleicht und sich möglichst auch wie ein Mensch verhält. Wie Turing geht es auch Descartes dabei nicht um das Aussehen, sondern um die Intelligenz. Descartes schlägt zwei »sehr sichere Mittel« vor, mit deren Hilfe man schnell herausfinden kann, dass es sich bei solchen Androiden nicht um echte Menschen handelt:

Das erste ist: Sie könnten niemals Worte oder andere Zeichen gebrauchen, indem sie sie zusammensetzen, wie wir es tun, um anderen unsere Gedanken kundzutun. Denn man kann sehr gut verstehen, dass eine Maschine so gebaut sein soll, Worte zu äußern, und man kann sogar verstehen, wenn sie einige Worte anlässlich körperlicher Vorgänge äußert, die irgendeine Veränderung in ihren Organen verursachen: etwa daß sie, wenn man sie an irgendeiner Stelle berührt, fragt, was man ihr sagen wolle, oder daß sie, berührt man sie an einer anderen Stelle, schreit, man tue ihr weh und dergleichen. Aber man kann nicht verstehen, daß sie Worte verschieden zusammenstellt, um auf den Sinn alles

dessen zu antworten, was in ihrer Gegenwart gesagt werden wird, wie es selbst die stumpfsinnigsten Menschen tun können.<sup>1</sup>

Anders als Turing kann sich Descartes nicht vorstellen, dass eine Maschine jemals Sprache beherrschen kann. Obwohl Siri und Alexa bislang den Turing-Test nicht bestehen, so dürften sie doch sprachliche Fähigkeiten besitzen, die weit über Descartes Vorstellungskraft hinausgehen. Sicher, teilweise sind die Antworten von ELIZA und Co nur vorprogrammiert, weswegen sie sich immer noch leicht als Maschinen überführen lassen, so wie Descartes sich das vorstellte. Aber Computer können durchaus »Worte verschieden [zusammenstellen]« und »Zeichen gebrauchen«, denn nichts anderes macht ELIZA. Und Watson und Wolfram Alpha können sich Antworten sogar teilweise aus ihrer Wissensdatenbank erschließen. Angesichts des aktuellen Fortschritts bei Sprachmodellen wie ChatGPT scheint Descartes Behauptung, dass keine Maschine je »Worte verschieden zusammenstellt, um auf den Sinn alles dessen zu antworten, was in ihrer Gegenwart gesagt werden wird«, längst widerlegt. Aber Descartes hatte ja noch ein weiteres Mittel vorgeschlagen.

Das zweite ist: Auch wenn solche Maschinen viele Dinge ebenso gut oder vielleicht sogar besser als irgendeiner von uns verrichten würden, würden sie unvermeidlich bei einigen anderen versagen, und anhand dieser Dinge ließe sich entdecken, daß sie nicht aus Erkenntnis tätig sind, sondern nur aus der Anordnung ihrer Organe. Denn anders als die Vernunft, die ein Universalinstrument ist, das bei allen Arten von Begabenheiten benutzt werden kann, benötigen diese Organe eine ganz bestimmte Anordnung für jede besondere Tätigkeit, und deshalb ist es praktisch unmöglich, daß es genügend viele Organe in einer Maschine gibt, um sie in allen Vorfällen des Lebens in derselben Weise wie unsere Vernunft tätig sein zu lassen.<sup>2</sup>

Descartes prophezeit weitsichtig, dass Maschinen viele Dinge besser als Menschen erledigen werden können. Aber auch an diesem Zitat sieht man, dass Descartes eine naive Vorstellung von Maschinen hat. Er kannte einfach noch keine Computer. Anders als Turing 300 Jahre spä-

---

<sup>1</sup> Descartes (2011), Fünfter Abschnitt, S. 97.

<sup>2</sup> Descartes (2011), Fünfter Abschnitt, S. 97f.

ter konnte Descartes sich nicht vorstellen, dass es je Maschinen geben könnte, die viele verschiedene Tätigkeiten flexibel ausführen können, und nicht für jede einzelne Tätigkeit eine ganz bestimmte Anordnung ihrer »Organe« benötigen. Eben solche Maschinen sind aber Computer. Mit ein paar wenigen elektronischen Organen kann man unterschiedliche Programme ausführen, die verschiedene Tätigkeiten verrichten. Dadurch, dass man mehrere Programme auf einem Computer installieren kann, ist der Computer das »Universalinstrument« schlechthin. Mehr noch: Schachprogrammen sind nicht einzelne Züge durch eine besondere »Anordnung der Organe« einprogrammiert, sondern sie überlegen sich die besten Züge je nach Stellung. Sie sind sozusagen »aus Erkenntnis tätig«.

Descartes irrt also, wenn er glaubt, dass der Gebrauch von Zeichen uns Menschen vorbehalten ist. Descartes irrt auch, wenn er glaubt, dass es keine Maschine mit genügend vielen Organen geben kann, um beliebige Tätigkeiten ausführen zu können. Seine Intuition, dass Maschinen sich mit Sprache und Vernunft schwertun, ist dadurch allerdings nicht unbedingt widerlegt – sie tun sich aber nicht aus den Gründen schwer, die Descartes anführt. Um die aktuellen Entwicklungen und die Grenzen von KI richtig einschätzen zu können, muss man zuerst verstehen, warum Descartes' Vorstellungen über Maschinen falsch sind. Man muss zuerst verstehen, warum der Computer eine wahrhaft revolutionäre Erfindung ist.

## **Uhrwerke sind die Urahnen der Computer**

Descartes kannte zwar keine modernen Computer, aber er kannte komplexe mechanische Apparate, Uhrwerke zum Beispiel. So wie das heute jeder von Kuckucksuhren kennt, gab es damals schon Spielfiguren, die sich auf ähnliche Weise bewegten und auch Geräusche machten. Angeichts solcher Automaten schien es ihm plausibel, dass die Körper von Tieren und Menschen nichts weiter als extrem komplizierte Maschinen sind. Im frühen 19. Jahrhundert waren mechanische Apparate so weit entwickelt, dass es Automaten gab, die wie menschliche Musiker aussahen und durch eine komplizierte Mechanik sich so bewegten, dass sie auf Instrumenten Stücke spielen konnten (im Deutschen Museum in München kann man in der Informatikabteilung einen mechanischen

Trompeter aus dem Jahr 1810 bewundern).<sup>3</sup> Dass körperliche Tätigkeiten von Maschinen übernommen werden können, überraschte schon damals niemanden mehr. Schließlich baut man ja Maschinen gerade deswegen, um körperliche Tätigkeiten zu automatisieren. Ein Handwebstuhl ist ein Werkzeug, das benutzt wird, um Menschen bei der körperlichen Tätigkeit des Webens zu unterstützen. Eine Webmaschine automatisiert diesen Prozess so weit, dass die sich wiederholenden Bewegungen nicht mehr von einem Menschen ausgeführt werden müssen, sondern selbstständig von der Maschine übernommen werden.

Körperliche Tätigkeiten, die sich wiederholen, lassen sich leicht mechanisieren. Was aber, wenn die Bewegungen nicht immer gleich sind? Was, wenn ich an einem Tag auf der Maschine ein Blütenmuster weben will, am nächsten aber ein Blättermuster? Auch dafür gab es schlaue Lösungen, denn die mechanischen Webmaschinen Anfang des 19. Jahrhunderts waren programmierbar. Verschiedene Webmuster konnten an einem sogenannten »Jacquardwebstuhl« durch verschiedene Lochkarten eingestellt werden.

Maschinen sind also offensichtlich in der Lage, viele körperliche Aufgaben für uns zu übernehmen. Aber wie sieht es mit geistigen Aufgaben aus? Lässt sich das Denken mechanisieren? Falls es eine Mechanik des Denkens gäbe, dann müsste man diese auch in Automaten nachbauen können, so wie das auch mit dem Spielen von Musik und dem Weben von Mustern funktioniert. Selbst im automatenbesessenen 18. Jahrhundert war man von der Technologie, die es einer Maschine etwa erlaubt, Schach zu spielen, weit entfernt. Erst am Anfang des 20. Jahrhunderts baute Leonardo Torres Quevedo in Spanien einen ersten, einfachen Schachautomaten für Turmendspiele.

Trotzdem erschienen Schachautomaten vielen Menschen schon im 18. Jahrhundert möglich, da sie Webmaschinen und Musikautomaten kannten. Wolfgang von Kempelen baute eine Apparatur, die so aussah, als ob ein »Roboter« in türkischer Tracht vor einem Schachbrett sitzt und spielt. In Wirklichkeit wurde diese Figur über eine ausgeklügelte Mechanik von einem Menschen gesteuert, der so geschickt im Inneren der Maschine versteckt war, dass es selbst heutige Zauberkünstler noch verzückt. Jedenfalls entstand der überzeugende Eindruck, der Automat würde Schach spielen. Dieser Automat, der sogenannte »Schachtürke«, wurde europaweit bestaunt. Genau wie bei heutiger KI wurden pub-

---

<sup>3</sup> Siehe <https://digital.deutsches-museum.de/item/4423/>.

likumswirksame Duelle mit den besten Schachspielern veranstaltet – und angeblich mit vielen Herrschern.<sup>4</sup> Der Schachroboter war damals die Hauptattraktion einer äußerst erfolgreichen Automatenshow, die durch ganz Europa tingelte. Nicht immer wurde der Taschenspielertrick durchschaut und die Zeitungen waren voll von reißerischen Berichten über dieses Wunderwerk der Technik: ein denkender Automat. Als er später in Amerika zu sehen war, wurde so ausgiebig über diese Sensation berichtet, dass eine Zeitung sogar das Gefühl hatte, sich für die andauernde Berichterstattung entschuldigen zu müssen.<sup>5</sup>

Die eigentliche Computer-Revolution startete schon vorher und war weit weniger spektakulär. Dabei ging es nicht um Show-Veranstaltungen, bei denen Menschen gegen Maschinen antraten, sondern um ganz praktische Probleme. Auch unter geistigen Tätigkeiten gibt es solche, die extrem mühsam sind und sich ständig wiederholen. So beschwerte sich der berühmte Universalgelehrte Gottfried Wilhelm Leibniz schon im 17. Jahrhundert, dass er seine Zeit mit »knechtischen Rechenarbeiten« verschwenden musste.<sup>6</sup> Kein Wunder, dass er an einer Maschine arbeitete, die ihm diese lästige Tätigkeit abnehmen sollte. Da es schon damals einen großen Bedarf an Berechnungen für praktische Probleme gab, etwa in der Buchhaltung oder bei Landvermessungen, war er nicht der einzige, der an Rechenmaschinen arbeitete. Der Astronom Wilhelm Schickard in Württemberg arbeitete genauso an ihnen wie der Mathematiker, Physiker und Philosoph Blaise Pascal in Frankreich.

Diese frühen mechanischen Rechenapparate funktionierten nach ähnlichen Prinzipien wie alte Handzähler oder die Registrierkasse im

<sup>4</sup> Würde es Sie heutzutage interessieren, den Bundeskanzler in dem Computerspiel StarCraft gegen eine KI spielen zu sehen?

<sup>5</sup> Das Buch von Standage (2002) erzählt die aufregende Geschichte dieses angeblichen Schachautomaten, die sich bis ins 19. Jahrhundert erstreckt. Obwohl die meisten Berichte darüber, dass er gegen verschiedene Herrscher gespielt hat, wohl falsch sind, ist ein Spiel gegen Napoleon gut belegbar (Kap. 7). Standage zitiert auch einige der reißerischen Zeitungsberichte (Kap. 9, die Entschuldigung über die ständige Berichterstattung findet sich auf S. 153). Viele Leute waren trotzdem zu Recht skeptisch und davon überzeugt, dass es sich um einen elaborierten Streich handeln musste. So auch der Schriftsteller Edgar Allan Poe, der von Kempelens Schachautomaten einen langen Artikel widmete (Poe, 1836; Standage, 2002, Kap. 10).

<sup>6</sup> Der vollständige, häufig zitierte Ausspruch lautet: »Es ist unwürdig, die Zeit von hervorragenden Leuten mit knechtischen Rechenarbeiten zu verschwenden, weil beim Einsatz der Maschine auch der Einfältigste die Ergebnisse sicher hinschreiben kann.« (Poser, 2016, S. 388)

Tante-Emma-Laden. Dabei beherrschten sie teilweise nicht einmal alle Grundrechenarten, wenn sie überhaupt verlässlich funktionierten. Die komplizierte Feinmechanik in diesen Geräten war damals nur schwer in ausreichender Präzision herzustellen. Selbst Charles Babbage, dessen *Analytical Engine* als der erste Entwurf für einen modernen Computer gilt, scheiterte noch im 19. Jahrhundert an den Tücken der Feinmechanik. (Babbage hat angeblich auch zweimal gegen von Kempelens Schachautomaten verloren, obwohl er den Taschenspielertrick durchschaute.)<sup>7</sup>

## Wie Computer rechnen

Erst Fortschritte in Elektrotechnik und Elektronik erlaubten es seit der Mitte des 20. Jahrhunderts, auf mechanische Teile zu verzichten und elektronische Rechenmaschinen präzise und billig genug zu fertigen, um im großen Stil eingesetzt zu werden. Durchgesetzt hat sich dabei ein Entwurf, der auf den ungarisch-amerikanischen Mathematiker John von Neumann zurückgeht, weshalb moderne Computer manchmal auch *Von-Neumann-Rechner* genannt werden.

Die praktischen Details, wie man einen mechanischen oder elektronischen Computer baut, sodass alle Teile verlässlich und schnell zusammenarbeiten, waren bei der Entwicklung des modernen Computers entscheidend. Aber genauso wichtig war es für die Computerpioniere, erst einmal theoretisch zu verstehen, was genau ein Computer eigentlich ist. Alan Turing erfand 1937 eine sehr einfache, hypothetische Maschine, die seither als *Turingmaschine* bekannt ist.<sup>8</sup> Obwohl richtige Computer nicht nach dem Vorbild der Turingmaschine gebaut werden, erlaubt es diese hypothetische Maschine, die entscheidenden Aspekte eines Computers genau zu verstehen.

Dabei ist es überraschenderweise gar nicht so wichtig zu verstehen, wie genau die Maschine im Detail mit mechanischen oder elektronischen Teilen gebaut werden könnte. Die Idee, mit der Turing startet, ist einfach: Die Maschine soll genau das machen, was Menschen machen, die mit Papier und Bleistift rechnen. Nehmen wir als Beispiel die

<sup>7</sup> Siehe Michel (2012) und Standage (2002).

<sup>8</sup> Siehe Turing (1937). Eine kurze und verständliche Einführung findet sich in Kap. 28 des schönen Buches von Dewdney (1989).

schriftliche Addition. Wir wollen wissen, was  $932+92$  ist. Das können wir natürlich leicht im Kopf ausrechnen. Aber versuchen wir uns zu erinnern, wie wir in der Schule gelernt haben, zwei Zahlen schriftlich zu addieren. Zuerst schreiben wir dazu die zwei Zahlen auf unserem karierten Papier rechtsbündig untereinander. Um uns zu erinnern, dass wir addieren wollen, schreiben wir ein Plus vor jede Zahl und zur besseren Lesbarkeit fügen wir auch noch führende Nullen ein, ziehen zwei Striche darunter und markieren die Kästchen, in die wir das Ergebnis schreiben wollen, durch Fragezeichen:

$$\begin{array}{r}
 +932 \\
 +092 \\
 \hline
 \end{array}$$

????

Jetzt stellen wir uns vor, dass es eine Maschine gibt, die mit einem Lese- und Schreibkopf über die Kästchen des karierten Papiers fahren kann. Die Maschine kann erkennen, welches Zeichen in einem Kästchen steht und das Zeichen löschen und überschreiben. Genau das passiert auch im Speicher eines Computers. Der Speicher eines Computers ist wie ein sehr großes Kästchenpapier, das mit Bleistift und Radierer beliebig beschriftet und überschrieben werden kann. Sie können ja mal überlegen, wie Sie so eine Maschine mit mechanischen Teilen bauen würden. Das ist gar nicht so leicht. Mit elektronischen Bauteilen ist es einfacher. Für unsere Zwecke reicht es aber zu wissen, dass es geht.

Neben dem Papier und dem Lese-Schreib-Kopf braucht die Maschine noch einen Mechanismus, der den Kopf steuert. Abhängig davon, was die Maschine einliest, soll sie bestimmte Ausgaben machen. Die Maschine könnte zum Beispiel so konstruiert sein, dass sie erst die erste und dann die zweite Zeile Zeichen für Zeichen einliest und abhängig davon, was sie gelesen hat, nacheinander unterschiedliche Ziffern in die letzte Zeile schreibt. Wieder wäre es nicht leicht, so einen Mechanismus mechanisch zu bauen, aber elektronisch ist das heutzutage ohne weiteres möglich.

Eine schlichte Maschine – so wie sie sich Descartes wohl vorgestellt hatte – hätte für jede mögliche Eingabe einen eigenen Mechanismus, der bestimmt, was der Automat in dieser speziellen Situation machen soll. Wenn die Maschine in der ersten Zeile die vier Zeichen  $+001$  nacheinander liest und dann in der zweiten die vier Zeichen  $+001$ , dann

fährt sie in die letzte Zeile und beschreibt diese nacheinander mit 0002. Wenn die Maschine in der ersten Zeile +001 liest und in der zweiten aber +002, dann schreibt sie in die letzte Zeile 0003.<sup>9</sup> Und so weiter für alle möglichen Eingaben. Für jede mögliche Kombination an Eingaben müsste man einen speziellen Mechanismus in die Maschine einbauen, der für diese Situation die richtige Ausgabe produziert. So eine Maschine würde nicht wirklich rechnen, sondern hätte lediglich die richtigen Antworten für alle dreistelligen Additionsaufgaben fest verdrahtet. Wenn wir die Eingaben in den ersten zwei Zeilen auf die Zahlen von 0 bis 999 beschränken, dann gibt es 1000 mal 1000, also eine Million mögliche Eingaben. Wenn unser Kästchenpapier nicht nur 4 mal 4 Kästchen groß ist, sondern 5 Kästchen breit, dann wären es schon 100 Millionen. Bei 6 Kästchen 10 Milliarden. Wenn wir beliebig große Zahlen addieren wollen, dann ist es tatsächlich unmöglich, eine Maschine mit so vielen »Organen« zu bauen. Auf diese Weise kann man sicher keine funktionierende Rechenmaschine – oder gar Schachmaschine – bauen. Damit hatte Descartes recht.

Eine Maschine, in der alle Antworten fest verdrahtet sind, ist ohnehin nicht, was wir wollen. Wir wollen eine Maschine bauen, die Zahlen addiert wie der Mensch. Bei der schriftlichen Addition fangen wir in dem Kästchen rechts oben an, und deshalb muss die Maschine ihren Lese-Schreib-Kopf als Erstes da hinfahren. Dann brauchen wir einen Mechanismus, der die Spalte nach unten fährt und die Zeichen nacheinander liest. In unserem Beispiel liest die Maschine dann erst eine 2, dann noch eine 2 und dann das =. Zum Addieren brauchen wir einen Mechanismus, der abhängig davon, was in einer Spalte gelesen wurde, in die unterste Zeile der gleichen Spalte unterschiedliche Ziffern schreibt. Wenn die erste Zeile eine 2 war und die zweite eine 2 und die dritte ein =, dann muss in die letzte Zeile eine 4 geschrieben werden. Ebenso brauchen wir natürlich jeweils weitere Teilmechanismen für alle anderen Paare von Ziffern zwischen 0 und 9, die in den zwei Zeilen stehen könnten. Das sind 10 mal 10 Ziffern, macht Hundert fest verdrahtete »Organe«, also wesentlich weniger als die Millionen, die wir vorher hatten. Nachdem die Maschine so die rechte Spalte verarbeitet hat, steht auf unserem Papier:

---

<sup>9</sup> Falls Sie sich über die zusätzliche führende Null wundern: Die Summe von +500 und +500 ist 1000. Wir brauchen deshalb für die Ausgabe eine Ziffer mehr als für die Eingaben.

$$\begin{array}{r}
 +932 \\
 +092 \\
 \hline
 \hline
 \end{array}
 \quad
 \begin{array}{r}
 \text{?} \\
 \text{?} \\
 \text{?} \\
 4
 \end{array}$$

Dann muss die Maschine ihren Lesekopf wieder in die erste Zeile zurückfahren und ein Kästchen nach links versetzen. Wenn die Maschine da kein + liest, sondern eine Ziffer – wie hier die 3 –, ist die Rechnung noch nicht fertig. Und wir benutzen denselben Mechanismus wie zuvor, um die nächste Spalte zu verarbeiten. Wenn in der ersten Zeile eine 3 steht, in der zweiten eine 9 und in der dritten ein =, schreibt der Mechanismus in die letzte Zeile eine 2 und merkt sich für die nächste Spalte eine 1, indem er dort das = überschreibt. Das ist die ›Eins im Sinn‹ (oder ›Eins gemerkt‹) und für diesen Fall brauchen wir weitere Regeln, um damit korrekt umzugehen. So kann die Maschine bis zur ersten Spalte, in der links oben ein + steht, weitermachen. Danach steht in der letzten Zeile das vollständige Ergebnis:

+932	+932	+932
+092	+092	+092
=1==	11==	11==
??24	?024	1024

Die Maschine führt den Algorithmus zur schriftlichen Addition aus, den wir in der Schule gelernt haben. Ein Algorithmus besteht aus einer Folge von regelbasierten Anweisungen in der Form, wie wir sie bei der Beschreibung der Maschine ausführlich genutzt haben, zum Beispiel: Wenn Du beim Runterfahren einer Spalte eine 1, eine 2 und ein = liest, dann schreib in die letzte Zeile eine 3 und fahre mit der nächsten Spalte fort. Diese Anweisungen müssen unmissverständlich sein und niemals darf der Fall eintreten, dass man nicht weiß, was man tun soll. Nur dann spricht man von einem Algorithmus. Eine Maschine, die einen Algorithmus ausführen soll, braucht für jede Anweisung einen entsprechenden Mechanismus.

Die Anzahl der »Organe«, die unsere Additionsmaschine braucht, ist immer noch recht groß, weil die Anzahl der Rechenregeln, die wir benutzt haben, recht groß ist. Für jedes Paar der Ziffern von 0 bis 9 brauchen wir eine Regel, also einen Mechanismus, der die passende Zahl in die jeweils letzte Zeile schreibt. Das macht 10 mal 10 Regeln. Für

die ›Eins im Sinn‹ brauchen wir noch zusätzliche Regeln. Die meisten Menschen, die nicht mehr besonders viel mit der Hand rechnen, können wahrscheinlich nicht mal alle diese 10 mal 10 Regeln auswendig (tatsächlich muss man sich ja nur knapp die Hälfte davon merken, weil  $5+3$  dasselbe ist wie  $3+5$ ). Man kann zwar wesentlich schneller rechnen, falls man auswendig weiß, dass  $5+3=8$  ist, aber im Notfall kann man auch an den Fingern abzählen. Dieses Abzählen könnte man zusätzlich in unsere Additionsmaschine einbauen, in der Hoffnung, die Maschine dadurch weniger komplex zu machen.

Eine andere und radikalere Vereinfachung unserer Additionsmaschine erreicht man, indem man statt mit zehn Ziffern nur mit zwei Ziffern rechnet – also statt dem sonst üblichen Dezimal- das Binärsystem benutzt. So braucht man statt 10 mal 10 nur noch 2 mal 2 (also 4) Regeln für die Additionsmaschine.<sup>10</sup> Turing hat die Mechanismen für seine hypothetische Rechenmaschine noch weiter vereinfacht, indem er statt des handelsüblichen Karopapiers einen langen Papierstreifen an Kästchen nutzt. Statt die Zahlen untereinander zu schreiben, werden sie in einer solchen Turingmaschine nebeneinander geschrieben. Dadurch wird der Additionsalgorithmus etwas komplizierter, weil die Ziffern, die addiert werden sollen, nicht direkt untereinander stehen. Dafür kann man den Papierstreifen mit einer einfachen Mechanik unter dem Lese-Schreib-Kopf hin- und herbewegen und muss keine komplizierte Apparatur bauen, um ihn in zwei Dimensionen zu bewegen. Falls man jetzt einen beliebig langen Papierstreifen zur Verfügung hat, kann man mit dieser endlichen Maschine beliebig große Zahlen addieren.

Wer weiß, wie man eine Additionsmaschine baut, kann mit den gleichen Prinzipien auch eine Subtraktions- oder Multiplikationsma-

---

<sup>10</sup> Man unterscheidet zwischen einer Ziffernfolge und der Zahl, die durch die Ziffernfolge dargestellt wird. Dieselbe Zahl kann in verschiedenen Systemen durch unterschiedliche Ziffernfolgen dargestellt werden. Die Ziffernfolge 42 im Dezimalsystem bedeutet 4·10+2·1: vier Zehner und zwei Einser. Im Dezimalsystem steht jede Ziffer für eine Zehnerpotenz. Die Einser sind  $10^0$ , die Zehner sind  $10^1$ , die Hunderter sind  $10^2$ , die Tausender sind  $10^3$  und so weiter. Im Binärsystem ist die Basis nicht 10, sondern 2. Statt Zehnern, Hundertern und Tausendern gibt es die entsprechenden Zweierpotenzen: Zweier, Vierer und Achter. Die Einser sind  $2^0$ , die Zweier sind  $2^1$ , die Vierer sind  $2^2$  und die Achter sind  $2^3$ . Im Binärsystem wird die Zahl, die wir üblicherweise als die Ziffernfolge 42 schreiben, durch die Ziffernfolge 1·32+0·16+1·8+0·4+1·2+0·1.

schine bauen – oder eine Maschine für beliebige andere Berechnungen, die man mit Papier und Bleistift durchführen kann. All diese geistigen Tätigkeiten lassen sich mit solchen Turingmaschinen mechanisieren. Leibniz wäre begeistert. Endlich hat jemand die Technologie erfunden, die ihn von der Knechtschaft des Rechnens erlöst. Was für viele von Leibniz' Zeitgenossen noch ein Wunder der Technik gewesen wäre, ist heutzutage alltäglich: Computer können rechnen. Daher der Name. Das ist wirklich total nützlich. Aber für alle, die nicht irgendein MINT-Fach studieren oder Buchhalter sind, auch irgendwie langweilig. So langweilig wie Rechnen in der Schule. Aber warum kann man mit Computern eigentlich auch Texte schreiben, Bilder bearbeiten oder Schach spielen? Und was hat das alles mit Rechnen zu tun?

## Computer verarbeiten Zeichen

Keines der Konstruktionsprinzipien, die wir für unsere Maschine benutzt haben, ist spezifisch für Zahlen. Einer solchen Maschine ist es egal, ob sie Ziffern, Buchstaben oder andere Zeichen als Eingaben bekommt. Wichtig ist nur, dass es unmissverständliche Regeln gibt, die beschreiben, wie die Zeichen verarbeitet werden sollen. In unserem Alltag hat Rechnen etwas mit Zahlen zu tun. Für Informatiker hingegen bedeutet Rechnen ganz allgemein, irgendwelche Zeichen zu verarbeiten. Computer sind eigentlich keine Rechenmaschinen, sondern regelbasierte Zeichenverarbeitungsmaschinen. All die Sachen, die Computer so tun können, sind schlicht Zeichenverarbeitungsaufgaben. Ada Lovelace, die berühmte Pionierin der modernen Informatik, hat das schon 1843 verstanden, als sie über die Analytical Engine schrieb, dass diese Maschine Musik komponieren oder mathematische Formeln beweisen könnte und dabei »algebraische Muster webt, wie der Jacquardwebstuhl Blüten und Blätter«.<sup>11</sup>

Dazu ein Beispiel aus der Sprachverarbeitung, das an ELIZA erinnert: Wir wollen eine Maschine bauen, die einfache Aussagesätze in Warum-Fragen umwandelt. Der Satz **ICH BIN TRAURIG** soll zu **WARUM BIST DU TRAURIG** werden oder **ER IST EIN TYRANN** soll zu

---

<sup>11</sup> Im Original: »weaves algebraical patterns just as the Jacquard loom weaves flowers and leaves« (Hollings, Martin & Rice, 2020). Siehe auch Toole (1998), S. 179 und S. 182.

WARUM IST ER EIN TYRANN werden. Dabei beschränken wir uns der Einfachheit halber auf Sätze, die mit einem Personalpronomen und dem Verb »sein« gebildet werden. Die Eingabe schreiben wir wieder auf das Karopapier, ein Wort in jede Zeile, also zum Beispiel:

ICH  
BIN  
TRAURIG

Die Maschine soll sich nach folgenden Regeln verhalten: Lies das Wort in der ersten Zeile. Wenn es **ICH** ist, ersetze es durch **DU**. Wenn es **DU** ist, ersetze es durch **ICH**. Wenn es **ER** oder **SIE** ist, mache nichts. Ähnliche Regeln gelten für **WIR**, **IHR** und **SIE**. Nachdem diese Anweisungen abgearbeitet sind, steht auf dem Papier:

DU  
BIN  
TRAURIG

Dann liest die Maschine das Wort in der nächsten Zeile. Wenn es **BIN** ist, dann ersetzt sie es durch **BIST**. Wenn es **BIST** ist, ersetzt sie es durch **BIN**. Wenn es **IST** ist, macht sie nichts. Wieder gelten ähnliche Regeln für den Plural. Jetzt steht auf dem Papier:

DU  
BIST  
TRAURIG

Jetzt vertauscht die Maschine die erste und die zweite Zeile. Das kann sie machen, indem sie als Zwischenspeicher eine Zeile vor der ersten Zeile neu belegt und dort die alte erste Zeile hineinkopiert. Die neue erste und zweite Zeile enthalten jetzt im Beispiel beide das Wort **DU**. Dann überschreibt sie mit der jetzigen dritten Zeile, im Beispiel dem Wort **BIST**, die zweite Zeile. Danach kopiert sie die neue erste Zeile in die dritte. Während wir noch dieses dadaistische Gedicht auf uns wirken lassen, greift der letzte Mechanismus in der Maschine und überschreibt die erste Zeile mit dem Wort **WARUM**:

DU	DU	DU	WARUM
DU	BIST	BIST	BIST
BIST	BIST	DU	DU
TRAURIG	TRAURIG	TRAURIG	TRAURIG

Diese Maschine ist weit davon entfernt, jeden beliebigen Aussagesatz in eine Frage umwandeln zu können. Bauen wir der Maschine aber immer mehr Regeln zur Verarbeitung von Wörtern und Zeichen ein, wird ihr Verhalten auch immer komplexer und interessanter. Das sind genau die Mechanismen, die ELIZA erlauben, die Illusion einer Konversation zu erzeugen. Wir haben im letzten Kapitel gesehen, dass es bisher nicht gelungen ist, auf diese Art eine Maschine zu bauen, die den Turing-Test besteht. Trotzdem: Was war gleich wieder Dein erstes sicheres Mittel, um Maschinen von Menschen zu unterscheiden, Descartes? »Sie könnten niemals Worte oder andere Zeichen gebrauchen«?

Aber es gab ja noch das zweite sichere Mittel: »Auch wenn solche Maschinen viele Dinge ebenso gut oder vielleicht sogar besser als irgendeiner von uns verrichten würden, würden sie unvermeidlich bei einigen anderen versagen«, woraus man schließen könnte, dass es Maschinen sind. »Denn anders als die Vernunft, die ein Universalinstrument ist, das bei allen Arten von Begebenheiten benutzt werden kann, benötigen diese Organe eine ganz bestimmte Anordnung für jede besondere Tätigkeit«. Das kann man so lesen: Für jede Aufgabe, die eine Maschine erledigen soll, muss man einen Mechanismus einbauen, der genau das macht. Aber dann bräuchte man echt viele »Organe« – viel zu viele.

Bisher sieht es so aus, als ob Descartes da irgendwie schon recht hat. Wir haben eine Maschine gebaut, die addieren kann. Mit ähnlichen Prinzipien können wir auch Maschinen bauen, die subtrahieren, multiplizieren, dividieren, Sprache verarbeiten oder Schach spielen können. Und im Prinzip könnten wir all diese Mechanismen auch in eine einzige Maschine packen. Allerdings würde diese Maschine dann zu komplex, als dass man sie tatsächlich noch bauen könnte. So funktioniert aber auch kein moderner Computer.

## Computer sind programmierbare Maschinen

Ein Computer hat nicht für jede Aufgabe, die er erledigen soll, einen eigenen Mechanismus, der fest in seiner Hardware eingebaut ist. Jedes Kind weiß heute, was Descartes sich nicht vorstellen konnte: Der Computer kann ganz leicht eine neue Fähigkeit lernen, indem man die passende Software installiert. Dann startet man das installierte Programm, und der Computer macht, was man will (also im Prinzip, außer es klappt mal wieder was nicht).

Aber was ist eigentlich ein Programm? Bei der Waschmaschine gibt es auch unterschiedlichen Programme. Je nachdem, ob ich Wolle oder Buntwäsche in die Maschine fülle, will ich, dass die Maschine unterschiedliche Sachen macht. Was genau das ist, wird durch das Programm bestimmt. Webstühle und später auch Strickmaschinen konnte man früher mit Lochkarten programmieren. Je nachdem, welche Lochkarte eingesteckt war, führte die Maschine unterschiedliche Web- oder Strickmuster aus. Ein Programm besteht aus Anweisungen, die einer Maschine – zum Beispiel über eine Lochkarte – sagen, was sie wann machen soll.

Unsere Maschinen, die einen Lese-Schreib-Kopf besitzen und diesen über Karopapier fahren, sind besondere Maschinen. Statt Material wie Garn, Wolle oder Wäsche zu verarbeiten, wurden sie dafür gebaut, Zeichen zu verarbeiten. Diese Tatsache macht es uns leicht, eine programmierbare Maschine zu bauen.

Bauen wir also so eine programmierbare Maschine! Wir haben weiterhin einen Lese-Schreib-Kopf, der mit Bleistift und Radiergummi bestückt ist, und ein Karopapier, über das die Maschine fahren kann. Jetzt spendieren wir der Maschine ein zweites Blatt Karopapier. Um die Blätter voneinander zu unterscheiden, nennen wir das erste ›Arbeitsblatt‹ und das zweite ›Programmblatt‹. Weiterhin stellen wir uns vor, dass beide Blätter so groß sind, dass uns niemals der Platz ausgeht. (Wem das zu unrealistisch ist, der kann sich stattdessen eine Maschine mit großen Papierrollen vorstellen oder mit Endlospapier, wie es bei alten Nadeldruckern benutzt wurde.) Das Arbeitsblatt nutzen wir genauso wie zuvor. Wir schreiben zum Beispiel den Satz **ICH BIN TRAURIG** darauf und wollen ihn wieder in eine Frage umwandeln. Statt wie zuvor eine Maschine zu konstruieren, die genau macht, was wir wollen, schreiben wir die Anweisungen und Regeln auf das Programmblatt. Da könnte stehen:

```

LIES ZEILE
WENN ZEILE == "ICH" DANN ZEILE := "DU"
WENN ZEILE == "DU" DANN ZEILE := "ICH"
WENN ZEILE == "ER" DANN ZEILE := "ER"
WENN ZEILE == "SIE" DANN ZEILE := "SIE"
...
NÄCHSTE ZEILE
LIES ZEILE
WENN ZEILE == "BIN" DANN ZEILE := "BIST"
...

```

Die eigentliche Maschine konstruieren wir jetzt so, dass sie dieses Programm ausführen kann. Dazu muss die Maschine die Zeichen, die auf dem Programmpapier stehen, interpretieren. Die Maschine soll immer genau das machen, was mit den Zeichen auf dem Programmblatt von uns gemeint war. Dazu liest die Maschine das Programmblatt Zeile für Zeile ein und führt je nachdem, was sie liest, unterschiedliche Aktionen aus. Wenn sie `LIES ZEILE` liest, dann liest sie eine Zeile auf dem Arbeitsblatt. Wenn sie `WENN ZEILE == "ICH"` liest, dann liest sie den Rest der Programmzeile nur, wenn in der aktuellen Zeile des Arbeitsblattes `ICH` steht. Wenn die Maschine `DANN ZEILE := "DU"` liest, überschreibt sie die aktuelle Zeile des Arbeitsblattes mit `DU`, und so weiter.

Auf diese Weise können wir eine Maschine bauen, die die Zeichen auf dem Arbeitsblatt so verarbeitet, wie die Zeichen auf dem Programmblatt es ihr vorgeben. Indem wir unterschiedliche Programme auf das Programmblatt der Maschine schreiben, kann diese neue Maschine andere Papier-und-Bleistift-Maschinen nachahmen, wie zum Beispiel unsere Warum-Frage-Maschine oder unsere Additionsmaschine. Tatsächlich kann man die Maschine leicht so bauen, dass sie alle anderen Papier-und-Bleistift-Maschinen durch ein entsprechendes Programm nachahmen kann. Eine programmierbare Maschine, die diese Eigenschaft hat, nennt man *›universell‹*.

Moderne programmierbare Computer sehen im Detail ganz anders aus als unsere Papier-und-Bleistift-Maschine. Weder gibt es da Karopapier, noch gibt es einen Lese-Schreib-Kopf, der Zeichen auf das Papier malt und wieder wegradieren kann. Aber man kann zeigen, dass alles, was ein moderner, elektronischer Computer kann, auch von unserer universellen Papier-und-Bleistift-Maschine nachgeahmt wer-

den kann, und umgekehrt. Statt mit Bleistift auf Papier schreibt ein Computer Bits magnetisch und elektronisch in seine Speicher. Beide Maschinen sind universelle Computer; Computer, die alles berechnen können, was ein anderer Computer auch kann. Tatsächlich lässt sich jede Zeichenverarbeitungsmaschine, die jemals irgendjemand auf der Welt erfunden hat, auf unserer universellen Papier-und-Bleistift-Maschine nachahmen. Jeder Mac, jeder PC, jeder Supercomputer, ja, sogar jeder Quantencomputer.<sup>12</sup>

Lesen Sie den letzten Absatz noch einmal! Das ist der Grund, warum programmierbare Computer eine wahrhaft revolutionäre Erfindung sind. Statt für jede Zeichenverarbeitungsaufgabe mühselig eine neue Maschine bauen zu müssen, brauchen wir lediglich ein neues Programmblatt. Wir brauchen nur noch eine Maschine für alles. Das ist der Grund, warum Computer überall sind. Auf Ihrem Schreibtisch, in Ihrem Mobiltelefon, in Ihrem Auto und sogar in Ihrer Waschmaschine. Immer wenn eine Maschine unterschiedliche Aufgaben für uns erledigen soll oder in irgendeiner Form Zeichen verarbeiten muss, baut man einfach einen Standardcomputer ein und programmiert ihn, wie man es braucht. Ihr Mobiltelefon soll Ihre Waschmaschine fernsteuern? ›There's an app for that.‹ Tja, lieber Descartes, der Computer ist in gewisser Weise das ultimative »Universalinstrument«. Eine einzige Maschine kann unglaublich viele verschiedene Aufgaben erledigen; einfach, indem man das passende Programm lädt.

In dieser Hinsicht ist der Computer uns Menschen nicht unähnlich. Kein Kind wird mit der Fähigkeit geboren, schriftliche Addition mit Papier und Bleistift auszuführen. Das Kind besitzt keine speziellen »Organe« für Addition. Ein Kind, das in der Schule den Additionsalgorithmus lernt, installiert gewissermaßen eine App: ein Programm, das das Kind bei Bedarf ausführen kann. Genau diese Fähigkeit, Anweisungen und Regeln zu speichern und auszuführen, haben sich die Erfinder des Computers ja gerade beim Menschen abgeschaut.

Oft scheint es aber, als ob Computer viel mehr können als wir. Sie können besser rechnen, besser Schach spielen und vielleicht bald auch

---

<sup>12</sup> Da die Informatiker sich wirklich viel Mühe gegeben haben, Maschinen zu erfinden, die mehr können als die einfachste aller Papier-und-Bleistift-Maschinen, die Turingmaschine, und bisher immer gescheitert sind, glaubt keiner, dass es eine ›natürliche‹ Zeichenverarbeitungsmaschine gibt, die mehr als die Turingmaschine kann. Das ist Church's These der theoretischen Informatik. Siehe Kapitel 60 in Dewdney (1989).

besser Auto fahren. Und es stimmt, sie können all das schneller und zuverlässiger. Aber alles, was Computer machen, ist schlussendlich nur Verarbeitung von Zeichen. Diese Zeichenverarbeitung könnte auch von einem Menschen, der Anweisungen und Regeln folgen kann, mit genügend Zeit, Papier und Bleistiften erledigt werden. Doch weil der Mensch viel langsamer Zeichen verarbeitet und dabei auch mehr Fehler macht, sind wir froh, dass es Computer gibt. Aber kein Computer, der bisher gebaut wurde, kann Aufgaben bearbeiten, die nicht im Prinzip auch von einem Menschen erledigt werden könnten. Kein Wunder, denn der Mensch war das Vorbild für unsere Papier-und-Bleistift-Maschine – und für Computer überhaupt! In der Vergangenheit verübten Computer zwar meist nur langweilige Aufgaben, für die es keine Intelligenz braucht, aber das heißt nicht, dass KI eine neue Entwicklung der letzten paar Jahre ist. Von Anfang an ging es bei der Entwicklung von Computern auch darum, das Denken zu mechanisieren.

## Verstehen Computer Sprache?

Der aktuelle Fortschritt in der KI zeigt, dass die Informatik mit der Mechanisierung des Denkens weit gekommen ist. Computer spielen Schach und sogar Go, das viel schwieriger als Schach ist. Computer steuern selbstständig Flugzeuge und Autos. Und Computer verarbeiten Sprache. Descartes wäre sicher beeindruckt und er würde schnell einsehen, dass er eine recht naive Vorstellung davon hatte, was eine Maschine ist. Trotzdem würde Descartes vielleicht dabei bleiben: Auch diese neumodischen Zeichenverarbeitungsmaschinen verstehen keine Sprache und werden es auch nie können.

Er wäre nicht der einzige, der das glaubt. Zwar kann ein Computer zweifellos Zeichen und Worte verarbeiten, aber er versteht dabei nicht, was die Zeichen bedeuten. Er folgt lediglich den Regeln, die ihm ein Programmierer vorgegeben hat. Der Philosoph John Searle hat diese Ansicht mit seinem Gedankenexperiment vom >Chinesischen Zimmer< gelungen veranschaulicht.<sup>13</sup>

Stellen Sie sich vor, sie können kein Chinesisch (für die meisten Leser wahrscheinlich nicht sehr schwer). Sie sitzen in einem Zimmer und der einzige Kontakt zur Außenwelt ist über eine Klappe, durch die

---

13 Siehe Searle (1980).

jemand chinesische Zeichen zu Ihnen hereinreicht. Sie können durch dieselbe Klappe chinesische Zeichen wieder herausgeben. Andere Zeichen werden nicht akzeptiert und kommen postwendend wieder zurück. Glücklicherweise hat Ihnen jemand einen großen Stapel Papier und genügend Bleistifte und Radiergummis bereitgestellt. Daneben gibt es ein sehr dickes Buch mit detaillierten Anweisungen und Regeln auf Deutsch, die genau bestimmen, was Sie machen sollen, wenn bestimmte chinesische Zeichen durch die Klappe gereicht werden. Sie antworten dann entsprechend dem Regelbuch mit chinesischen Zeichen, die sie wieder herausreichen. Sie sind eine Papier-und-Bleistift-Maschine. Ihr Stapel Papier ist Ihr Arbeitsblatt und das Buch ist Ihr Programmblatt. In dem Gedankenexperiment wird nun angenommen, dass das Regelbuch so gut durchdacht ist, dass ein Chineser, der mit Ihnen Zeichen über die Klappe austauscht, denkt, er würde sich mit jemandem austauschen, der Chinesisch kann. Sie würden den chinesischen Turing-Test bestehen.

Trotzdem verstehen Sie natürlich kein Wort Chinesisch. Und genauso versteht auch kein Computer, was die Zeichen, die er verarbeitet, eigentlich bedeuten. Er folgt nur stumpf den Regeln und Anweisungen auf seinem Programmblatt, ohne jegliches Verständnis. Im Fall von ChatGPT und anderen Sprachmodellen wurde das Regelbuch nicht von Menschen geschrieben. Stattdessen basiert die Zeichenverarbeitung auf statistischen Regeln. Diese statistischen Regeln wurden automatisch – von einem anderen Computerprogramm – aus großen Mengen von Text extrahiert und in einem Sprachmodell zusammengefasst. Das ändert aber nichts an Searles Argument, dass ein Computer zwar durchaus Sprache verarbeiten, aber keineswegs verstehen kann.

Dazu müsste der Computer wissen, was die Zeichen, die er verarbeitet, bedeuten. Es ist denkbar, dass ein Roboter, der mit verschiedenen Sensoren ausgestattet ist, weiß, dass das Wort »warm« einen ganz bestimmten Zustand seines Temperatursensors bezeichnet. Es müsste nur auf seinem Programmblatt vermerkt sein, dass meistens, wenn jemand das Wort »warm« sagt, der Sensor Werte über 17° Celsius anzeigt. Bei Bedarf könnte er die Bedeutung dort nachschlagen, und er wüsste, was das Wort »warm« für seine Sensoren bedeutet. Es ist sogar denkbar, dass ihm dieses Wissen nicht einprogrammiert werden muss, sondern dass er es von alleine lernt, indem er sich auf seinem Arbeitsblatt immer notiert, was der Temperatursensor gerade gezeigt hat, als jemand das Wort »warm« benutzt hat. So könnte er auch lernen und sich

merken, dass es im Sommer oft warm ist und sein Helligkeitssensor aufgrund der Sonne gleichzeitig auch stark ausschlägt. Und um zu lernen, wie die Sonne golden leuchten kann und wie Knospen und Blüten im Mai aussehen, könnte man solche sprachlichen Beschreibungen mit Kamerabildern kombinieren. Auf diese Weise können tatsächlich Sprachmodelle entwickelt werden, die auch Bilder verstehen. Aber wie kann ein Computer, der niemals geliebt hat, jemals ein Gedicht verstehen, in dem jemand mit einem Sommertag verglichen wird?

Descartes, der jetzt versteht, dass Computer universelle Zeichenverarbeitungsmaschinen sind, wäre daher vielleicht trotzdem noch überzeugt, dass Computer niemals wirklich Sprache genauso wie wir verstehen werden. Zumindest nicht, solange sie nicht die gleichen Erfahrungen machen wie wir. Descartes müsste aber zugestehen, dass Computer tatsächlich Universalinstrumente sind. Computer können mit wenigen »Organen« viele verschiedene Tätigkeiten ausführen. Obwohl er sich nun vorstellen kann, dass Programme sogar aus Erfahrung lernen können, indem sie Informationen auf ihr Arbeitsblatt schreiben (zum Beispiel, dass das Wort ›warm‹ eine Temperatur von mehr als 17° Celsius bedeutet), so würde ihm sicher nicht entgehen, dass in den meisten Fällen das Programm auf dem Programmblatt von einem Menschen geschrieben wurde. Selbst im Fall von ChatGPT wurde das Regelbuch zwar nicht von Menschen geschrieben, aber das Programm, das die statistischen Regeln automatisch aus den Daten erzeugt, schon. Ist die Maschine, so könnte Descartes fragen, somit wirklich »aus Erkenntnis tätig«?

Wer kontrolliert, wann die Maschine welches Programm ausführt? Wer ändert das Programm ab, falls es nicht das tut, was es soll? Am Ende ist das immer noch der Mensch. Aber Programme können sich durchaus untereinander kontrollieren und umprogrammieren. Das Betriebssystem Ihres Computers ist ein Programm, das andere Programme kontrolliert, indem es diese vom Arbeitsblatt, auf dem diese ›installiert‹ sind, zeitweise auf das Programmblatt kopiert und wieder löscht. So sorgt das Betriebssystem dafür, dass diese Programme ausgeführt und beendet werden. Da Programme nur aus Zeichen bestehen, die in irgendeinem Computerspeicher stehen, kann ein Programm diese Zeichen ändern und damit verändern, was die Programme tun. Dafür sind Computerviren ein gutes Beispiel. Das sind Programme, die andere Programme ändern und dazu bringen, den Virus weiterzutragen. Es ist sogar leicht möglich, dass ein Programm die Regeln

auf dem Programmblatt, die beschreiben, was es tun soll, selber ändert. So können Computerprogramme geschrieben werden, die selbstständig lernen.

Ein Masterkontrollprogramm, das alle Programme kontrollieren und umprogrammieren kann, ist also denkbar. Ein Programm, das sich selber umprogrammieren kann, so wie wir unser Verhalten dank unserer Vernunft verändern können, ist möglich. Science-Fiction-Filme und Romane sind daher voll von solchen sich selbst weiterentwickelnden KI-Programmen, die die Kontrolle über alle Maschinen übernehmen und so die Weltherrschaft an sich reißen. Einige KI-Forscherinnen und -Forscher machen sich deshalb große Sorgen. Zu Recht?