

Die Turing-Maschine

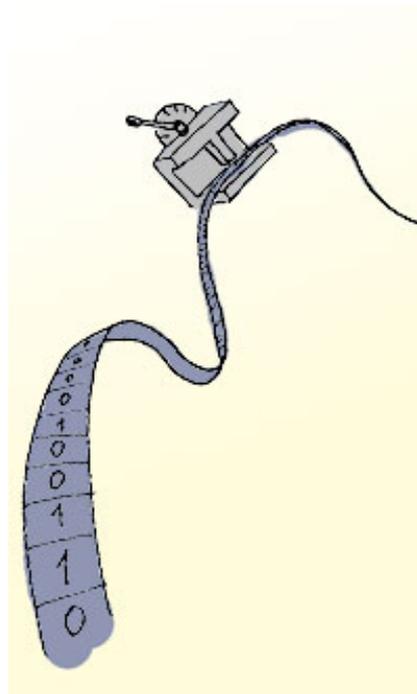
Rechnen mit Maschinen, die es gar nicht gibt

Turing-Maschinen werden nicht gebaut - das wäre ziemlich sinnlos. Sie sind allerdings ein praktisches Konzept um auf theoretischer Ebene über die Möglichkeiten von Rechenautomaten nachdenken zu können. Prinzipiell haben sie die selben Fähigkeiten wie jeder Computer, sie sind aber präzise und einfach zu beschreiben.

Schritt für Schritt, am unendlichen Band

Alan Turings Rechenmaschinen-Konzept ist simpel: Er stellte sich ein unendlich langes Band vor, das in einzelne Felder eingeteilt ist. Auf diesen Feldern können bestimmte Zahlenwerte gespeichert werden - zum Beispiel null und eins. Schritt für Schritt bewegt sich nun eine Maschine entlang dieses Bandes und liest sie die Zahl, die dort auf dem Band geschrieben steht. Die Maschine kann eine bestimmte Anzahl verschiedener Zustände annehmen - wie ein Hebel, der auf eine von verschiedenen möglichen Positionen eingestellt ist. Außerdem hat die Maschine eine Tabelle, die genau festlegt, welche Aktionen ausgeführt werden - abhängig von der Hebel-Stellung und der aktuellen Zahl am Band.

Nach den Regeln dieser Tabelle kann die Maschine ihren Zustand ändern, also den Hebel in eine andere Stellung bringen. Sie kann die Zahl auf dem Band ändern (oder auch gleichlassen) und schließlich kann sie sich entlang des Bands einen Schritt nach rechts oder nach links bewegen - oder stehenbleiben.



Eine Turing-Maschine auf einem endlosen Datenband

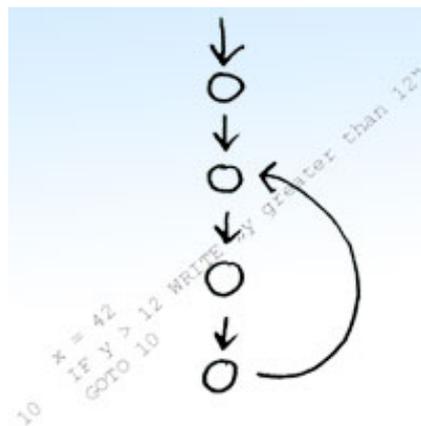
Von A nach B und wieder zurück

Zu Beginn ist das gesamte Band überall mit dem selben Zeichen beschrieben, zum Beispiel mit der Null. Man kann sich nun leicht eine ganz einfache Variante einer Turing-Maschine ausdenken: Nehmen wir an, sie kennt nur zwei innere Zustände (A

und B) und gehorcht ganz simplen Regeln: Wenn sich die Maschine im Zustand A befindet, dann ändert sie die Zahl, die am Band steht (von null auf eins oder umgekehrt), sie schaltet ihren inneren Zustand auf B und rückt entlang des Bandes eine Stelle nach rechts. Wenn sie im Zustand B ist, dann lässt sie die Zahl am Band gleich, ändert ihren Zustand auf A und rückt auch eine Stelle nach rechts. Man sieht sofort: Wenn wir diese einfache Maschine auf das Band loslassen, das voll mit lauter Null-Symbolen ist, wird die Turing-Maschine Schritt für Schritt immer weiter nach rechts rücken (etwas anderes haben wir ihr mit unseren Regeln ja nicht beigebracht) und sie wird auf dem Band immer abwechselnd „null“ und „eins“ hinterlassen, ohne jemals aufzuhören, weil sie in jedem Schritt den inneren Zustand zwischen A und B hin und her schaltet.

Alles, was man rechnen kann

Zugegeben: Das ist keine besonders sinnvolle Turing-Maschine. Aber wenn man will kann man Turing-Maschinen mit äußerst komplizierte Regelsystemen und einer großen Anzahl von inneren Zuständen definieren, die dann in der Lage wären, aufwändige Rechenprobleme zu lösen. Alan Turing konnte zeigen, dass eine solche Maschine prinzipiell jede mögliche Rechenaufgabe durchführen kann. Jedes Programm, das heute auf unseren Computern läuft kann auch von einer Turing-Maschine erledigt werden. Was eine "mögliche Rechenaufgaben" ist, kann man freilich nur schwer definieren. Turing war der Meinung, dass sich grundsätzlich jede Rechnung, die ein Mensch ausführen kann, auch von einer Turing-Maschine berechnen lässt - das ist die sogenannte „Church-Turing-These“.



Ein Programm kann zu einem früheren Punkt zurückkehren und letztlich immer im Kreis laufen.

Für theoretische Überlegungen in der Informatik ist das wichtig. Nachdem die Turing-Maschine so einfach aber doch so mächtig ist, kann man sie verwenden um zu erforschen, welche mathematischen Problemstellungen überhaupt lösbar sind. In der Schule lernt man eine Menge Rechenvorschriften - normalerweise kommt man am Ende zu einem Ergebnis, wenn man sich an diese Vorschriften hält. Es gibt aber auch Rechenvorschriften, die uns immer weiterrechnen lassen, für immer, ohne Ende. Gibt es eine Möglichkeit, schon vorher zu wissen, ob unsere Rechenvorschriften auf ein Ergebnis führen, oder ob sie uns in einer endlosen Rechen-Schleife gefangen halten werden? Alan Turing konnte diese Frage beantworten, indem er sie auf seine Turing-Maschinen umlegte. Er bewies: Ein Computerprogramm, das von jedem anderen Computerprogramm vorhersagt, ob es mit seiner Rechnung jemals an ein Ende kommen wird, kann es nicht geben.