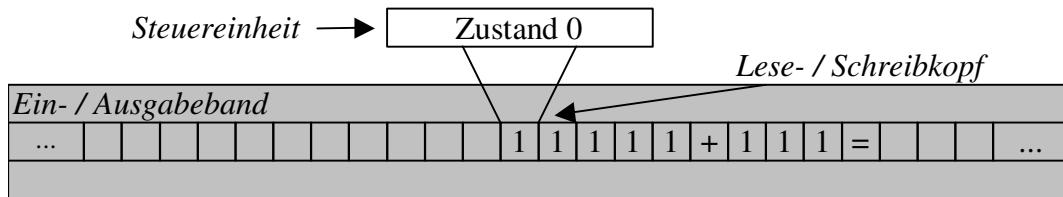


ARBEITSBLATT ZU TURINGMASCHINEN

Das Halteproblem: Kann man ein Programm schreiben, welches einen beliebigen Algorithmus dahingehend überprüft, ob er terminiert?

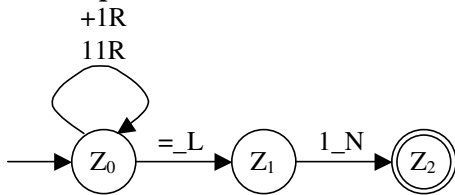
Das Entscheidungsproblem: Gibt es ein Verfahren, das für jede ausreichend formalisierte Aussage der Mathematik entscheidet, ob diese wahr oder falsch ist?
(nach Hilbert)

Erst im Jahr 1936 konnten sowohl Alonzo Church als auch Alan Turing beweisen, dass obiges Entscheidungsproblem nicht lösbar ist. Möglich wurde dieser Beweis erst durch die Definition einer Maschine, welche sich in der theoretischen Leistungsfähigkeit von einem heutigen Computer im Prinzip nicht unterscheidet. Auch eine Turingmaschine hat einen Speicher (Ein- / Ausgabeband), einen Prozessor (Steuereinheit) und einen Datenbus (Lese- / Schreibkopf):



Ein Programm, welches die Einsen auf dem Band addiert, könnte nun wie folgt dargestellt werden:

als Graph:



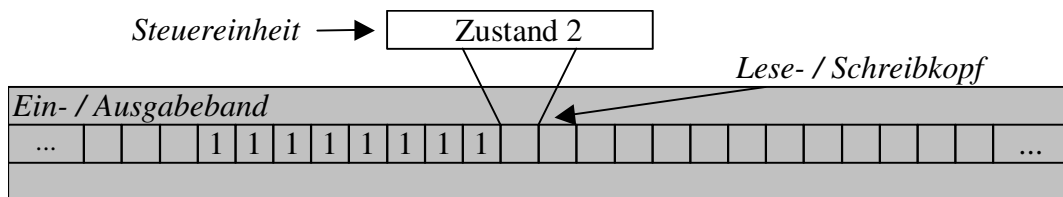
als Tafel:

Zustand \ Zeichen	_	1	+	=
Z ₀	---	1 R Z ₀	1 R Z ₀	_ L Z ₁
Z ₁	---	_ N Z ₂	---	---
Z ₂	_ N Z ₂	---	---	---

+1R bedeutet: Der Lesekopf liest ein +, überschreibt es mit 1 und bewegt den Kopf nach R.
Durch den Pfeil wird deutlich, dass die Steuereinheit im Zustand Z₀ bleibt.

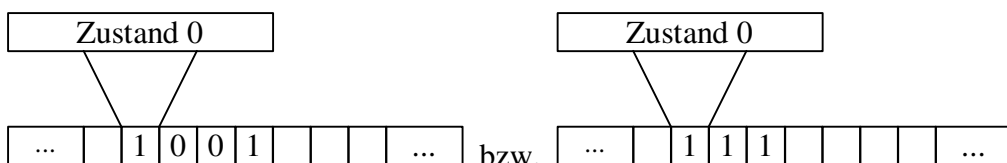
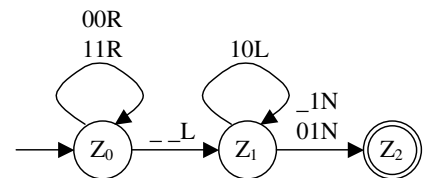
In der Tafel findet sich dieser Zustandsübergang hier

Das Programm ist beendet, wenn ein Endzustand erreicht ist. In unserem Fall sähe die Turingmaschine nach Abarbeitung des Programms wie folgt aus:



Aufgabe 1: Teste das Beispielprogramm mit Hilfe des Turing-Simulators.

Aufgabe 2: Gib zum rechts abgebildeten Programm die Tafel an. Führe das Programm mit den folgenden Bandbelegungen durch und überlege dir, was das Programm leistet.



Aufgabe 3: Erweitere das Programm aus Aufgabe 2 so, dass der Lesekopf am Ende des Programms wieder auf dem ersten Zeichen des Bandwortes steht.