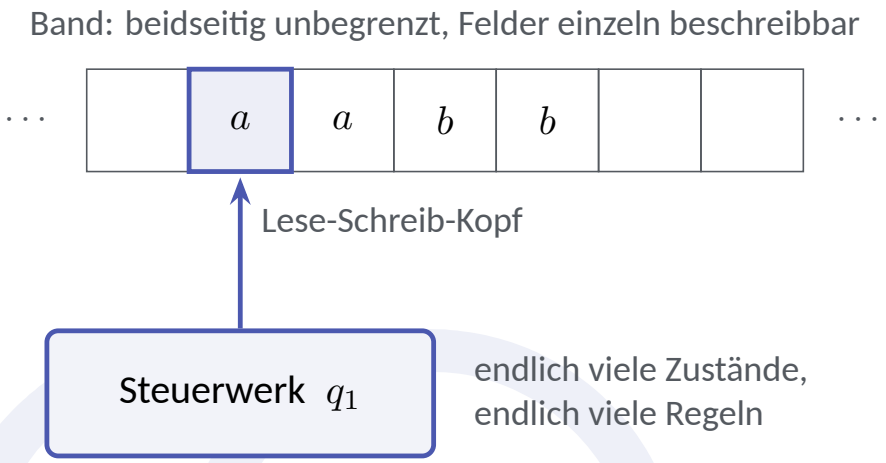
1.2 Die Turingmaschine

LK

> LK
>
> Auch dieser Abschnitt ist LK-Stoff (*Berechenbarkeit und ihre Grenzen*). Im Grundkurs genügt die Grundidee: Es gibt ein formales Modell dafür, was „berechenbar" heißt.

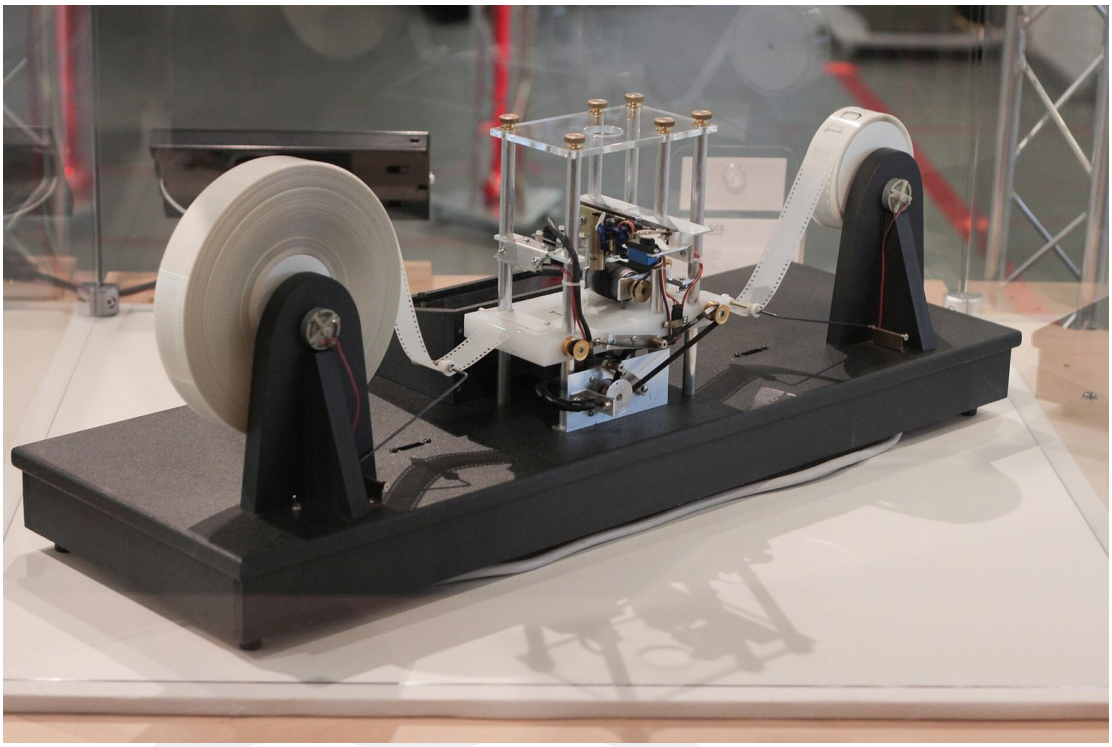
Die Maschine, die Turing beschrieb, ist absichtlich armselig. Sie besteht aus drei Teilen.



Aufbau einer Turingmaschine.

Gebaut hat man sie trotzdem — nicht, weil es nötig wäre, sondern weil es schön ist. Mike Davey stellte 2010 eine funktionierende Turingmaschine her: Links und rechts die Rollen mit dem Band, in der Mitte der Lese-Schreib-Kopf auf seinem Schlitten, darunter das Steuerwerk. Das Band ist tatsächlich unbegrenzt — es wird einfach nachgelegt.

Gymnasium Ottweiler



Turingmaschine, gebaut von Mike Davey (2010). Foto von 2012: Rocky Acosta, CC BY 3.0, via Wikimedia Commons.

Vergleichen Sie das Foto mit der Zeichnung darüber: Jedes Bauteil der Definition ist darin wiederzufinden — und nichts darüber hinaus.

Definition: Turingmaschine

Eine **Turingmaschine** besteht aus

- einem **Band**, das in Felder geteilt und nach beiden Seiten unbegrenzt ist. Jedes Feld trägt ein Zeichen oder ist leer (□).
- einem **Lese-Schreib-Kopf**, der auf genau einem Feld steht. Er kann das Zeichen dort lesen, es überschreiben und sich um ein Feld nach links oder rechts bewegen.
- einem **Steuerwerk**, das sich in einem von **endlich vielen Zuständen** befindet.

Das Verhalten legen **Regeln** fest. Eine Regel hat die Form

$$(\text{Zustand},\ \text{gelesenes Zeichen}) \longrightarrow (\text{zu schreibendes Zeichen},\ \text{Bewegung},\ \text{neuer Zustand})$$

mit der Bewegung L (links), R (rechts) oder N (stehen bleiben). Gibt es für die aktuelle Lage keine Regel, hält die Maschine an.

Merke

Endlich sind nur Zustände und Regeln. **Unbegrenzt ist allein das Band.** Genau darin liegt der Unterschied zum endlichen Automaten aus Kapitel 5: Dieser hat kein Band und muss sich alles im Zustand merken — und weil er nur endlich viele Zustände hat, kann er sich nur endlich viel merken.

1.2.1 Eine erste Maschine: unär addieren

Zahlen werden als Striche geschrieben: 3 ist 111. Auf dem Band steht eine Aufgabe wie 11+111, und am Ende soll dort das Ergebnis stehen.

Der Trick ist hübsch: Man ersetzt das + durch einen Strich — dann stehen alle Striche beieinander, aber es ist einer zu viel. Also läuft die Maschine ans Ende und löscht dort einen.

Zustand	liest	schreibt	Bew.	neu	Bedeutung
q_0	1	1	R	q_0	über die ersten Striche laufen
q_0	+	1	R	q_1	Pluszeichen zum Strich machen
q_1	1	1	R	q_1	weiter bis ans Ende
q_1	□	□	L	q_2	Ende erreicht, einen zurück
q_2	1	□	N	q_a	letzten Strich löschen, fertig

Beispiel: Ablauf für 11+111

Der Kopf steht jeweils auf dem eingerahmten Feld.

```
q0   [1] 1  +  1  1  1
q0    1 [1] +  1  1  1
q0    1  1 [+] 1  1  1
q1    1  1  1 [1] 1  1
q1    1  1  1  1 [1] 1
q1    1  1  1  1  1 [1]
q1    1  1  1  1  1  1 [□]
q2    1  1  1  1  1 [1] □
qa    1  1  1  1  1 [□] □
```

Acht Schritte, und auf dem Band stehen fünf Striche: $2 + 3 = 5$.

> In der Webfassung steht hier ein Bedienfeld zum Ausprobieren: **Turingmaschine**.

1.2.2 Eine Maschine, die ein Automat nicht sein kann

In Kapitel 5 wird gezeigt, dass kein endlicher Automat die Wörter $a^n b^n$ — also ab, aabb, aaabbb, … — erkennen kann: Er müsste beliebig weit zählen, und dazu reichen endlich viele Zustände nicht.

Eine Turingmaschine schafft es, weil sie sich auf dem Band Notizen machen kann. Ihre Idee ist die eines Kindes, das Äpfel und Birnen paarweise zuordnet: **immer ein a und ein b zusammen durchstreichen.** Bleibt am Ende nichts übrig, waren es gleich viele.

Gestrichene a werden zu X, gestrichene b zu Y.

Zustand	liest	schreibt	Bew.	neu	Bedeutung
q_0	a	X	R	q_1	ein a streichen
q_0	Y	Y	R	q_3	kein a mehr da — Rest prüfen
q_0	□	□	N	q_a	leeres Wort, annehmen
q_1	a	a	R	q_1	nach rechts zum ersten b
q_1	Y	Y	R	q_1	über Gestrichenes hinweg
q_1	b	Y	L	q_2	zugehöriges b streichen
q_2	a	a	L	q_2	zurück nach links
q_2	Y	Y	L	q_2	zurück nach links
q_2	X	X	R	q_0	am letzten X — nächste Runde
q_3	Y	Y	R	q_3	nur noch Y erlaubt
q_3	□	□	N	q_a	sauber aufgegangen, annehmen

Beispiel: Ablauf für aabb

```
q0  [a] a  b  b        q1   X  X  Y [b]
q1   X [a] b  b        q2   X  X [Y] Y
q1   X  a [b] b        q2   X [X] Y  Y
q2   X [a] Y  b        q0   X  X [Y] Y
q2  [X] a  Y  b        q3   X  X  Y [Y]
q0   X [a] Y  b        q3   X  X  Y  Y [□]
q1   X  X [Y] b        qa   angenommen
```

Dreizehn Schritte. Zweimal ist die Maschine hin- und hergelaufen — einmal für jedes Paar.

> In der Webfassung steht hier ein Bedienfeld zum Ausprobieren: **Turingmaschine**.

Probieren Sie im Bedienfeld auch `aab` und `abb` — die Maschine bleibt dann stehen, ohne q_a zu erreichen.

Stolperstein

Fehlt eine Regel, hält die Maschine einfach an, **ohne** anzunehmen. Bei `aab` etwa steht die Maschine irgendwann in q_1 vor einem $\square$ — dafür gibt es keine Zeile in der Tabelle. Das Wort ist damit abgelehnt. Ablehnen heißt also nicht „Nein sagen", sondern nur „nicht in q_a enden".

1.2.3 Die Church-Turing-These

Turing und Church hatten unabhängig voneinander zwei ganz verschiedene Begriffe von „berechenbar" entwickelt — Turing seine Maschine, Church den λ-Kalkül. Dann stellte sich heraus: Beide umfassen genau dieselben Probleme. Dasselbe gilt für jedes weitere Modell, das seither vorgeschlagen wurde, und für jede Programmiersprache, die je gebaut wurde.

Merke: Church-Turing-These

Alles, was überhaupt im intuitiven Sinn **berechenbar** ist, kann eine Turingmaschine berechnen.

Das ist **kein Satz, sondern eine These**, und sie lässt sich grundsätzlich nicht beweisen: Auf der einen Seite steht ein exakter Begriff, auf der anderen das Alltagswort „berechenbar". Zwischen beiden kann man keinen Beweis führen. Widerlegen ließe sie sich dagegen sofort — durch ein Verfahren, das offensichtlich rechnet und das keine Turingmaschine nachahmen kann. In neunzig Jahren hat niemand eines gefunden.

Merke

Die These hat eine unbequeme Folge, die diesen Kurs durchzieht: Zeigt man, dass eine Turingmaschine ein Problem nicht lösen kann, dann kann **kein Rechner** es lösen — kein heutiger, kein zukünftiger, egal wie schnell. Das ist die Grundlage von Abschnitt 4.4.

Arbeitsauftrag 1

1. Lassen Sie die Additionsmaschine auf `111+1` laufen und schreiben Sie das Protokoll wie im Beispiel auf.
2. Was tut die Additionsmaschine, wenn auf dem Band gar kein `+` steht? Begründen Sie mit der Tabelle.
3. Geben Sie eine Turingmaschine an, die ein Band aus Nullen und Einsen von links nach rechts durchläuft und dabei jede `0` durch eine `1` ersetzt und umgekehrt.
4. Lassen Sie die zweite Maschine auf `ab` und auf `abb` laufen. Bei welchem Schritt genau scheitert `abb`?
5. Warum braucht die zweite Maschine zwei verschiedene Streichzeichen `X` und `Y`? Was ginge schief, wenn beide `X` hießen?

Selbstcheck

Können Sie in einem Satz sagen, was eine Turingmaschine kann, was ein endlicher Automat nicht kann — und woran das liegt?