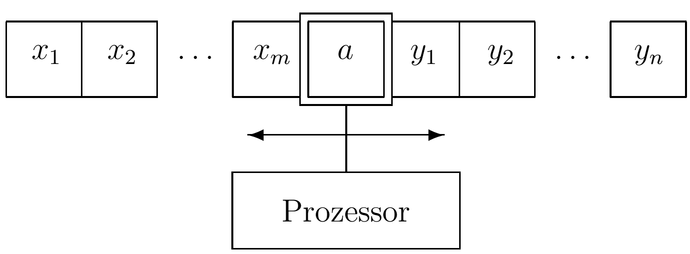
Turing-Maschinen

Das Konzept der Turing-Maschine wurde von Alan Turing in den 30er Jahren dieses Jahrhunderts eingeführt und stellt damit eines der ältesten Berechenbarkeitsmodelle dar. Die Idee war, den mechanischen Anteil des Rechnens mit Bleistift und Radiergummi auf Papier formal zu fassen.

Begriff und Arbeitsweise

Eine Turing-Maschine *TM* funktioniert so: Sie befindet sich zu jeder Zeit in einem Zustand, der aus einer vorgegebenen endlichen Menge S von Zuständen stammt, mit denen sich also endlich viele verschiedene Fälle darstellen lassen. Sie hat ein Arbeitsband, das aus einer linearen Kette von Zellen besteht, die von links nach rechts geordnet sind. Eine Zelle kann unbeschrieben oder mit einem Zeichen aus einem endlichen Alphabet A beschrieben sein. Zu jeder Zeit ist das Band endlich lang, kann aber unter bestimmten Umständen links und rechts durch unbeschriebene Zellen verlängert werden. Außerdem hat die Maschine einen Lese-Schreib-Kopf, der entweder am rechten Ende des Bandes oder auf einer der Zellen steht. Formal ist deshalb das Arbeitsband durch zwei Zeichenketten $u, v \in A'^*$ beschrieben, wobei u den Bandinhalt links vom Kopf und v den Bandinhalt rechts vom Kopf – einschließlich der aktuellen Zelle, falls der Kopf nicht ganz rechts steht – darstellt. A' umfasst A und enthält außerdem ein Sonderzeichen $\square$, das unbeschriebene Zellen repräsentiert. Zu Beginn befindet sich die Maschine in einem ausgezeichneten Anfangszustand, der Lese-Schreib-Kopf steht ganz links und alle Zellen sind beschrieben (d.h. $u = \lambda$ und $v \in A^*$). Die Maschine kann Arbeitsschritte vollziehen, die von ihrem "Programm" – der Zustandsüberführungsrelation d – abhängen. Diese ordnet jedem Zustand und jedem gelesenen Zeichen aus A' mögliche Folgezustände, mögliche zu schreibende Zeichen aus A' und mögliche Bewegungen des Kopfes zu. Als Bewegungen stehen zur Verfügung "l" für eine Zelle nach links, "r" für eine Zelle nach rechts und "n" für Nicht-Bewegen. Ein konkreter Schritt ändert dann den aktuellen Zustand sowie den Inhalt der Zelle unter dem Lese-Schreib-Kopf und führt eine Bewegung aus – und das alles gemäß der Zustandsüberführungsrelation. Solche Arbeitsschritte können beliebig wiederholt werden. Die Schrittfolge endet notwendigerweise, wenn die Zustandsüberführungsrelation keinen Folgezustand mehr vorsieht. Das tritt insbesondere ein, wenn der aktuelle Zustand ein Endzustand ist. Alles was in diesem Fall unter dem Kopf und rechts davon steht, wird als Ergebnis der Berechnung angesehen, wenn da keine Zelle unbeschrieben ist.



Diese Konzeption kann folgendermaßen formalisiert werden:

1. Eine *Turing-Maschine* ist ein System $TM = (S, A, d, s_0, F)$, wobei S eine endliche Menge von *Zuständen*, A ein endliches *Alphabet*, $s_0 \in S$ ein *Anfangszustand*, $F \subseteq S$ eine Menge von *Endzuständen* und d eine *Zustandsüberführungsrelation* ist, die jedem Zustand $s \in S$ und jedem Zeichen $a \in A' = A \cup \{\square\}$ eine Teilmenge $d(s, a) \subseteq S \times A' \times \{n, l, r\}$ zuordnet. Dabei ist $\square$ ein Sonderzeichen (d.h. $\square \notin A$), und n, l und r sind drei Steuerzeichen. Außerdem wird gefordert, dass $d(s', a)$ leer ist für alle $s' \in F$ und $a \in A'$.

2. *TM* ist *deterministisch*, falls $d(s, a)$ für jedes $s \in S$ und $a \in A'$ höchstens ein Element enthält.

3. Eine *Konfiguration* hat die Form usv mit $u, v \in A'^*$ und $s \in S$.

4. Eine *Anfangskonfiguration* hat die Form $\lambda s_0 w$ mit $w \in A^*$.

5. Eine *Endkonfiguration* hat die Form $us'v$ mit $u \in A'^*, s' \in F$ und $v \in A^*$.

6. *Folgekonfigurationen* entstehen für alle $s, s' \in S$, $u, v \in A'^*$ und $a, c \in A'$ wie folgt:

$$\begin{array}{lll}
\text{(i)} & usav \longmapsto us'bv, & \text{falls } (s', b, n) \in d(s, a) \\
\text{(ii)} & usav \longmapsto ubs'v, & \text{falls } (s', b, r) \in d(s, a) \\
\text{(iii)} & ucsav \longmapsto us'cbv & \multirow{2}{*}{$\left.\begin{array}{l}\\ \\ \end{array}\right\}$ falls $(s', b, l) \in d(s, a)$} \\
\text{(iv)} & \lambda sav \longmapsto s'\square bv & \\
\text{(v)} & us\lambda \longmapsto us\square &
\end{array}$$

7. Die Arbeitsweise der Turing-Maschine besteht in der beliebigen Iteration solcher Konfigurationsübergänge:

$$u_1 s_1 v_1 \longmapsto u_2 s_2 v_2 \longmapsto \cdots \longmapsto u_k s_k v_k,$$

wofür kurz auch $u_1 s_1 v_1 \overset{k-1}{\longmapsto} u_k s_k v_k$ geschrieben werden kann, wenn die Zwischenschritte nicht explizit gebraucht werden. Ist auch die Schrittzahl unwesentlich, kann $k - 1$ durch $*$ ersetzt werden.

8. Eine (partielle) Funktion $f\colon A^* \to A^*$ wird von einer Turing-Maschine *TM berechnet*, falls für alle $v, w \in A^*$ gilt:

$$f(w) = v \quad \text{gdw.} \quad \lambda s_0 w \overset{*}{\longmapsto} us'v \text{ für geeignete } u \in A'^* \text{ und } s' \in F.$$

In diesem Fall wird f auch mit f_{TM} bezeichnet.

Deterministische Turing-Maschinen

Während bei einer beliebigen Turing-Maschine eine Konfiguration mehrere Folgekonfigurationen besitzen kann, ist bei einer deterministischen Maschine immer höchstens ein Übergang möglich. Jede Anfangskonfiguration kann also in genau einer Weise durch Konfigurationsübergänge ausgerechnet werden, wobei die Übergänge entweder unendlich fortsetzbar sind und die Maschine nicht hält, oder aber die Folge in einer Konfiguration endet, zu der es keine Folgekonfiguration gibt. Ist diese eine Endkonfiguration, so ist die Berechnung erfolgreich. Sonst hält die Maschine ohne Ergebnis.

Ohne Beweis sei angemerkt, dass deterministische und nichtdeterministische Turing-Maschinen dieselbe Klasse von Funktionen berechnen.

Übersetzung von Turing-Maschinen in CE-S

Als ein Beispiel, wie verschiedene Berechnungsmodelle miteinander verglichen werden können, soll gezeigt werden, dass jede von einer Turing-Maschine berechnete Funktion auch in CE-S spezifizierbar ist. Dazu wird, was in der Informatik häufig vorkommt, ein Übersetzer gebaut. Der Übersetzer, der TRANSLATOR heißen soll, verwandelt jede Turing-Maschine TM in eine Spezifikation **spec**(TM), in der insbesondere die Funktion $f\colon A^* \to A^*$ deklariert ist. Es kann gezeigt werden, dass für die von TM berechnete Funktion f_{TM} und für $v, w \in A^*$ die Gleichheit $f_{TM}(w) = v$ genau dann gilt, wenn $f(w) = v$ im Sinne gleichwertiger Terme in **spec**(TM) gilt. TM und **spec**(TM) berechnen also dieselben Funktionen, was sich einprägsam als *Korrektheit* von TRANSLATOR ausdrücken lässt.

Die bei der Übersetzung verwendete Idee ist, für jeden Zustand $s \in S$ eine Funktion $f_s\colon A'^* \times A'^* \to A'^*$ einzuführen, so dass eine Konfiguration usv durch den Term $f_s(u, v)$ ausgedrückt wird und die Bildung von Folgekonfigurationen direkt als Gleichungen widergespiegelt werden kann. Weitere Gleichungen modellieren die Rolle von Anfangs- und Endkonfigurationen:

$$
\begin{array}{lll}
\multicolumn{3}{l}{\mathbf{spec}(TM)} \\
\text{opns:} & f\colon A^* \to A^* & \\
 & f_s\colon A'^* \times A'^* \to A'^* & \text{für alle } s \in S \\
\text{vars:} & u, v \in A'^*, w, x \in A^*, c \in A'^* & \\
\text{eqns:} & f(w) = f_{s_0}(\lambda, w) & \\
 & f_s(u, av) = f_{s'}(u, bv) & \text{mit } (s', b, n) \in d(s, a) \\
 & f_s(u, av) = f_{s'}(ub, v) & \text{mit } (s', b, r) \in d(s, a) \\
 & \left.\begin{array}{l} f_s(uc, av) = f_{s'}(u, cbv) \\ f_s(\lambda, av) = f_{s'}(\lambda, \Box bv) \end{array}\right\} & \text{mit } (s', b, l) \in d(s, a) \\
 & f_s(u, \lambda) = f_s(u, \Box) & \text{mit } s \in S \\
 & f_{s'}(u, x) = x & \text{mit } s' \in F
\end{array}
$$

Beachte, dass dabei a und b keine Variablen, sondern Konstanten sind und dass jeder Eintrag der Form $(s', b, n) \in d(s, a)$ oder $(s', b, r) \in d(s, a)$ in der Zustandsüberführungsrelation von TM zu einer Gleichung und jeder Eintrag der Form $(s', b, l) \in d(s, a)$ zu zwei Gleichungen führt. Insbesondere enthält **spec**(TM) also nur – wie es der Definition von CE-S entspricht – endlich viele Gleichungen, weil A und S endlich sind.

Insgesamt erhält man einen Übersetzer, der als Diagramm so aussieht:

$$
\xrightarrow{\quad TM \quad} \boxed{\text{TRANSLATOR}} \xrightarrow{\quad \mathbf{spec}(TM) \quad}
$$

Er erweist sich im folgenden Sinne als korrekt:

Theorem

Sei TM eine Turing-Maschine, die die Funktion $f_{TM}\colon A^* \to A^*$ berechnet. Sei **spec**(TM) die zugehörige CE-S-Spezifikation mit der Deklaration $f\colon A^* \to A^*$. Dann gilt für alle $v, w \in A^*$:

$$f_{TM}(w) = v \quad gdw. \quad f(w) = v \text{ in } \mathbf{spec}(TM).$$

Beweis: $f_{TM}(w) = v$ bedeutet nach Definition $\lambda s_0 w \overset{*}{\longmapsto} us'v$ für geeignete $u \in A'^*$ und $s' \in F$. Nach dem Lemma unten folgt daraus $f_{s_0}(\lambda, w) = f_{s'}(u, v)$ in *spectm*, was mit der ersten und letzten Gleichung – wie gewünscht – ergibt:

$$f(w) = f_{s_0}(\lambda, w) = f_{s'}(u, v) = v.$$

Betrachte umgekehrt $f(w) = v$ in *spectm*. Nach dem Lemma unten folgt $f(w) \overset{*}{\longrightarrow} v$, was genau dann gilt, wenn $f_{s_0}(\lambda, w) \overset{*}{\longrightarrow} f_{s'}(u, v)$ für geeignete $u \in A'^*$ und $s' \in F$, da auf $f(w)$ nur die erste Gleichung anwendbar ist und auf v nur die letzte Gleichung rückwärts. Nach dem Lemma folgt $\lambda s_0 w \overset{*}{\longmapsto} us'v$, was gerade $f_{TM}(w) = v$ ergibt.

Lemma

Sei TM eine Turing-Maschine und **spec**(TM) die zugehörige CE-S-Spezifikation.

1. Dann gilt für alle $s, s' \in S$ und $u, v, u', v' \in A'^*$:

$$usv \overset{*}{\longmapsto} u's'v' \quad gdw. \quad f_s(u, v) \overset{*}{\longrightarrow} f_{s'}(u', v')$$

2. Sei $v, w \in A^*$ mit $f(w) \overset{*}{\longleftrightarrow} v$. Dann gilt $f(w) \overset{*}{\longrightarrow} v$.

Beachte, dass der zweite Teil besagt, dass die Terme $f(w)$ und v in **spec**(TM) genau dann gleichwertig sind, wenn sich $f(w)$ durch Gleichungsanwendungen von links nach rechts in v überführen lässt, dass also in diesem Fall die Symmetrie keine Rolle spielt.

Die Beweise beider Teile des Lemmas, die hier nicht aufgeführt werden, lassen sich mit vollständiger Induktion über die Zahl von Konfigurationsübergängen bzw. Gleichungsanwendungen führen. Um das in Teil 2 zu bewerkstelligen, muß eine allgemeinere Behauptung gezeigt werden, nämlich:

Sei $\overline{v} \in A^*$. Sei $x = \overline{u}$ für $\overline{u} \in A^*$ oder $x = f_s(u, v)$ für $s \in S$ und $u, v \in A^*$ oder $x = f(w)$ für $w \in A^*$. Sei ferner $x \overset{*}{\longleftrightarrow} \overline{v}$. Dann gilt $x \overset{*}{\longrightarrow} \overline{v}$.