Vorlesungsmitschrift

Algorithmen und Berechenbarkeit

Vorlesung 12

Letztes Update: 2017/12/10 - 17:41 Uhr

Beispiele für Turingmaschinen

Beispiel 1

Gegeben sei die Sprache L_1, die aus allen Wörtern über `0` und `1` besteht, und deren Wörter auf `0` enden:

$$L = \{w0 \mid w \in \{0,1\}^*\}$$

Eine TM, die L_1 entscheidet, könnte folgendermaßen aussehen:

$$M_1 = (\{q_0, q_1, \bar{q}\}, \{0,1\}, \{0,1,\mathrm{B}\}, \mathrm{B}, q_1, \bar{q}, \delta)$$

Die Übergangsfunktionen sind dabei

δ	`0`	`1`	`B`
q_0	$\underline{\{q_0, 0, R\}}$	$\{q_1, 1, R\}$	$\{\bar{q}, 1, N\}$
q_1	$\{q_0, 0, R\}$	$\{q_1, 1, R\}$	$\{\bar{q}, 0, N\}$

Zu lesen ist zum Beispiel die unterstrichene Übergangsfunktion wie folgt: Man befindet sich in Zustand q_0 und ließt eine `0` ein, dann bleibt man in Zustand q_0, schreibt eine `0` an die Stelle, an der vorher schon eine `0` war, und bewegt den SLK nach rechts.

Oftmals ist es ratsam, dass man sich klarmacht, was die Zustände bedeuten:

→ q_0 bedeutet, man hat als letztes Zeichen eine `0` gelesen

→ q_1 bedeutet, man hat als letztes Zeichen keine `0` gelesen

Für das Wort $w_1 =$ `010101` ergibt sich der folgende Ablauf:

$$\begin{array}{llllllll} \Rightarrow & q_1 010101 & \vdash & 0q_0 10101 & \vdash & 01q_1 0101 & \vdash & 010q_0 101 \\ \vdash & 0101q_1 01 & \vdash & 01010q_0 1 & \vdash & 010101q_1 B & \vdash & 010101\bar{q}0 \end{array}$$

Da der letzte Zustand erreicht ist und w_1 mit einer `1` endet, schreibt die TM in der letzten Konfiguration eine `0`.

Beispiel 2

Gegeben sei die Sprache L_2:

$$L_2 = \{0^n 1^n \mid n \geq 1\}$$

Eine TM, die L_2 entscheidet, könnte folgendermaßen aussehen:

$$M_2 = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, \bar{q}\}, \{0, 1\}, \{0, 1, \mathrm{B}\}, \mathrm{B}, q_0, \bar{q}, \delta)$$

Die Übergangsfunktionen sind dabei

δ	`0`	`1`	`B`
q_0	$\{q_0, 0, R\}$	$\{q_1, 1, R\}$	$\{\bar{q}, 0, N\}$
q_1	$\{\bar{q}, 0, N\}$	$\{q_1, 1, R\}$	$\{q_2, B, L\}$
q_2	$\{\bar{q}, 0, N\}$	$\{q_3, B, L\}$	$\{\bar{q}, 0, N\}$
q_3	$\{q_3, 0, L\}$	$\{q_3, 1, L\}$	$\{q_4, B, R\}$
q_4	$\{q_5, B, R\}$	$\{\bar{q}, 0, N\}$	$\{\bar{q}, 0, N\}$
q_5	$\{q_6, 0, R\}$	$\{q_6, 1, R\}$	$\{\bar{q}, 1, N\}$
q_6	$\{q_6, 0, R\}$	$\{q_6, 1, R\}$	$\{q_2, B, L\}$

- $\rightarrow$ q_0 bedeutet, man hat bis jetzt keine `1` gelesen
- $\rightarrow$ q_1 bedeutet, man hat $\texttt{0}^i$ und danach mindestens eine `1` gelesen
- $\rightarrow$ q_2 bedeutet, der SLK steht auf dem letzten Zeichen (also nicht `B`)
- $\rightarrow$ q_3 bedeutet, der SLK wird nach ganz links bewegt
- $\rightarrow$ q_4 bedeutet, der SLK steht auf dem ersten Zeichen (also nicht `B`)
- $\rightarrow$ q_5 bedeutet, man überprüft, ob man schon fertig ist
- $\rightarrow$ q_6 bedeutet, der SLK wird nach ganz rechts bewegt

Die Entscheidung, ob ein Wort in der Sprache liegt, läuft in zwei Phasen ab:

1. In dieser Phase überprüft man, ob die Eingabe der Form $0^i 1^j$ mit $i \geq 0$, $j \geq 1$ entspricht
2. Zu Beginn dieser Phase steht der SLK auf dem letzten Zeichen der Eingabe. In dieser zweiten Phase überprüft man durch abwechselndes Löschen der jeweils äußeren Zeichen, ob $i = j$.

Beispielhaft wird die zweite Phase für das Wort $w_2 =$ `000111` aufgezeigt:

$$\begin{array}{llllllll}
\Rightarrow & 00011q_2 1 & \vdash & 0001q_3 1B & \vdash & \cdots & \vdash & q_3 B00011 \\
\vdash & Bq_4 00011 & \vdash & Bq_5 0011 & \vdash & 0q_5 011 & \vdash & \cdots \\
\vdash & 0011q_6 B & \vdash & 001q_2 1 & \vdash & \cdots\cdots & \vdash & Bq_5 BBB \\
\vdash & \bar{q}1 & & & & & &
\end{array}$$

Da der letzte Zustand erreicht ist und für w_2 gilt $i = j$, schreibt die TM in der letzten Konfiguration eine `1`.

Programmierung einer TM

Die *Programmierung* einer TM ist relativ mühsam, dennoch können alle Sprachkonstrukte aus *normalen Programmiersprachen* dargestellt werden:

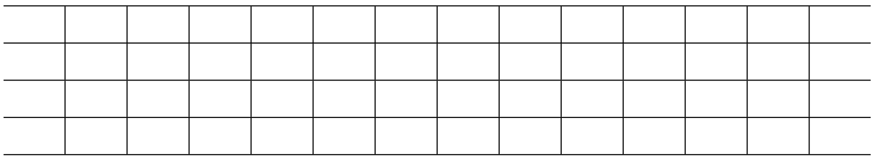
- Eine boolsche Variable im Programm kann folgendermaßen in den Zustandsraum kodiert werden

$$\begin{aligned} \text{Zustände alt:} \quad & Q \\ \text{Zustände neu:} \quad & Q \times \{0,1\} \end{aligned}$$

- Eine Variable x mit konstant vielen Zuständen $\{0, 1, 2, 3, \cdots, k\}$ kann ebenfalls in den Zustandsraum kodiert werden.

$$Q_{\text{neu}} = \{Q \times \{0, 1, 2, 3, \cdots, k\}\}$$

- Man kann mit einer einspurigen TM auch eine k-spuringe TM simulieren mit jeweils einem Bandalphabet für ein k-Tupel.



$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \Rightarrow \text{Neues Bandalphabt } |\Gamma_{\text{neu}}| = |\Gamma_{\text{alt}}|^n$$

  Das ist zum Beispiel dann nützlich, wenn zwei Binärzahlen miteinander addiert werden. Anstatt auf $\text{bin}(a)\#\text{bin}(b)$ zu operieren, kopiert man $\text{bin}(b)$ auf die zweite Spur und berechnet in der dritten Spur das Ergebnis.

- Konstant viele Variablen mit nicht konstantem Wertebereich $\rightarrow$ jeweils eine Spur pro Variable

- Unterprogramm und Rekursion $\rightarrow$ Reservierung einer Spur pro Prozeduraufruf

- $\cdots$

Mehrband TM

Eine Mehrband TM hat für jede Spur einen unabhängigen SLK. Die Zustandsübergänge haben die Form

$$\delta : (Q \backslash \{\bar{q}\} \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, N\}^k)$$

Das erste Band enthält die Ein- bzw. Ausgabe, alle anderen Bänder sind initial mit `B`s gefüllt.

Satz: Eine k-Band TM $\mathcal{M}$, die mit einer Rechenzeit von $f(n)$ und einem Platzverbrauch von $s(n)$ auskommt, kann von einer `1`-Band TM $\mathcal{M}'$ mit der Rechenzeit von $\mathcal{O}(f^2(n))$ und einem Platzbedarf von $\mathcal{O}(s(n))$ simuliert werden.

Beweisidee: Man simuliert eine k-Band TM durch eine $2k$-Band TM. Zum Beispiel wird die folgende 2-Band TM

...	B	0	1	2	1	2	1	B	B	...
...	B	B	1	1	3	2	B	B	B	...

durch

...	B	0	1	2	1	2	1	B	B	...
...	B	B	B	#	B	B	B	B	B	...
...	B	B	1	1	3	2	B	B	B	...
...	B	B	B	B	B	B	#	B	B	...

simuliert.
Für einen Rechenschritt der k-Band TM wird über den beschriebenen Teil der $2k$-Band TM gelaufen und die Zeichen unter den SLK der k-Bänder werden eingesammelt (und im Zustand gespeichert). In einem zweiten Durchlauf über die Bänder werden die Zeichen und Kopfpositionen upgedatet.

Anzahl beschriebener Zellen $\leq f(n)$

$\Rightarrow$ Ein Schritt der k-Band TM kostet $\mathcal{O}(f(n))$ Schritte auf der $2k$-Band TM

$\Rightarrow$ $\mathcal{O}(f^2(n))$