Script generated by TTT

Title: Nipkow: Theo (06.06.2019)

Date: Thu Jun 06 14:24:28 CEST 2019

Duration: 63:10 min

Pages: 61

Definition 5.9

Eine Turingmaschine (TM) ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ so dass

- Q ist eine endliche Menge von Zuständen.
- Σ ist eine endliche Menge, das Eingabealphabet.
- Γ ist eine endliche Menge, das Bandalphabet, mit $\Sigma \subset \Gamma$
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$ ist die Übergangsfunktion. δ darf partiell sein.
- $q_0 \in Q$ ist der Startzustand.
- $\Box \in \Gamma \setminus \Sigma$ ist das Leerzeichen.
- $F \subseteq Q$ ist die Menge der akzeptierenden oder Endzustände.

Annahme: $\delta(q, a)$ is nicht definiert für alle $q \in F$ und $a \in \Gamma$.

Eine nichtdeterministische Turingmaschine hat eine Übergangsfunktion $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R, N\})$.

Intuitiv bedeutet $\delta(q, a) = (q', b, D)$:

Definition 5.10

Eine Konfiguration einer Turingmaschine ist ein Tripel $(\alpha, q, \beta) \in \Gamma^* \times Q \times \Gamma^*$.

Dies modelliert

- Bandinhalt: $\dots\Box\alpha\beta\Box\dots$
- Zustand: q
- Kopf auf dem ersten Zeichen von $\beta\Box$

Die Startkonfiguration der Turingmaschine bei Eingabe $w \in \Sigma^*$ ist (ϵ, q_0, w).

Die Berechnung der TM M wird als Relation $\to_M$ auf Konfigurationen formalisiert. Falls $\delta(q, \mathit{first}(\beta)) = (q', c, D)$:

$$(\alpha, q, \beta) \to_M \begin{cases} (\alpha, q', c\ \mathit{rest}(\beta)) & \text{falls } D = N \\ (\alpha c, q', \mathit{rest}(\beta)) & \text{falls } D = R \\ (\mathit{butlast}(\alpha), q', \mathit{last}(\alpha)\ c\ \mathit{rest}(\beta)) & \text{falls } D = L \end{cases}$$

Die Berechnung der TM M wird als Relation $\to_M$ auf Konfigurationen formalisiert. Falls $\delta(q, \mathit{first}(\beta)) = (q', c, D)$:

$$(\alpha, q, \beta) \to_M \begin{cases} (\alpha, q', c\ \mathit{rest}(\beta)) & \text{falls } D = N \\ (\alpha c, q', \mathit{rest}(\beta)) & \text{falls } D = R \\ (\mathit{butlast}(\alpha), q', \mathit{last}(\alpha)\ c\ \mathit{rest}(\beta)) & \text{falls } D = L \end{cases}$$

wobei

$$\begin{array}{ll} \mathit{first}(aw) = a & \mathit{first}(\epsilon) = \Box \\ \mathit{rest}(aw) = w & \mathit{rest}(\epsilon) = \epsilon \\ \mathit{last}(wa) = a & \mathit{last}(\epsilon) = \Box \\ \mathit{butlast}(wa) = w & \mathit{butlast}(\epsilon) = \epsilon \end{array}$$

für $a \in \Gamma$ und $w \in \Gamma^*$.

Die Berechnung der TM M wird als Relation $\to_M$ auf Konfigurationen formalisiert. Falls $\delta(q, \mathit{first}(\beta)) = (q', c, D)$:

$$(\alpha, q, \beta) \to_M \begin{cases} (\alpha, q', c\ \mathit{rest}(\beta)) & \text{falls } D = N \\ (\alpha c, q', \mathit{rest}(\beta)) & \text{falls } D = R \\ (\mathit{butlast}(\alpha), q', \mathit{last}(\alpha)\ c\ \mathit{rest}(\beta)) & \text{falls } D = L \end{cases}$$

wobei

$$\begin{array}{ll} \mathit{first}(aw) = a & \mathit{first}(\epsilon) = \Box \\ \mathit{rest}(aw) = w & \mathit{rest}(\epsilon) = \epsilon \\ \mathit{last}(wa) = a & \mathit{last}(\epsilon) = \Box \\ \mathit{butlast}(wa) = w & \mathit{butlast}(\epsilon) = \epsilon \end{array}$$

für $a \in \Gamma$ und $w \in \Gamma^*$.

Falls M nichtdeterministisch ist: $\boxed{\delta(q, \mathit{first}(\beta)) \ni (q', c, D)}$

Beispiel 5.11 (Unär +1)

$$M = (\overset{Q}{\{q, f\}}, \overset{\Sigma}{\{1\}}, \overset{\Gamma}{\{1, \Box\}}, \delta, q, \Box, \{f\})$$

Beispiel 5.11 (Unär +1)

$$M = (\{q, f\}, \{1\}, \{1, \Box\}, \delta, q, \Box, \{f\})$$

$$\begin{aligned} \delta(q, \Box) &= (f, 1, N) \\ \delta(q, 1) &= (q, 1, R) \end{aligned}$$

Beispiel 5.11 (Unär +1)

$$M = (\{q, f\}, \{1\}, \{1, \Box\}, \delta, q, \Box, \{f\})$$

$$\begin{aligned} \delta(q, \Box) &= (f, 1, N) \\ \delta(q, 1) &= (q, 1, R) \end{aligned}$$

- Beispiellauf:

$$(\epsilon, q, 11) \rightarrow_M$$

Beispiel 5.11 (Unär +1)

$$M = (\{q, f\}, \{1\}, \{1, \Box\}, \delta, q, \Box, \{f\})$$

$$\begin{aligned} \delta(q, \Box) &= (f, 1, N) \\ \delta(q, 1) &= (q, 1, R) \end{aligned}$$

- Beispiellauf:

$$(\epsilon, q, 11) \rightarrow_M (1, q, 1) \rightarrow_M$$

Beispiel 5.11 (Unär +1)

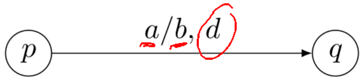
$$M = (\{q, f\}, \{1\}, \{1, \Box\}, \delta, q, \Box, \{f\})$$

$$\begin{aligned} \delta(q, \Box) &= (f, 1, N) \\ \delta(q, 1) &= (q, 1, R) \end{aligned}$$

- Beispiellauf:

$$(\epsilon, q, 11) \rightarrow_M (1, q, 1) \rightarrow_M (11, q, \epsilon) \rightarrow_M (11, f, 1)$$

Eine graphische Notation für $\delta(p, a) = (q, b, d)$:



Beispiel 5.12 (Binär $+1$)

ZB $1011 \mapsto 1100$

Beispiel 5.12 (Binär $+1$)

ZB $1011 \mapsto 1100$

$$M = (\{q_0, q_1, q_2, q_f\}, \{0, 1\}, \{0, 1, \Box\}, \delta, q_0, \Box, \{q_f\})$$

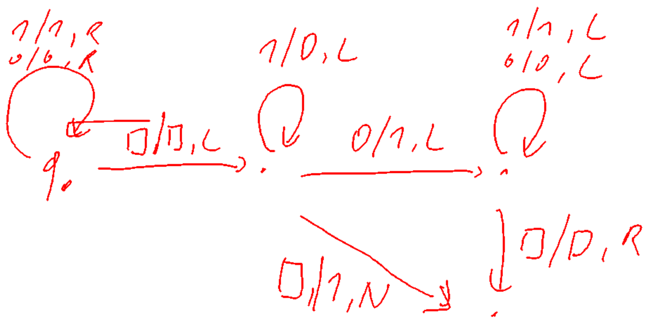
Beispiel 5.12 (Binär $+1$)

ZB $1011 \mapsto 1100$

$$M = (\{q_0, q_1, q_2, q_f\}, \{0, 1\}, \{0, 1, \Box\}, \delta, q_0, \Box, \{q_f\})$$

wobei

$$\begin{array}{llllll}
\delta(q_0, 0) & = (q_0, 0, R) & \delta(q_1, 1) & = (q_1, 0, L) & \delta(q_2, 0) & = (q_2, 0, L) \\
\delta(q_0, 1) & = (q_0, 1, R) & \delta(q_1, 0) & = (q_2, 1, L) & \delta(q_2, 1) & = (q_2, 1, L) \\
\delta(q_0, \Box) & = (q_1, \Box, L) & \delta(q_1, \Box) & = (q_f, 1, N) & \delta(q_2, \Box) & = (q_f, \Box, R)
\end{array}$$



Beispiel 5.12 (Binär +1)

ZB $1011 \mapsto 1100$

$$M = (\{q_0, q_1, q_2, q_f\}, \{0, 1\}, \{0, 1, \Box\}, \delta, q_0, \Box, \{q_f\})$$

wobei

$$\begin{array}{llllll}
\delta(q_0, 0) &= (q_0, 0, R) & \delta(q_1, 1) &= (q_1, 0, L) & \delta(q_2, 0) &= (q_2, 0, L) \\
\delta(q_0, 1) &= (q_0, 1, R) & \delta(q_1, 0) &= (q_2, 1, L) & \delta(q_2, 1) &= (q_2, 1, L) \\
\delta(q_0, \Box) &= (q_1, \Box, L) & \delta(q_1, \Box) &= (q_f, 1, N) & \delta(q_2, \Box) &= (q_f, \Box, R)
\end{array}$$

Beispiellauf:

$$(\epsilon, q_0, 101) \rightarrow_M (1, q_0, 01) \rightarrow_M$$

Beispiel 5.12 (Binär +1)

ZB $1011 \mapsto 1100$

$$M = (\{q_0, q_1, q_2, q_f\}, \{0, 1\}, \{0, 1, \Box\}, \delta, q_0, \Box, \{q_f\})$$

wobei

$$\begin{array}{llllll}
\delta(q_0, 0) &= (q_0, 0, R) & \delta(q_1, 1) &= (q_1, 0, L) & \delta(q_2, 0) &= (q_2, 0, L) \\
\delta(q_0, 1) &= (q_0, 1, R) & \delta(q_1, 0) &= (q_2, 1, L) & \delta(q_2, 1) &= (q_2, 1, L) \\
\delta(q_0, \Box) &= (q_1, \Box, L) & \delta(q_1, \Box) &= (q_f, 1, N) & \delta(q_2, \Box) &= (q_f, \Box, R)
\end{array}$$

Beispiellauf:

$$(\epsilon, q_0, 101) \rightarrow_M (1, q_0, 01) \rightarrow_M (10, q_0, 1) \rightarrow_M (101, q_0, \epsilon) \rightarrow_M$$
$$(10, q_1, 1\Box) \rightarrow_M (1, q_1, 00\Box) \rightarrow_M$$

Beispiel 5.12 (Binär +1)

ZB $1011 \mapsto 1100$

$$M = (\{q_0, q_1, q_2, q_f\}, \{0, 1\}, \{0, 1, \Box\}, \delta, q_0, \Box, \{q_f\})$$

wobei

$$\begin{array}{llllll}
\delta(q_0, 0) &= (q_0, 0, R) & \delta(q_1, 1) &= (q_1, 0, L) & \delta(q_2, 0) &= (q_2, 0, L) \\
\delta(q_0, 1) &= (q_0, 1, R) & \delta(q_1, 0) &= (q_2, 1, L) & \delta(q_2, 1) &= (q_2, 1, L) \\
\delta(q_0, \Box) &= (q_1, \Box, L) & \delta(q_1, \Box) &= (q_f, 1, N) & \delta(q_2, \Box) &= (q_f, \Box, R)
\end{array}$$

Beispiellauf:

$$(\epsilon, q_0, 101) \rightarrow_M (1, q_0, 01) \rightarrow_M (10, q_0, 1) \rightarrow_M (101, q_0, \epsilon) \rightarrow_M$$
$$(10, q_1, 1\Box) \rightarrow_M (1, q_1, 00\Box) \rightarrow_M$$
$$(\epsilon, q_2, 110\Box) \rightarrow_M$$

Beispiel 5.12 (Binär +1)

ZB $1011 \mapsto 1100$

$$M = (\{q_0, q_1, q_2, q_f\}, \{0,1\}, \{0,1,\Box\}, \delta, q_0, \Box, \{q_f\})$$

wobei

$$\begin{array}{lll}
\delta(q_0,0) = (q_0,0,R) & \delta(q_1,1) = (q_1,0,L) & \delta(q_2,0) = (q_2,0,L) \\
\delta(q_0,1) = (q_0,1,R) & \delta(q_1,0) = (q_2,1,L) & \delta(q_2,1) = (q_2,1,L) \\
\delta(q_0,\Box) = (q_1,\Box,L) & \delta(q_1,\Box) = (q_f,1,N) & \delta(q_2,\Box) = (q_f,\Box,R)
\end{array}$$

Beispiellauf:

$$\begin{array}{l}
(\epsilon, q_0, 101) \to_M (1, q_0, 01) \to_M (10, q_0, 1) \to_M (101, q_0, \epsilon) \to_M \\
(10, q_1, 1\Box) \to_M (1, q_1, 00\Box) \to_M \\
(\epsilon, q_2, 110\Box) \to_M (\epsilon, q_2, \Box 110\Box) \to_M
\end{array}$$

Definition 5.13

Eine Turingmaschine M akzeptiert die Sprache

$$L(M) = \{w \in \Sigma^* \mid \exists q \in F, \alpha, \beta \in \Gamma^*.\ (\epsilon, q_0, w) \to_M^* (\alpha, q, \beta)\}$$

Eine Funktion $f : \Sigma^* \to \Sigma^*$ heißt Turing-berechenbar gdw es eine Turingmaschine M gibt, so dass für alle $u, v \in \Sigma^*$ gilt

$$f(u) = v \quad \Leftrightarrow \quad \exists r \in F.\ (\epsilon, q_0, u) \to_M^* (\Box \ldots \Box, r, v\Box \ldots \Box)$$

Ausgabe

Definition 5.13

Eine Turingmaschine M akzeptiert die Sprache

$$L(M) = \{w \in \Sigma^* \mid \exists q \in F, \alpha, \beta \in \Gamma^*.\ (\epsilon, q_0, w) \to_M^* (\alpha, q, \beta)\}$$

Eine Funktion $f : \Sigma^* \to \Sigma^*$ heißt Turing-berechenbar gdw es eine Turingmaschine M gibt, so dass für alle $u, v \in \Sigma^*$ gilt

$$f(u) = v \quad \Leftrightarrow \quad \exists r \in F.\ (\epsilon, q_0, u) \to_M^* (\Box \ldots \Box, r, v\Box \ldots \Box)$$

Eine Funktion $f : \mathbb{N}^k \to \mathbb{N}$ heißt Turing-berechenbar gdw es eine Turingmaschine M gibt, so dass für alle $n_1, \ldots n_k, m \in \mathbb{N}$ gilt

Trennzeichen

$$\begin{array}{l}
f(n_1, \ldots, n_k) = m \quad \Leftrightarrow \\
\exists r \in F.\ (\epsilon, q_0, bin(n_1)\#bin(n_2)\# \ldots \#bin(n_k)) \\
\qquad \to_M^* (\Box \ldots \Box, r, bin(m)\Box \ldots \Box)
\end{array}$$

wobei $bin(n)$ die Binärdarstellung der Zahl n ist.

Zum Halten/Terminieren von TM

Zum Halten/Terminieren von TM

Eine TM hält wenn sie eine Konfiguration $(\alpha, q, a\beta)$ erreicht und $\delta(q, a)$ nicht definiert oder (bei einer nichtdetermistischen TM) $\delta(q, a) = \emptyset$.

Nach Annahme hält eine TM immer, wenn sie einen Endzustand erreicht. Damit ist die von einer TM berechnete Funktion wohldefiniert.

Zum Halten/Terminieren von TM

$\to^* (\ u\ , v', v' D\ )$

Zum Halten/Terminieren von TM

Intuitiv bedeutet $\delta(q, a) = (q', b, D)$:

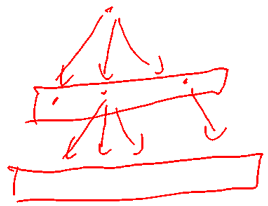
- Wenn sich M im Zustand q befindet,
- und auf dem Band a liest,

Satz 5.14

Zu jeder nichtdeterministischen TM N gibt es eine deterministische TM M mit $L(N) = L(M)$.

Beweis:

M durchsucht den Baum der Berechnungen von N in *Breitensuche*, beginnend mit der Startkonfiguration (ϵ, q_0, w), eine Ebene nach der anderen.



Satz 5.14

Zu jeder nichtdeterministischen TM N gibt es eine deterministische TM M mit $L(N) = L(M)$.

Beweis:

M durchsucht den Baum der Berechnungen von N in *Breitensuche*, beginnend mit der Startkonfiguration (ϵ, q_0, w), eine Ebene nach der anderen.
Gibt es in dem Baum (auf Ebene n) eine Konfigurationen mit Endzustand, so wird diese (nach Zeit $O(c^n)$) gefunden. □

Satz 5.15

Die von Turingmaschinen akzeptierten Sprachen sind genau die Typ-0-Sprachen der Chomsky Hierarchie.

Satz 5.15

Die von Turingmaschinen akzeptierten Sprachen sind genau die Typ-0-Sprachen der Chomsky Hierarchie.

Beweis:

Wir beschreiben nur die Beweisidee. (Mehr Details: [Schöning]).

Satz 5.15

Die von Turingmaschinen akzeptierten Sprachen sind genau die Typ-0-Sprachen der Chomsky Hierarchie.

Beweis:

Wir beschreiben nur die Beweisidee. (Mehr Details: [Schöning]).

„$\Longrightarrow$“: Grammatikregeln können direkt die Rechenregeln einer TM simulieren.

$(\alpha, q, \beta) \to_M (\alpha', q', \beta')$

Satz 5.15

Die von Turingmaschinen akzeptierten Sprachen sind genau die Typ-0-Sprachen der Chomsky Hierarchie.

Beweis:

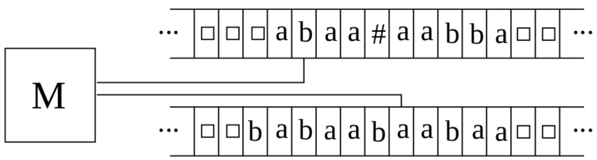
Wir beschreiben nur die Beweisidee. (Mehr Details: [Schöning]).

„$\Longrightarrow$“: Grammatikregeln können direkt die Rechenregeln einer TM simulieren.

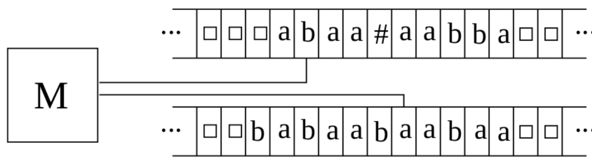
„$\Longleftarrow$“: Die (nichtdeterministische) TM versucht von ihrer Eingabe aus das Startsymbol der Grammatik zu erreichen, indem sie die Produktionen der Grammatik von rechts nach links anwendet, (nichtdeterministisch) an jeder möglichen Stelle. $\square$

$\alpha \to_P \beta$

Eine beliebte Modellvariante ist die k-Band-Turingmaschine:



Eine beliebte Modellvariante ist die k-Band-Turingmaschine:



Die k Köpfe sind völlig unabhängig voneinander:

$$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, N\}^k$$

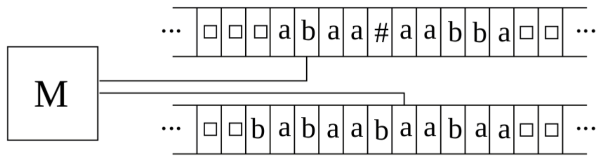
Satz 5.16

Jede k-Band-Turingmaschine kann effektiv durch eine 1-Band-TM simuliert werden.

Satz 5.16

Jede k-Band-Turingmaschine kann effektiv durch eine 1-Band-TM simuliert werden.

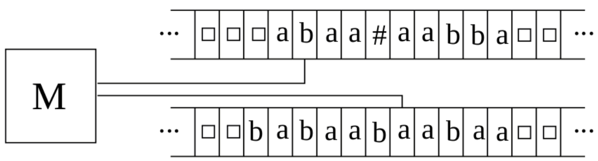
Beweisidee: Aus



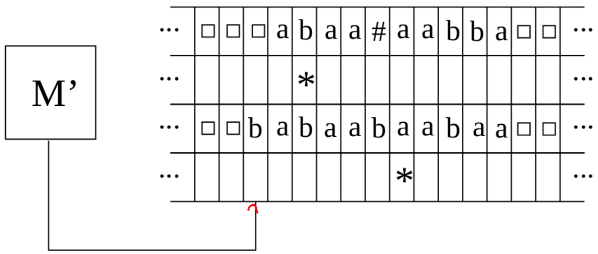
Satz 5.16

Jede k-Band-Turingmaschine kann effektiv durch eine 1-Band-TM simuliert werden.

Beweisidee: Aus



wird



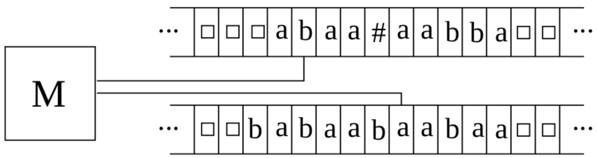
Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$

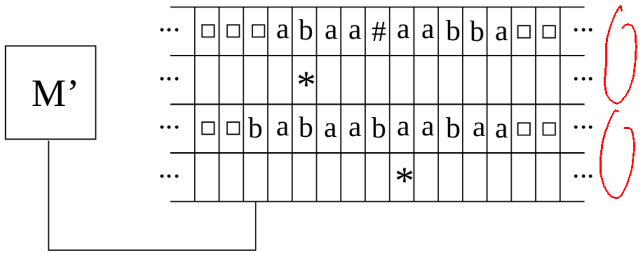
Satz 5.16

Jede k-Band-Turingmaschine kann effektiv durch eine 1-Band-TM simuliert werden.

Beweisidee: Aus



wird



Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'
  - startet mit Kopf links von allen $\star$.
  - geht nach rechts bis alle $\star$ überschritten sind, und merkt sich dabei (in Q') die Zeichen über jedem $\star$.

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'
  - startet mit Kopf links von allen $\star$.
  - geht nach rechts bis alle $\star$ überschritten sind, und merkt sich dabei (in Q') die Zeichen über jedem $\star$.
  - hat jetzt alle Information, um δ_M anzuwenden.

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'
  - startet mit Kopf links von allen $\star$.
  - geht nach rechts bis alle $\star$ überschritten sind, und merkt sich dabei (in Q') die Zeichen über jedem $\star$.
  - hat jetzt alle Information, um δ_M anzuwenden.
  - geht nach links über alle $\star$ hinweg und führt dabei δ_M aus.

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'
  - startet mit Kopf links von allen $\star$.
  - geht nach rechts bis alle $\star$ überschritten sind, und merkt sich dabei (in Q') die Zeichen über jedem $\star$.
  - hat jetzt alle Information, um δ_M anzuwenden.
  - geht nach links über alle $\star$ hinweg und führt dabei δ_M aus.

Beobachtung:

n Schritte von M lassens sich durch
$O(n^2)$ Schritte von M' simulieren.

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'
  - startet mit Kopf links von allen $\star$.
  - geht nach rechts bis alle $\star$ überschritten sind, und merkt sich dabei (in Q') die Zeichen über jedem $\star$.
  - hat jetzt alle Information, um δ_M anzuwenden.
  - geht nach links über alle $\star$ hinweg und führt dabei δ_M aus.

Beobachtung:

n Schritte von M lassens sich durch
$O(n^2)$ Schritte von M' simulieren.

Denn nach n Schritten von M trennen $\leq 2n$ Felder linkesten und rechtesten Kopf.

Beweisskizze:

- $\Gamma' := (\Gamma \times \{\star, \square\})^k$
- M' simuliert einen M-Schritt durch mehrere Schritte: M'
  - startet mit Kopf links von allen $\star$.
  - geht nach rechts bis alle $\star$ überschritten sind, und merkt sich dabei (in Q') die Zeichen über jedem $\star$.
  - hat jetzt alle Information, um δ_M anzuwenden.
  - geht nach links über alle $\star$ hinweg und führt dabei δ_M aus.

Beobachtung:

n Schritte von M lassens sich durch
$O(n^2)$ Schritte von M' simulieren.

Denn nach n Schritten von M trennen $\leq 2n$ Felder linkesten und rechtesten Kopf. Obige Simulation eines M-Schritts braucht daher $O(n)$ M'-Schritte. Simulation von n Schritten: $O(n^2)$ Schritte.

5.3 Programmieren mit Turingmaschinen

Die folgenden Basismaschinen sind leicht programmierbar:

- Band i := Band $i + 1$

5.3 Programmieren mit Turingmaschinen

Die folgenden Basismaschinen sind leicht programmierbar:

- Band i := Band $i + 1$
- Band i := Band $i - 1$

Seien $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, q_i, \square, F_i)$, $i = 1, 2$.
Die sequentielle Komposition (Hintereinanderschaltung) von M_1 und M_2 bezeichnen wir mit

$$\longrightarrow M_1 \longrightarrow M_2 \longrightarrow$$

Seien $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, q_i, \Box, F_i)$, $i = 1, 2$.
Die sequentielle Komposition (Hintereinanderschaltung) von M_1 und M_2 bezeichnen wir mit

$$\longrightarrow M_1 \longrightarrow M_2 \longrightarrow$$

Seien $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, q_i, \Box, F_i)$, $i = 1, 2$.
Die sequentielle Komposition (Hintereinanderschaltung) von M_1 und M_2 bezeichnen wir mit

$$\longrightarrow M_1 \longrightarrow M_2 \longrightarrow$$

Sie ist wie folgt definiert:

$$M := (Q_1 \cup Q_2, \Sigma, \Gamma_1 \cup \Gamma_2, \delta, q_1, \Box, F_2)$$

Seien $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, q_i, \Box, F_i)$, $i = 1, 2$.
Die sequentielle Komposition (Hintereinanderschaltung) von M_1 und M_2 bezeichnen wir mit

$$\longrightarrow M_1 \longrightarrow M_2 \longrightarrow$$

Sie ist wie folgt definiert:

$$M := (Q_1 \cup Q_2, \Sigma, \Gamma_1 \cup \Gamma_2, \delta, q_1, \Box, F_2)$$

wobei (oE) $Q_1 \cap Q_2 = \emptyset$ und

$$\delta := \delta_1 \cup \delta_2 \cup$$

Seien $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, q_i, \Box, F_i)$, $i = 1, 2$.
Die sequentielle Komposition (Hintereinanderschaltung) von M_1 und M_2 bezeichnen wir mit

$$\longrightarrow M_1 \longrightarrow M_2 \longrightarrow$$

Sie ist wie folgt definiert:

$$M := (Q_1 \cup Q_2, \Sigma, \Gamma_1 \cup \Gamma_2, \delta, q_1, \Box, F_2)$$

wobei (oE) $Q_1 \cap Q_2 = \emptyset$ und

$$\delta := \delta_1 \cup \delta_2 \cup \{(f_1, a) \mapsto (q_2, a, N) \mid f_1 \in F_1, a \in \Gamma_1\}$$

Sind f_1 und f_2 Endzustände von M so bezeichnet

$$\begin{array}{llll} \longrightarrow & M & \xrightarrow{f_1} & M_1 \quad \longrightarrow \\ & \downarrow f_2 & & \\ & M_2 & & \\ & \downarrow & & \end{array}$$

eine Fallunterscheidung, dh eine TM, die vom Endzustand f_1 von M nach M_1 übergeht, und von f_2 aus nach M_2.

Sind f_1 und f_2 Endzustände von M so bezeichnet

$$\begin{array}{llll} \longrightarrow & M & \xrightarrow{f_1} & M_1 \quad \longrightarrow \\ & \downarrow f_2 & & \\ & M_2 & & \\ & \downarrow & & \end{array}$$

eine Fallunterscheidung, dh eine TM, die vom Endzustand f_1 von M nach M_1 übergeht, und von f_2 aus nach M_2.

Die folgende TM nennen wir „Band=0?“ (bzw „Band $i = 0$?“):

$$\delta(q_0, 0) = (q_0, 0, R)$$

Sind f_1 und f_2 Endzustände von M so bezeichnet

$$\begin{array}{llll} \longrightarrow & M & \xrightarrow{f_1} & M_1 \quad \longrightarrow \\ & \downarrow f_2 & & \\ & M_2 & & \\ & \downarrow & & \end{array}$$

eine Fallunterscheidung, dh eine TM, die vom Endzustand f_1 von M nach M_1 übergeht, und von f_2 aus nach M_2.

Die folgende TM nennen wir „Band=0?“ (bzw „Band $i = 0$?“):

$$\delta(q_0, 0) = (q_0, 0, R)$$

Sind f_1 und f_2 Endzustände von M so bezeichnet

$$\begin{array}{llll} \longrightarrow & M & \xrightarrow{f_1} & M_1 \quad \longrightarrow \\ & \downarrow f_2 & & \\ & M_2 & & \\ & \downarrow & & \end{array}$$

$\longrightarrow$ Band = 0? $\xrightarrow{ja}$ M_1
$\downarrow$ nein
M_2

eine Fallunterscheidung, dh eine TM, die vom Endzustand f_1 von M nach M_1 übergeht, und von f_2 aus nach M_2.

Die folgende TM nennen wir „Band=0?“ (bzw „Band $i = 0$?“):

$$\delta(q_0, 0) = (q_0, 0, R)$$
$$\delta(q_0, \Box) = (ja, \Box, L)$$

Sind f_1 und f_2 Endzustände von M so bezeichnet

$$\begin{array}{llll} \longrightarrow & M & \xrightarrow{f_1} & M_1 \longrightarrow \\ & \downarrow f_2 & & \\ & M_2 & & \\ & \downarrow & & \end{array}$$

eine Fallunterscheidung, dh eine TM, die vom Endzustand f_1 von M nach M_1 übergeht, und von f_2 aus nach M_2.

Die folgende TM nennen wir „Band=0?“ (bzw „Band $i = 0$?“):

$$\begin{array}{rcll} \delta(q_0, 0) & = & (q_0, 0, R) & \\ \delta(q_0, \Box) & = & (ja, \Box, L) & \\ \delta(q_0, a) & = & (nein, a, N) & \text{für } a \neq 0, \Box \end{array}$$

wobei ja und $nein$ Endzustände sind.

Analog zur Fallunterscheidung kann man auch eine TM für eine Schleife konstruieren

$$\begin{array}{ll} \longrightarrow & \text{Band } i = 0? \xrightarrow{ja} \\ & \uparrow \downarrow nein \\ & M \end{array}$$

die sich wie `while` Band $i \neq 0$ do M verhält.

Moral: Mit TM kann man imperativ programmieren:

`:=`

`;`

`if`

`while`