

Theoretische Informatik II

Vorlesungsmitschrift vom 01.12.2000

Skript zur Vorlesung vom 01.12.2000

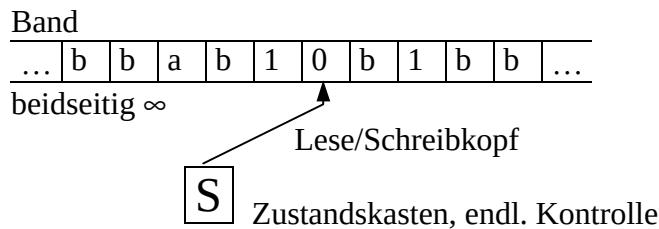
Inhalt:

<u>6. AUFGÄHNBARE SPRACHEN - CHOMSKY 0 SPRACHEN</u>	3
<u>6.1. TURINGMASCHINEN</u>	3
Arbeitsweise von Turingmaschinen	3
Definition A	4
Definition B	4
Definition C	4
Definition D	5
Definition E	6
Beispiele	6
Varianten von Turingmaschinen	7
<u>6.2. REGISTERMASCHINEN</u>	8
Unterschied von Turingmaschinen und Registermaschinen	8
Definition F	8
Definition G	9
Definition H	9
Beispiele	10
<u>Anhang</u>	11
Definition A (Kapitel 5)	11
Literaturverzeichnis	11

6. Aufzählbare Sprachen - Chomsky 0 Sprachen

- Grammatiken ohne Einschränkungen
[siehe: Definition A, Kapitel 5]
- hier: weniger interessant und praktisch
relevant die Sprachklassen Chomsky 0 ($\rightarrow$ Wertproblem nicht lösbar)
- hier: interessant nur Maschinenmodelle, da sie den verwendeten Maschinen in ihrer Leistungsfähigkeit entsprechen.
 $\rightarrow$ Leistungsfähigkeit unserer Computer, Bauelemente, Programmiersprachen
Programmierbarkeit ist theoretisch analysierbar

6.1. Turingmaschinen (A. M. Turing, 1936, *1912-†1954)



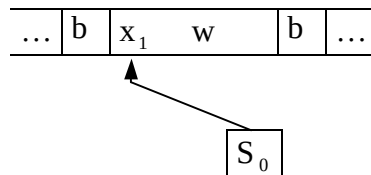
Bandalphabet Γ endlich

$b \in \Gamma$ Blanksymbol

endlich viele Felder $\neq b$ (zu jedem Zeitpunkt)

Arbeitsweise:

- a) Initialisierung: $w \in X^*$, $w = x_1 \dots x_n$, $x_i \in X \subset \Gamma$

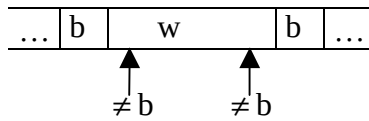


- b) Schritt ausführen:

(aktueller Zustand, Zeichen unter LSK) $\xrightarrow{\delta}$
(neuer Zustand, neues Zeichen unter LSK, LSK-Bewegung (l, r, h))

$$\delta(s, a) = (s', a', P); s', s \in S; a, a' \in \Gamma; P \in \{L, R, h\}$$

c) Berechnungsende: Ausgabe:



Definition A: Ein 6-Tupel $\tau = (S, X, \Gamma, \delta, s_0, b)$ heißt Turingmaschine, wenn gilt:

- S : endliche nichtleere Menge von Zuständen
- $s_0 \in S$: Anfangszustand
- $X \subset \Gamma$: Eingabealphabet
- $b \in \Gamma \setminus X$: Blanksymbol
- $\delta: S \times \Gamma \rightarrow S \times \Gamma \times \{L, R, h\}$: partielle Funktion

Definition B: Ein Triple (s, f, i) heißt Konfiguration der Turingmaschine, wenn

- (i) $s \in S$ (aktueller Zustand)
- (ii) $f: \mathbb{Z} \rightarrow \Gamma$ (Bandinhalt: $f(j)$ ist der Inhalt der j -ten Zelle,
es existieren nur endlich viele j mit $f(j) \neq b$)
- (iii) $i \in \mathbb{Z}$ (Position des LSK)

$K_\tau :=$ Menge aller Konfigurationen von τ

Definition C: Arbeitsweise einer Turingmaschine (abstrakte Automaten-symbolik)

(1) Ausgangskonfiguration:

$w \in X^*$, $w = x_0 \dots x_k$, $x_i \in X$, denn

$\alpha: X^* \rightarrow K_\tau$

$\alpha(w) = (s_0, f_w, \delta)$ $f_w =$ Bandinschrift

$f_w(j) = \begin{cases} w_j, & 0 \leq j \leq k \\ b, & \text{sonst} \end{cases}$

(2) Schritt ausführen:

$$\hat{\delta}: K_{\tau} \rightarrow K_{\tau} \text{ mit } \hat{\delta}(s, f, i) = (s', f', i') : \Leftrightarrow$$

$$f(i) = a \in \Gamma, \delta(s, a) = (s', a', P),$$

$$f'(j) = \begin{cases} f(j), & j \neq i \\ a', & j = i \end{cases} \quad i' = \begin{cases} i+1, & \text{falls } P=R \\ i-1, & \text{falls } P=L \\ i, & \text{falls } P=h \end{cases}$$

(3) Ergebnis:

$$w: K_{\tau} \rightarrow \Gamma^* \text{ mit}$$

$$w(s, f, i) = \varepsilon, \text{ falls } f(j) = b \text{ für alle } j \in \mathbb{Z}$$

$$w(s, f, i) = v_0 v_1 \dots v_l, \text{ falls es ein } k \in \mathbb{Z} \text{ gibt, mit}$$

$$f(j) = b \text{ für alle } j < k, f(k) \neq b$$

$$f(j) = b \text{ für alle } j > k+r, f(k+r) \neq b$$

$$f(j) = v_{j-k} \text{ für } k \leq j \leq k+r$$

$\exists ! k$ (Es existiert genau ein $k \Rightarrow$ Ausgabe eindeutig)

$\neq b$		$\neq b$	
$b \dots$	v_0		$v_r \dots b$
k		$k+r$	

Definition D: Sei $\tau = (S, X, \Gamma, \delta, s_0, b)$ eine Turingmaschine. Die von τ berechnete Funktion $h_{\tau}: X^* \rightarrow \Gamma^*$ ist definiert durch

$$h_{\tau}(w) = \begin{cases} w(\hat{\delta}^{n+1}(a(w))), & \text{falls } m = \text{Min}\{j \mid \hat{\delta}^j(\alpha(w)) = (s, f, i) \text{ und} \\ & \text{es ex. } s', a' : \delta(s, f(i)) = (s', a', h)\} \\ \perp, & \text{sonst (Endlosschleife)} \end{cases}$$

Definition E: Seien A, B Alphabete

- a) Eine Abbildung $h: A^* \rightarrow B^*$ (partielle Funktion) heißt Turing-berechenbar, wenn es eine Turingmaschine $\tau = (S, X, \Gamma, \delta, s_0, b)$ gibt, mit $B \subseteq \Gamma$ und $h_\tau = h$
- b) $\tau_{A,B} = \{h \mid h: A^* \rightarrow B^* \text{ Turing-berechenbar}\}$
- c) $\mathcal{T}$ sei die Klasse aller Turing-berechenbarer Funktionen (A, B beliebig)
- d) Eine Menge $M \subseteq A^*$ heißt Haltebereich / Definitionsbereich einer Turingmaschine $\tau = (S, A, \Gamma, \delta, s_0, b)$, wenn $M = \{w \in A^* \mid h_\tau(w) \text{ definiert}\}$
- e) Eine Menge $M \subseteq B^*$ heißt Ergebnisbereich / Wertebereich einer Turingmaschine $\tau = (S, A, \Gamma, \delta, s_0, b)$, mit $B \subseteq \Gamma$, wenn $M = \{v \in B^* \mid \exists w \in A^* : h_\tau(w) = v\}$

Beispiele:

1) $\tau_1 = (\{s_0\}, \{\mid\}, \{b, \mid\}, \delta_1, s_0, b)$

$$\delta(s_0, b) = (s_0, \mid, h), \delta(s_0, \mid) = (s_0, \mid, r)$$

$$h_{\tau_1}: \{\mid\}^* \rightarrow \{b, \mid\}^*, h_{\tau_1}: \{\mid\}^n \rightarrow \mid^{n+1} \quad n \in \mathbb{N}_0$$

Haltebereich: $\{\mid\}^*$

Wertebereich: $\{\mid\}^+$

S	Γ	S	Γ	-
s_0	$\mid$	s_0	$\mid$	r
s_0	b	s_0	$\mid$	h

2) $\tau_2 = (\{s_0\}, \{\mid\}, \{b, \mid\}, \delta_2, s_0, b)$ Nullfunktion

$$h_{\tau_2}: \{w\} = \varepsilon, w \in \{\mid\}^*$$

Haltebereich: $\{\mid\}^*$

Wertebereich: $\{\varepsilon\}$

S	Γ	S	Γ	-
s_0	b	s_0	b	h
s_0	$\mid$	s_0	b	r

3) $\tau_3 = (\{s_0, s_1\}, \{\mid\}, \{b, \mid\}, \delta_3, s_0, b)$ Geradzahligkeitsmaschine

$$h_\tau = \begin{cases} \mid^{n+1}, & \text{falls } n \text{ durch } 2 \text{ teilbar} \\ \perp, & \text{sonst} \end{cases} \quad n \in \mathbb{N}_0$$

Haltebereich: $\{\mid^{2k} \mid k \in \mathbb{N}_0\}$

Wertebereich: $\{\mid^{2k+1} \mid k \in \mathbb{N}_0\}$

S	Γ	S	Γ	-
s_0	b	s_1	$\mid$	h
s_0	$\mid$	s_1	$\mid$	r
s_1	b	s_1	b	r
s_1	$\mid$	s_0	$\mid$	r

4) $\tau_4 = (\{s_0, s_1, s_2, s_3\}, \{|\}, \{, c, b\}, \delta_4, s_0, b)$ Verdoppelungsmaschine

Ziel: $h_{\tau_4} (())^n = c^{2^n}, n \in \mathbb{N}_0$

Beobachte: Der Wertebereich von τ_4
„entspricht“ dem Haltebereich von τ_3

Setze: $\tau_3 \circ \tau_4$

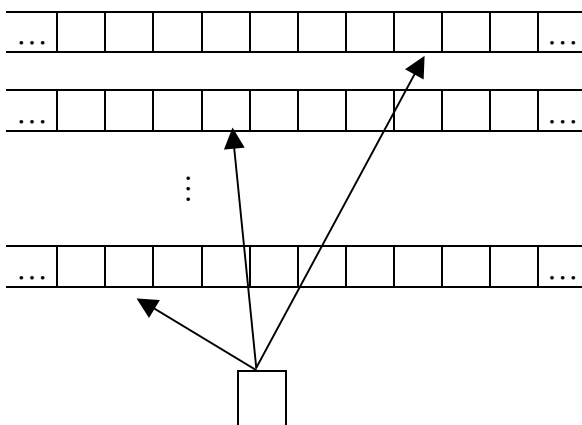
$h_{\tau_3 \circ \tau_4} (())^n = c^{2^{n+1}}$

* Die Alphabete sind isomorph

S	Γ	S	Γ	-
s_0	b	s_0	b	h
s_0		s_1	b	r
s_0	c	s_0	c	h
s_1	b	s_2	c	r
s_1		s_1		r
s_1	c	s_1	c	r
s_2	b	s_3	c	l
s_2		s_2		h
s_2	c	s_2	c	h
s_3	b	s_0	b	r
s_3		s_3		l
s_3	c	s_3	c	l

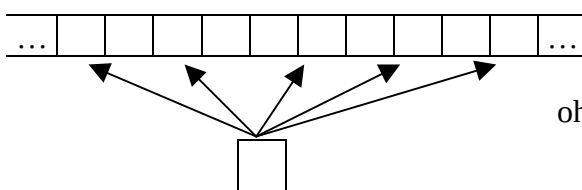
Varianten von Turingmaschinen:

a) k – Band – Turingmaschine mit k – Schreib-/Leseköpfen



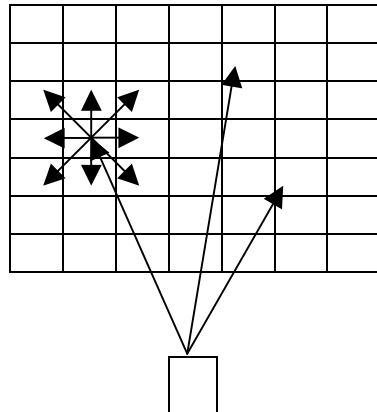
Man gewinnt nichts an Leistungsfähigkeit: Simuliere dazu k-Bänder durch ein einzelnes

b) k – Kopf – Turingmaschine



ohne Gewinn

c) k- dimensionale – Turingmaschine mit m – Köpfen



Simulation durch 1 Band Turingmaschine möglich

d) Endzustände statt der Aktion „halt“

6.2. Registermaschinen

Turingmaschinen: unendlich viel Speicher / endlicher Wertebereich der Zelle

Registermaschinen: endlich viel Speicher / unendlicher Wertebereich $\mathbb{N}_0$

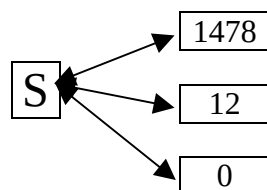
Definition F: Eine k – Registermaschine ρ beschreibt man durch ein 5 – Tupel

$\rho = (S, k, \delta, s_0, F)$, wobei

- (i) S ist die endliche Menge $\neq \emptyset$ von Zuständen
- (ii) $s_0 \in S$ Anfangszustand
- (iii) $k \in \mathbb{N}$ Anzahl der Register
- (iv) $F \subseteq S$ Endzustand
- (v) $\delta: (S \setminus F) \times \{0, 1\}^k \rightarrow S \times \{-1, 0, +1\}^k$

Nulltest
„0“: „=0“
„1“: „=1“

Registeroperationen



(3 Registermaschine)

Definition G:

- (i) $K_p = S \times \mathbb{N}_0^k$ heißt Menge der Konfiguration von $\tilde{n}$
 - (ii) $\alpha: \mathbb{N}_0 \rightarrow K_p$ ist definiert durch $\alpha(n) = (s_0, (n, 0, 0, \dots, 0))$
 - (iii) $\hat{\delta}: K_p \rightarrow K_p$ mit $\hat{\delta}(s, (i_1, \dots, i_k)) = (s', (i'_1, \dots, i'_k)) : \Leftrightarrow$ es gibt ein $\delta(s, (\text{sign}(i_1), \dots, \text{sign}(i_k))) = (s', (j_1, \dots, j_k))$
 $i_1, \dots, i_k$ aktuelle Registerbelegung
 $i'_1, \dots, i'_k$ neue Registerbelegung

$$i'_r = \begin{cases} i_r + j_r, & \text{sonst} \\ 0, & \text{falls } i_r = 0, j_r = -1 \text{ [modifizierte Subtraktion :]} \end{cases}$$

Wende Registeroperationen +1, -1, 0 auf die Register an
 - (iv) $\omega: K_p \rightarrow \mathbb{N}_0$ mit $\omega(s, (i_1, \dots, i_k)) = i_1$, falls $s \in F$
 - (v) $h_p: \mathbb{N}_0 \rightarrow \mathbb{N}_0$ ist definiert durch
- $$h_p(n) = \begin{cases} \omega(\hat{\delta}^m(\alpha(n))), & \text{falls : es ex. } m = \text{Min}\{j \mid \hat{\delta}^j(\alpha(n)) = (s, (i_1, \dots, i_k)) \text{ mit } s \in F\} \\ \perp, & \text{sonst} \end{cases}$$
- ist die berechnete Funktion von p .

Definition H:

- (i) eine Abbildung $h: \mathbb{N}_0 \rightarrow \mathbb{N}_0$ heißt Registermaschine k - berechenbar, wenn es eine Registermaschine mit k Registern gibt, so dass $h = h_p$
- (ii) $\mathcal{Rm}_k = \{h \mid h \text{ ist Registermaschine } k \text{ - berechenbar}\}$
- (iii) $\mathcal{Rm} = \bigcup_{k \geq 1} \mathcal{Rm}_k$
- (iv) $M \subseteq \mathbb{N}_0$ heißt Haltebereich einer Registermaschine, wenn es ein k und eine Registermaschine p mit k -Registern gibt, so dass
 $M = \{n \in \mathbb{N}_0 \mid h_p(n) \text{ ist definiert}\}$
- (v) $M \subseteq \mathbb{N}_0$ heißt Werte-/Ergebnisbereich einer Registermaschine, wenn es ein k gibt und eine Registermaschine p mit
 $M = \{m \in \mathbb{N}_0 \mid \exists n \in \mathbb{N}_0 : h_p(n) = m\}$

Beispiele: 1) $\rho_1 = (\{s_0, s_1\}, 1, \delta, s_0, \{s_1\})$

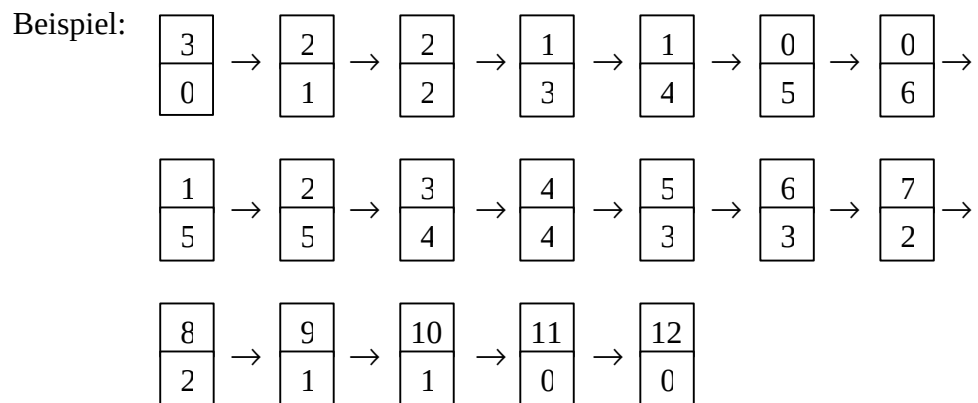
δ	S	Tests	S	Operation
	s_0	0	s_1	+1
	s_0	1	s_0	-1
	s_1	0	-	-
	s_1	1	-	-

Zähle Register auf 0 und addiere 1: $h_{\rho_1}(n) = 1$

„-“ $\hat{=}$ beliebig

2) $\rho_2 = (\{s_0, s_1, s_2, s_3, s_4\}, 2, \delta, s_0, \{s_4\})$

δ	S	Tests	S	Operation
	s_0	1 -	s_1	-1 +1
	s_0	0 0	s_4	0 0
	s_0	0 1	s_2	0 0
	s_1	- -	s_0	0 +1
	s_2	- 1	s_3	+1 -1
	s_2	- 0	s_4	0 0
	s_3	- -	s_2	+1 0



Arbeitsweise:

$$(s_0, \begin{array}{|c|} \hline n \\ \hline 0 \\ \hline \end{array}) \rightarrow^* (s_2, \begin{array}{|c|} \hline 0 \\ \hline 2n \\ \hline \end{array}) \rightarrow^* (s_4, \begin{array}{|c|} \hline 4n \\ \hline 0 \\ \hline \end{array})$$

Offenbar: $h_{\rho_2}(n) = 4n$

3) Erweiterung: Will man $h: \mathbb{N}_0^r \rightarrow \mathbb{N}_0^s$, so modifiziert man α und ω . So kann man auch die Addition berechnen.

Anhang:

Kapitel 5 Definition A: (typisieren von Grammatiken)

Sei $G = (N, T, P, \sigma)$ eine Grammatik

- a) G heißt vom Typ 3 oder rechtslinear oder regulär, wenn für alle $(\alpha, \beta) \in P$ gilt: $\alpha \in N, \beta \in T^* \cdot (\{\varepsilon\} \cup N)$
- b) G heißt vom Typ 2 oder kontextfrei, wenn für alle $(\alpha, \beta) \in P$ gilt: $\alpha \in N$
- c) G heißt vom Typ 1 oder kontextsensitiv, wenn für alle $(\acute{\alpha}, \hat{\alpha}) \in P$ gilt:
 $\alpha = \alpha_1 B \alpha_2, \beta = \alpha_1 \bar{\beta} \alpha_2$ für ein $B \in N$ und $\alpha_1, \alpha_2, \bar{\beta} \in (N \cup T)^*$
[kontextsensitiv: Nonterminals dürfen nur ersetzt werden, wenn der Kontext (α_1, α_2) stimmt.]
- d) G heißt vom Typ 0, wenn G Grammatik ist.
- e) G heißt Erweiterungstyp, wenn für alle $(\alpha, \beta) \in P$ gilt: $|\alpha| \leq |\beta|$
[Wortproblem lösbar]
- f) G heißt lineare Grammatik, wenn G kontextfrei ist und für alle $(\alpha, \beta) \in P$ gilt: $\beta \in T^* \cdot N \cdot T^* \cup T^*$

Literaturverzeichnis:

- [1] Erk, Katrin und Priebe, Lutz: Theoretische Informatik / Eine umfassende Einführung.
Springer-Verlag, Berlin, Heidelberg, New York, 1.Auflage, 2000
- [2] Schöning, Uwe: Theoretische Informatik – kurzgefasst
Spektrum-Verlag, Heidelberg, Berlin, 3.Auflage, 1999