Turingmaschinen

Def

Eine Turingmaschine (TM) ist ein 7-Tupel

$$M = (Z, \Sigma, \Gamma, \delta, z_0, \sqcup, E)$$

- Z: endl. Zustandsmenge
- Σ: Eingabealphabet
- $\Gamma \supseteq \Sigma$: Bandalphabet
- $\delta: Z \times \Gamma \to Z \times \Gamma \times \{L, R, N\}$ det. TM

  $\to 2^{Z \times \Gamma \times \{L, R, N\}}$ nicht-det. TM

  Transitionsfkt.
- $z_0 \in Z$ Anfangszustand
- $\sqcup \in \Gamma - \Sigma$ Blanksymbol
- $E \subseteq Z$ Endzustände

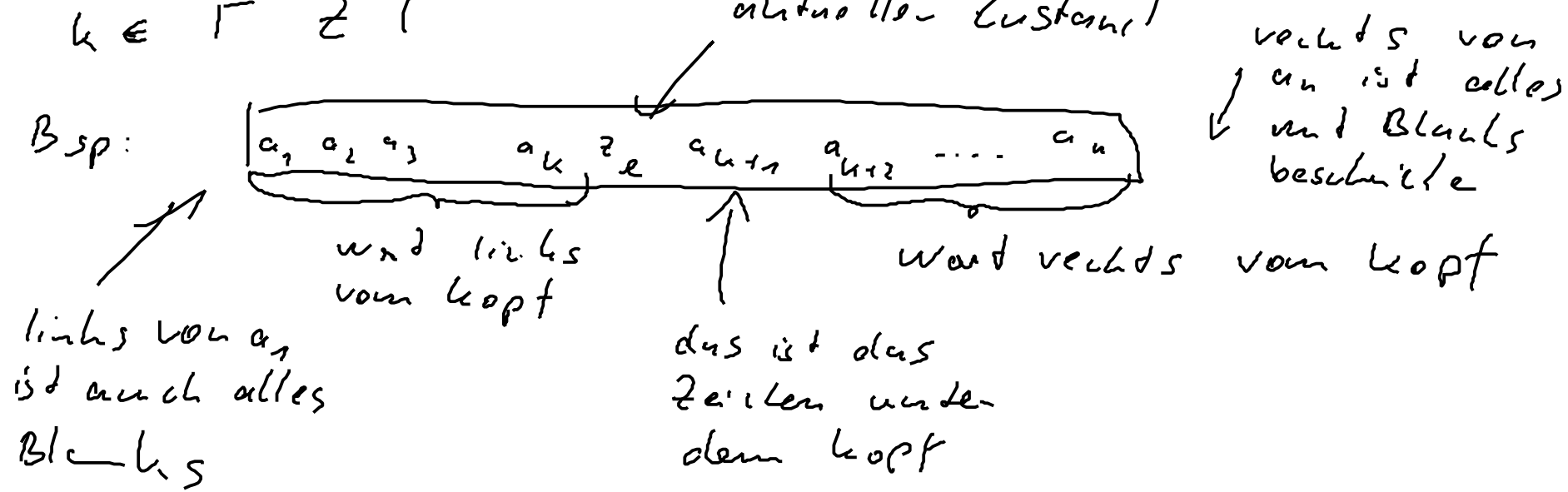
Idee:

$$\delta(z, a) \overset{=}{\Rightarrow} (z', a', x) \qquad x \in L, R, N$$

wenn die Maschine M im Zustand z ist und unter dem Lese-Schreibkopf ist schreibe a' aufs Band und bewege den Kopf in die entsprechende Richtung (links, rechts, stehen bleibe).

Def Eine Konfiguration einer TM ist ein Wort der Art

$k \in \Gamma^* Z \Gamma^*$

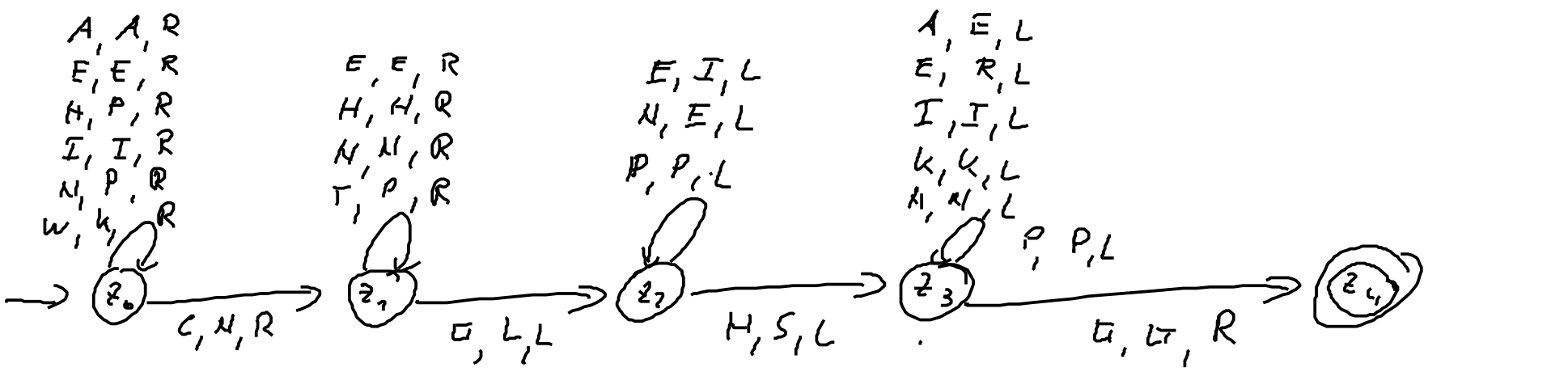


Def. Die Nachfolgekonfigurationsrelation $\vdash$ für det. TMs ist wie folgt def.:

$a_1 \dots a_{m-1} a_m z\, b_1 b_2 \dots b_n \vdash$

- $a_1 a_2 \dots a_m z' c\, b_2 \dots b_n$ falls $\delta(z, b_1) = (z', c, N)$
- $a_1 a_2 \dots a_m c\, z' b_2 \dots b_n$ falls $\delta(z, b_1) = (z', c, R)$, $m \geq 0$, $n \geq 2$
- $a_1 a_2 \dots a_n c\, z' \sqcup$ falls $\delta(z, b_1) = (z', c, R)$, $m \geq 0$ $n = 1$
- $a_1 a_2 \dots a_{m-1} z' a_m c\, b_2 \dots b_n$ falls $\delta(z, b_1) = (z', c, L)$, $m \geq 1$, $n \geq 1$
- $z' \sqcup c\, b_2 \dots b_n$ falls $\delta(z, b_1) = (z', c, L)$ $m = 0$, $n \geq 1$

Sinngemäß für n.i.d-det. TMs.



Def Die von einer TM M erkannte Sprache ist wie folgt def.:

$$T(M) = \{ x \in \Sigma^* \mid z_0 x \vdash^* \alpha z \beta,\ \alpha, \beta \in \Gamma^*,\ z \in E \}$$

Für Typ 1 - Sprachen braucht wir ein anderes Beschreibungsmittel

→ "Linear Bounded Automata" (LBA).

Erweiterung von Σ zu $\Sigma' = \Sigma \cup \{ \hat{a} \mid a \in \Sigma \}$. Eingabe ist dann $a_1 a_2 \dots a_{n-1} \hat{a}_n$

Def Eine nicht-det. TM heißt linear beschränkt wenn für alle Eingaben $a_1 a_2 \dots a_{n-1} a_n \in \Sigma^*$ und alle Konfigurationen $\alpha z \beta$ mit $z_0 a_1 a_2 \dots a_{n-1} \hat{a}_n \vdash^* \alpha z \beta$ gilt $|\alpha\beta| = n$. Die akz. Sprache ist dann:

$$T(M) = \{ a_1 a_2 \dots a_{n-1} a_n \in \Sigma^* \mid z_0 a_1 a_2 \dots a_{n-1} \hat{a}_n \vdash^* \alpha z \beta, \; \alpha, \beta \in \Gamma^*, \; z \in E \}.$$

Satz Die von LBAs erkannte Sprache sind genau alle Typ 1 - Sprachen.

Beweis

($\in$) Sei L eine Typ 1 - Sprache und $L = L(G)$, $G = (V, \Sigma, P, S)$.
Prinzipieller Aufbau einer nicht-det. TM M:
Schleife:

1) TM wählt eine der Regeln der Grammatik G aus: $u \to v$ $u, v \in (V \cup \Sigma)^*$, $|u| \leq |v|$.
2) Die TM sucht in der Eingabe ein Vorkommen von u.
3) Die TM ersetzt u durch v (ggfs mit Linksver-

verschieb_j des Teilwortes rechts von v).

4) Die TM löscht, falls nur noch S auf dem Band steht.
Falls ja, geht sie in den Endzustand.

Dann gilt:

$x \in L(G)$ gdw. es ex. eine Ableitung $S \Rightarrow^* x$

gdw. es ex. eine Rechnung auf M, die Ableitung rückwärts simuliert

gdw. $x \in T(M)$.

Da für $v \to v \in P$: $|v| \leq |v|$, ist M linear beschränkt.

($\Rightarrow$) Sei $A = T(M)$ für LBA M.

Wir erweitern Γ zu $\Delta = \Gamma \cup (Z \times \Gamma)$.

Die Konfiguration $k = \overbrace{a_1 \dots a_m z b_1 b_2 \dots b_n}^{m+1+n}$ wird dargestellt

durch $\tilde{k} = \underbrace{a_1 \dots a_m (z, b_1) b_2 \dots b_n}_{m+n}$

Also können δ-Übergänge als Chom-Regeln beschrieben werden

$$\boxed{\delta(z,a) \ni (z', b, L)}$$

$c(z,a) \to (z',c)\, b$ für alle $c \in \Gamma$

Die gesamte Regelsatz sei P'. Damit können wir Rechnungen von M simulieren:

$k \vdash^* k'$ gdw. mittels P' $\tilde{k} \Rightarrow^* \tilde{k}'$

Konstruiere jetzt $G = (V, \Sigma, P, S)$ mit

$V = \{S, A\} \cup (\Delta \times \Sigma)$

$P = \{ S \to A(\hat{a}, a) \mid a \in \Sigma \} \cup$ (1)

$\{ A \to A(a, a) \mid a \in \Sigma \} \cup$ (2)

$\{ A \to ((z_0, a), a) \mid a \in \Sigma \} \cup$ (3)

} erzeugt Anfangskonf und Eingabewort

$\{ (\alpha_1, a)(\alpha_2, b) \to (\beta_1, a)(\beta_2, b) \mid (\alpha_1 \alpha_2 \to \beta_1 \beta_2 \in P') \}$ (4)

$a, b \in \Sigma \} \cup$

} simuliert Rechnung auf den ersten Komponenten

jetzt nur wenn $z \in E$

$\{ ((z, a), b) \to b \mid z \in E, a \in \Gamma, b \in \Sigma \} \cup$ (5)

$\{ (a, b) \to b \mid a \in \Gamma, b \in \Sigma \}$ (6)

} aus der 2. Komponente wird das Eingabewort extrahiert

und (1)-(3) $S \Rightarrow^* ((z_0, a_1), a_1)((a_2, a_2) \dots (a_n, a_n) \Rightarrow^* (\alpha_1, a_1) \dots (\alpha_n, a_n)$

$\Rightarrow^* a_1 \dots a_n$

Konf $\tilde{k} = (z_0, a_1) \dots \hat{a}_n$ Eingabewort für M: $a_1 \dots a_n$

$x \in T(M)$ ddr. $z_0 x \vdash^* \alpha z \beta$ mit $z \in E$

gdw. $S \Rightarrow^* ((z_0, a_1), a_2) \dots (\hat{a}_n, a_n) \quad x = a_1 \dots a_n$

$\Rightarrow^* a_1 \dots a_n$

gdw. $x \in L(G)$

Offensichtlich: Alle Regeln sind vom Typ 1! □

Bem: Kind-Determinismus ist erforderlich.

Es ist unbekannt, ob LBAs durch det. LBAs akzeptiert sind.