Definition: *deterministischer Kellerautomat*

Ein nichtdeterministischer Kellerautomat heißt *deterministisch*, falls gilt:

Für alle $z \in Z$, $a \in \Sigma$, $A \in \Gamma$ ist

$$|\delta(z, a, A)| + |\delta(z, \varepsilon, A)| \leq 1.$$

Erläuterung: Dies bedeutet, daß der Automat zu jedem Zeitpunkt höchstens eine Alternative hat. Daher der Name *deterministisch*.

Um die von einem deterministischen Kellerautomaten M akzeptierte Sprache $L(M)$ zu definieren, müssen wir $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ noch um eine Menge $E \subseteq Z$ von *Endzuständen* erweitern. $L(M)$ besteht dann aus genau den Worten, für die es eine Folge von Übergängen gibt, so daß sich M nach Abarbeiten des Wortes in einem Endzustand befindet.

Definition. Eine kontextfreie Grammatik G heißt *deterministisch kontextfrei*, falls es einen deterministischen Kellerautomaten M gibt mit $L(G) = L(M)$.

Ausblick: $LR(k)$-Grammatiken

Idee: Wir wollen eine Grammatik, für die wir zu einem gegebenem Wort ”schnell” eine Ableitung erzeugen können.

Beispiel: Arithmetische Ausdrücke

Regeln:

$$\begin{aligned}
S &\to A \\
A &\to E \\
A &\to A + E \\
A &\to A - E \\
E &\to P \\
E &\to E * P \\
E &\to E/P \\
P &\to (A) \\
P &\to a
\end{aligned}$$

Problem: Beim Reduzieren von links kann man in Sackgassen laufen...

Behebung: Einführen von ”Lookaheads” (der Länge k). Interpretation: Die Regel darf nur dann angewendet werden, wenn die nächsten k Zeichen mit den erlaubten Lookaheads übereinstimmen.

Beispiel: Arithmetische Ausdrücke, Forts.

Regeln	Lookaheads (der Länge 1)
$S \to A$	ϵ
$A \to E$	$+, -,), \epsilon$
$A \to A + E$	$+, -,), \epsilon$
$A \to A - E$	$+, -,), \epsilon$
$E \to P$	beliebig
$E \to E * P$	beliebig
$E \to E/P$	beliebig
$P \to (A)$	beliebig
$P \to a$	beliebig

Definition: $LR(k)$-Grammatik
Eine kontextfreie Grammatik ist eine $LR(k)$-Grammatik, wenn man durch Lookaheads der Länge k erreichen kann, daß bei einer Reduktion von links nach rechts in jedem Schritt genau eine Regel anwendbar ist.

Korollar: Jede kontextfreie Sprache für dies es eine $LR(k)$-Grammatik gibt, ist deterministisch kontextfrei.

Definition. Eine *nichtdeterministische Turingmaschine* (kurz TM) wird durch ein 7-Tupel $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ beschrieben, das folgende Bedingungen erfüllt:

- Z ist eine endliche Menge von *Zuständen*.
- Σ ist eine endliche Menge, das *Eingabealphabet*.
- Γ ist eine endliche Menge, das *Bandalphabet*.
- $\delta : Z \times \Gamma \to \mathcal{P}(Z \times \Gamma \times \{L, R, N\})$, die *Übergangsfunktion*.
- $z_0 \in Z$ ist der *Startzustand*.
- $\square \in \Gamma \setminus \Sigma$, das *Leerzeichen*.
- $E \subseteq Z$, die Menge der *Endzustände*.

Die Turingmaschine heißt *deterministisch*, falls gilt

$$|\delta(z, a)| = 1 \qquad \text{für alle } z \in Z, a \in \Gamma.$$

Erläuterung: Intuitiv bedeutet $\delta(z, a) = (z', b, x)$ bzw. $\delta(z, a) \ni (z', b, x)$:

Wenn sich M im Zustand z befindet und unter dem Schreib-/Lesekopf das Zeichen a steht, so geht M im nächsten Schritt in den Zustand z' über, schreibt an die Stelle des a's das Zeichen b und bewegt danach den Schreib-/Lesekopf um eine Position nach rechts (falls $x = R$), links (falls $x = L$) oder läßt ihn unverändert (falls $x = N$).

Definition. Eine *Konfiguration* einer Turingmaschine ist ein Tupel $(\alpha, \beta, z) \in \Gamma^* \times \Gamma + \times Z$. Interpretation: Das Wort $w = \alpha\beta$ entspricht der Belegung des Bandes, wobei dieses rechts und links von w mit dem Leerzeichen $\square$ gefüllt sei. Der Schreib-/Lesekopf befindet sich auf dem ersten Zeichen von β.

Die *Startkonfiguration* der Turingmaschine bei Eingabe $x \in \Sigma^*$ entspricht der Konfiguration (ϵ, x, z_0). D.h., auf dem Band befindet sich genau die Eingabe $x \in \Sigma^*$, der Schreib-/Lesekopf befindet sich auf dem ersten Zeichen der Eingabe und die Maschine startet im Zustand z_0.

Definition. Eine Turingmaschine heißt *linear beschränkt* (kurz: LBA), falls für alle $z \in \mathbb{Z}$ gilt:

$$(z', a) \in \delta(z, \square) \qquad \Longrightarrow \qquad a = \square.$$

D.h., ein Leerzeichen wird nie durch ein anderes Zeichen überschrieben. Mit anderen Worten: Die Turingmaschine darf ausschliesslich die Positionen beschreiben, an denen zu Beginn die Eingabe x stand.

Satz. Die von linear beschränkten, nichtdeterministischen Turingmaschinen akzeptierten Sprachen sind genau die kontextsensitiven Sprachen.

Bew: siehe Literatur.

Analog kann man zeigen:

Satz. Die von Turingmaschinen akzeptierten Sprachen sind genau die Typ 0 Sprachen.

Hinweis. Hier ist es egal, ob man die Turingmaschine deterministisch oder nichtdeterministisch wählt. Genauer gilt: Zu jeder nichtdeterministischen Turingmaschine M gibt es eine deterministische Turingmaschine M' mit $L(M) = L(M')$.

Chomsky-Hierarchie

Typ 3	reguläre Grammatik DFA NFA regulärer Ausdruck
Det. kf.	$LR(k)$-Grammatik deterministischer Kellerautomat
Typ 2	kontextfreie Grammatik (nichtdeterministischer) Kellerautomat
Typ 1	kontextsensitive Grammatik (nichtdeterministische) linear beschränkte Turingmaschine
Typ 0	endliche Grammatik Turingmaschine

Abschlußeigenschaften

	Schnitt	Vereinigung	Komplement	Produkt	Stern
Typ 3	ja	ja	ja	ja	ja
Det. kf.	nein	nein	ja	nein	nein
Typ 2	nein	ja	nein	ja	ja
Typ 1	ja	ja	ja	ja	ja
Typ 0	ja	ja	nein	ja	ja

Wortproblem

Typ 3, gegeben als DFA	lineare Laufzeit
Det. kf, gegeben als DPDA	lineare Laufzeit
Typ 2, gegeben durch Grammatik in CNF	CYK-Algorithmus, Laufzeit $\mathcal{O}(n^3)$
Typ 1	$\langle$nächstes Kapitel$\rangle$
Typ 0	$\langle$nächstes Kapitel$\rangle$

Entscheidbarkeit

	Wortproblem	Leerheitsproblem	Äquivalenzproblem	Schnittproblem
Typ 3	ja	ja	ja	ja
Det. kf.	ja	ja	ja	nein
Typ 2	ja	ja	nein	nein
Typ 1	ja	nein	nein	nein
Typ 0	nein	nein	nein	nein