

# **Vorlesungsskript**

**zur Vorlesung „Theoretische Informatik“**

*Vorlesung vom 17.11.2000*

# Deterministisch-Kontextfreie Sprachen

$$L_{DPDA} \subset L_{NPDA}$$

Die Klasse der deterministisch-kontextfreien Sprachen ist eine echte Teilmenge der nichtdeterministisch-kontextfreien Sprachen. Es gibt also für alle DPDA's einen NPDA, welcher genauso leistungsfähig ist wie sein entsprechender DPDA. Umgekehrt folgt daraus also, dass es nicht für jeden NPDA einen DPDA gibt, welcher die gleiche Sprache akzeptiert.

Trotzdem sind DPDA's für die Praxis relevant, da man in der Regel eben mit deterministischen Systemen arbeitet. Sie werden z.B. zur Syntaxanalyse im Compilerbau benutzt.

Satz S: Die Klasse der det. kf. Sprachen  $L_{DPDA}$  ist

- a) nicht abgeschlossen gegen Vereinigung, Durchschnitt, Homomorphismen, Konkatenation, Sternbildung und Spiegelung
- b) abgeschlossen gegen Komplement, Durchschnitt mit regulären Sprachen.

Glaubhaftmachung:

1.)  $L = \{ww^R \mid w \in \{a,b\}^*\}$ .  $L$  ist kontextfrei und linear (siehe Übung), aber nicht deterministisch-kontextfrei, denn das eingelesene Wort  $w$  muß zunächst „geklert“ und schließlich Zeichen für Zeichen mit  $w^R$  „gematcht“ werden. Ein DPDA findet die Mitte nicht. Ein NPDA errät die Mitte und „matcht“ dann. Es gibt somit nicht für jede kontextfreie Sprache einen entsprechenden DPDA

2.)  $L' = \{wcw^R \mid w \in \{a,b\}^*\}$  ist det. kontextfrei. „c“ ist das Mittelsymbol. Setze nun

$h: \{a,b,c\}^* \rightarrow \{a,b\}^*$  mit  $h(a) = a$ ,  $h(b) = b$  und  $h(c) = \varepsilon$ . Dann ist  $h(L') = L \in L_{DPDA}$ . Somit ist  $L_{DPDA}$  nicht abgeschlossen gegen Homomorphismen. Die Aussage gilt auch für  $\varepsilon$ -freie Homomorphismen; setze dazu  $h(c)=aa$ .

3.)  $L'' = \{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\} \notin L_{DPDA}$

(wobei  $\{a^n b^n \mid n \geq 0\} \in L_{DPDA}$  und  $\{a^n b^{2n} \mid n \geq 0\} \in L_{NPDA}$ )

Ein DPDA kann nach Kellern von  $a^n$  nicht erkennen, ob anschließend jeweils  $b^2$  oder  $b$  gegen jedes  $a$  „gematcht“ werden muß. Offenbar gilt  $L'' \in L_{NPDA}$ .

$\Rightarrow L_{DPDA}$  ist nicht abgeschlossen gegen Vereinigung.

4.)  $L''' = \{a^n b^n c^m \mid n, m \geq 0\} \cap \{a^m b^n c^n \mid n, m \geq 0\} = \{a^n b^n c^n \mid n \geq 0\} \notin CF$

(wobei  $\{a^n b^n c^m \mid n, m \geq 0\} \in L_{DPDA}$  und  $\{a^m b^n c^n \mid n, m \geq 0\} \in L_{DPDA}$ )

$\Rightarrow L_{DPDA}$  ist nicht abgeschlossen gegen Durchschnitt.

5.) Konkatenation, Sternbildung, Spiegelung selbst überlegen.

Das war soweit Satz S,a). Nun zum Komplement. Benötigt wird hierzu Normalform von DPDAs.

**Probleme bei der Komplementbildung von DPDAs:**

- 1.) Der DPDA „friert ein“: Dann gibt es zu einer Konfiguration keine Folgekonfiguration, weil  $\delta$  undefiniert ist. Gilt nun  $(s, v, b_0 z) \in \hat{\delta}^k(s_0, w, b_0)$  und  $v \neq \varepsilon$  und der DPDA  $\rho$  „hänge“ in dieser Konfiguration, dann ist  $w$  nicht in der erkannten Sprache  $L(\rho)$ . Umgekehrt würde auch der Komplementautomat für die Komplementsprache in diesem Zustand hängen.  $\Rightarrow w \notin L(\rho), w \notin \overline{L(\rho)}$
- 2.) Wenn  $\rho$  ein Wort  $w$  komplett abgearbeitet hat und in einem Endzustand ist, kann  $\rho$  noch per  $\varepsilon$ -Übergänge in andere Nicht-Endzustände wechseln. Es existieren also Ableitungen:

$$(s, \varepsilon, b_0 z) \in \hat{\delta}^k(s_0, w, b_0), s \in F$$

$$(s', \varepsilon, b_0 z') \in \hat{\delta}^j(s_0, w, b_0), s' \notin F$$

Der DPDA für die Komplementsprache kann ebenso rechnen; wieder ist  $w \in L(\rho)$ .

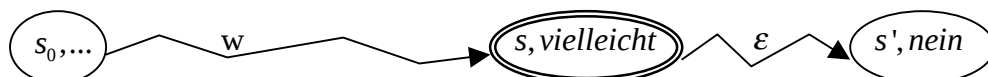
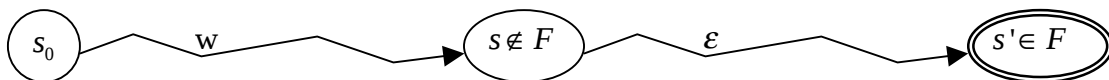
### Einschub:

Determinismus: Eindeutige Festlegung des Folgezustands der Rechenschritte

Determiniertheit: Funktionale Abhängigkeit zwischen Eingabe und Ausgabe

|                       | determiniert                         | nicht-determiniert                                  |
|-----------------------|--------------------------------------|-----------------------------------------------------|
| deterministisch       | Beliebige deterministische Programme | x:=random(1,2)<br>x:=x+x<br>write(x)                |
| nicht-deterministisch | Quicksort                            | x:=random(0,1)<br>if x=0 then x:=x+y<br>else x:=x*z |

Beide Probleme löst man, indem man zu einer Normalform von  $\rho$  übergeht und beim Übergang zum Komplementautomaten bestimmte Vorkehrungen trifft. Genauer:  
Zeige, dass man jeden DPDA so umformen kann, dass er nicht einfriert. Ferner ist beim Komplementautomaten nicht jeder Nicht-Endzustand von  $\rho$  ein Endzustand. Vielmehr merkt der Komplementautomat sich, dass er in einem seiner Endzustände akzeptieren könnte, wertet aber noch die  $\varepsilon$ -Übergänge von diesem Zustand ab, um zu sehen, ob  $\rho$  nicht noch akzeptiert.



**Lemma:** Zu jedem det. PDA  $\rho = (X, \Gamma, S, \delta, s_0, b_0, F)$  kann man einen DPDA

$\rho' = (X, \Gamma', S', \delta', s_0', b_0, F')$  konstruieren, mit:

- (1)  $\delta'$  ist vollständig definiert:

Dies erreicht man durch Hinzunahme eines absorbierenden Zustands.

- (2) Für alle  $s, s' \in S, \alpha \in X \cup \{\varepsilon\}, c \in \Gamma', v \in \Gamma^*$  gilt:

Aus  $\delta'(s, a, c) = (s', v)$  folgt:  $v = \varepsilon$  oder  $v = c$  oder  $|v| \leq 2$ .

Dies erreicht man durch Einführen neuer Zustände. Will man mehr als 2 Zeichen auf den Keller legen, definiert man Zwischenzustände.

- (3)  $\forall w \in X^* : \rho'$  terminiert bei Eingabe von  $w$  und  $\rho'$  liest das gesamte Wort  $w$ , d.h.

$\forall w \in X^*$  existiert ein  $k \in \mathbb{N}_0, v \in \Gamma^*$  und  $s \in S$  mit

(i)  $(s, \varepsilon, w) = \hat{\delta}^k(\alpha(w))$  [ $w$  vollständig gelesen]

(ii)  $\hat{\delta}^{k+1}(\alpha(w))$  ist nicht def. [weil ein Eingangszeichen notwendig ist]

Beweis: Schwierig [ $\rightarrow$  Hopcroft-Ullman]. Zeige, dass es zu keinem Zustand und zu keinem Kellersymbol  $c \in \Gamma'$  eine unendliche  $\varepsilon$ -Übergangsfolge gibt. Dies ist für DPDAs entscheidbar.

- (4)  $L(\rho) = L(\rho')$