## Beispiel 62

Wir wollen sehen, dass die Sprache

$$\{a^i b^i c^i;\ i \in \mathbb{N}_0\}$$

nicht kontextfrei ist.

Wäre sie kontextfrei, so könnten wir das Wort $a^n b^n c^n$ ($n$ die Konstante aus dem Pumping-Lemma) aufpumpen, ohne aus der Sprache herauszufallen. Wir sehen aber leicht, dass das Teilwort $v$ nur aus $a$'s bestehen kann und bei jeder möglichen Verteilung des Teilworts $x$ Pumpen entweder die Anzahl der $a$'s, $b$'s und $c$'s unterschiedlich ändert oder, wenn

$$(\#_a(vx) =)\ \#_b(vx) = \#_c(vx) > 0\ ,$$

dass $b$'s und $c$'s in der falschen Reihenfolge auftreten.

Zur Vereinfachung von Beweisen wie in dem gerade gesehenen Beispiel führen wir die folgende Verschärfung des Pumping-Lemmas ein:

## Satz 63 (Ogdens Lemma)

*Für jede kontextfreie Sprache* $L$ *gibt es eine Konstante* $n \in \mathbb{N}$, *so dass für jedes Wort* $z \in L$ *mit* $|z| \geq n$ *die folgende Aussage gilt: Werden in* $z$ *mindestens* $n$ *(beliebige) Buchstaben markiert, so lässt sich* $z$ *zerlegen in*

$$z = uvwxy,$$

*so dass*

1. *in* $vx$ *mindestens ein Buchstabe und*
2. *in* $vwx$ *höchstens* $n$ *Buchstaben markiert sind und*
3. $(\forall i \in \mathbb{N}_0)[uv^iwx^iy \in L]$ .

**Bemerkung:** Das Pumping-Lemma ist eine triviale Folgerung aus Ogdens Lemma (markiere alle Buchstaben in $z$).

## Beweis:

Sei $G = (V, \Sigma, P, S)$ eine Grammatik in Chomsky-Normalform mit $L(G) = L$. Wähle $n = 2^{|V|+1}$. Sei $z \in L$ und seien in $z$ mindestens $n$ Buchstaben markiert. In einem Ableitungsbaum für $z$ markieren wir alle (inneren) Knoten, deren linker *und* rechter Teilbaum *jeweils* mindestens ein markiertes Blatt enthalten. Es ist nun offensichtlich, dass es einen Pfad von der Wurzel zu einem Blatt gibt, auf dem mindestens $|V| + 1$ markierte innere Knoten liegen.

## Beweis:

...

Wir betrachten die letzten $|V|+1$ markierten inneren Knoten eines Pfades mit maximaler Anzahl markierter Knoten; nach dem Schubfachprinzip sind zwei mit demselben Nichtterminal, z.B. $A$, markiert. Wir nennen diese Knoten $v_1$ und $v_2$. Seien die Blätter des Teilbaumes mit der Wurzel $v_2$ insgesamt mit $w$ und die Blätter des Teilbaumes mit der Wurzel $v_1$ insgesamt mit $vwx$ beschriftet. Es ist dann klar, dass die folgende Ableitung möglich ist:

$$S \to^* uAy \to^* uvAxy \to^* uvwxy\,.$$

Es ist auch klar, dass der Mittelteil dieser Ableitung weggelassen oder beliebig oft wiederholt werden kann.

## Beweis:

...

Es bleibt noch zu sehen, dass $vx$ mindestens einen und $vwx$ höchstens $n$ markierte Buchstaben enthält. Ersteres ist klar, da auch der Unterbaum von $v_1$, der $v_2$ nicht enthält, ein markiertes Blatt haben muss.

Letzteres ist klar, da der gewählte Pfad eine maximale Anzahl von markierten inneren Knoten hatte und unterhalb von $v_1$ nur noch höchstens $|V|$ markierte Knoten auf diesem Pfad sein können. Der Teilbaum mit Wurzel $v_1$ kann also maximal $2^{|V|+1} = n$ markierte Blätter haben. Formal kann man z.B. zeigen, dass ein Unterbaum, der auf jedem Ast maximal $k$ markierte (innere) Knoten enthält, höchstens $2^k$ markierte Blätter enthält. $\square$

## Beispiel 64

$$L = \{a^i b^j c^k d^l;\ i = 0 \text{ oder } j = k = l\}\,.$$

Sei $n$ die Konstante aus Ogdens Lemma. Betrachte das Wort $ab^nc^nd^n$ und markiere darin $bc^nd$. Nun gibt es eine Zerlegung $ab^nc^nd^n = uvwxy$, so dass $vx$ mindestens ein markiertes Symbol enthält und $uv^2wx^2y \in L$.

Es ist jedoch leicht zu sehen, dass dies einen Widerspruch liefert, da $vx$ höchstens zwei verschiedene der Symbole $b$, $c$, $d$ enthalten kann, damit beim Pumpen nicht die Reihenfolge durcheinander kommt.

## 4.5 Algorithmen für kontextfreie Sprachen/Grammatiken

### Satz 65

*Sie $G = (V, \Sigma, P, S)$ kontextfrei. Dann kann die Menge $V'$ der Variablen $A \in V$, für die gilt:*

$$(\exists w \in \Sigma^*)[A \rightarrow^* w]$$

*in Zeit $O(|V| \cdot s(G))$ berechnet werden.*

### Beweis:

Betrachte folgenden Algorithmus:

$\Delta := \{A \in V;\ (\exists(A \rightarrow w) \in P$ mit $w \in \Sigma^*\}$; $V' := \emptyset$;
**while** $\Delta \neq \emptyset$ **do**
  $V' := V' \cup \Delta$
  $\Delta := \{A \in V \setminus V';\ (\exists A \rightarrow \alpha) \in P$ mit $\alpha \in (V' \cup \Sigma)^*\}$
**od**

Induktion über die Länge der Ableitung. □

### Definition 66

$A \in V$ heißt nutzlos, falls es keine Ableitung

$$S \to^* w, \quad w \in \Sigma^*$$

gibt, in der $A$ vorkommt.

### Satz 67

*Die Menge der nutzlosen Variablen kann in Zeit $O(|V| \cdot s(G))$ bestimmt werden.*

**Beweis:**

Sei $V''$ die Menge der nicht nutzlosen Variablen.
Offensichtlich gilt: $V'' \subseteq V'$ ($V'$ aus dem vorigen Satz).
Falls $S \notin V'$, dann sind alle Variablen nutzlos.

Ansonsten:

$\Delta := \{S\}$; $V'' := \emptyset$;
**while** $\Delta \neq \emptyset$ **do**
  $V'' := V'' \cup \Delta$
  $\Delta := \{B \in V' \setminus V'';\ (\exists A \to \alpha B \beta) \in P$ mit $A \in V'',$
    $\alpha, \beta \in (V' \cup \Sigma)^*\}$
**od**

Induktion über Länge der Ableitung: Am Ende des Algorithmus ist $V''$ gleich der Menge der nicht nutzlosen Variablen. $\square$

**Bemerkung:** Alle nutzlosen Variablen und alle Produktionen, die nutzlose Variablen enthalten, können aus der Grammatik entfernt werden, ohne die erzeugte Sprache zu ändern.

## Korollar 68

*Für eine kontextfreie Grammatik $G$ kann in Zeit $O(|V| \cdot s(G))$ entschieden werden, ob $L(G) = \emptyset$.*

## Beweis:

$L(G) = \emptyset \Longleftrightarrow S \notin V''$ (bzw. $S \notin V'$) $\square$

## Satz 69

*Für eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ ohne nutzlose Variablen und in Chomsky-Normalform kann in linearer Zeit entschieden werden, ob*

$$|L(G)| < \infty\,.$$

## Beweis:

Definiere gerichteten Hilfsgraphen mit Knotenmenge $V$ und

$$\text{Kante } A \to B \iff (A \to BC) \text{ oder } (A \to CB) \in P\,.$$

$L(G)$ ist endlich $\iff$ dieser Graph enthält keinen Zyklus.

Verwende DFS, um in linearer Zeit festzustellen, ob der Graph Zyklen enthält. $\square$

## Satz 70

*Seien kontextfreie Grammatiken* $G_1 = (V_1, \Sigma_1, P_1, S_1)$ *und* $G_2 = (V_2, \Sigma_2, P_2, S_2)$ *gegeben. Dann können in linearer Zeit kontextfreie Grammatiken für*

1. $L(G_1) \cup L(G_2)$,
2. $L(G_1)L(G_2)$,
3. $(L(G_1))^*$

*konstruiert werden. Die Klasse der kontextfreien Sprachen ist also unter Vereinigung, Konkatenation und Kleene'scher Hülle abgeschlossen.*

## Beweis:

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $V_1 \cap V_2 = \emptyset$.

1. $V = V_1 \cup V_2 \cup \{S\}$; $S$ neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 | S_2\}$
2. $V = V_1 \cup V_2 \cup \{S\}$; $S$ neu
   $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$
3. $V = V_1 \cup \{S, S'\}$; $S, S'$ neu
   $P = P_1 \cup \{S \to S' | \epsilon, S' \to S_1 S' | S_1\}$

Falls $\epsilon \in L(G_1)$ oder $\epsilon \in L(G_2)$, sind noch Korrekturen vorzunehmen, die hier als Übungsaufgabe überlassen bleiben. $\square$

## Satz 71

*Die Klasse der kontextfreien Sprachen ist nicht abgeschlossen unter Durchschnitt oder Komplement.*

## Beweis:

Es genügt zu zeigen (wegen de Morgan (1806–1871)): nicht abgeschlossen unter Durchschnitt.

$$L_1 := \{a^i b^i c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$

$$L_2 := \{a^i b^j c^j;\ i, j \geq 0\} \text{ ist kontextfrei}$$

$$L_1 \cap L_2 = \{a^i b^i c^i;\ i \geq 0\} \text{ ist nicht kontextfrei}$$

□

## Satz 72

*Die Klasse der kontextfreien Sprachen ist abgeschlossen gegenüber Substitution (mit kontextfreien Mengen).*

## Beweis:

Ersetze jedes Terminal $a$ durch ein neues Nichtterminal $S_a$ und füge zu den Produktionen $P$ für jedes Terminal $a$ die Produktionen einer kontextfreien Grammatik $G_a = (V_a, \Sigma, P_a, S_a)$ hinzu. Forme die so erhaltene Grammatik in eine äquivalente Chomsky-2-Grammatik um. □

## 4.6 Greibach-Normalform

### Definition 73

Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik. $G$ ist in Greibach-Normalform (benannt nach Sheila Greibach (UCLA)), falls jede Produktion $\neq S \rightarrow \epsilon$ von der Form

$$A \rightarrow a\alpha \text{ mit } a \in \Sigma,\ \alpha \in V^*$$

ist.

### Lemma 74

*Sei $G = (V, \Sigma, P, S)$ kontextfrei, $(A \rightarrow \alpha_1 B \alpha_2) \in P$, und sei $B \rightarrow \beta_1 | \beta_2 | \ldots | \beta_r$ die Menge der $B$-Produktionen (also die Menge der Produktionen mit $B$ auf der linken Seite). Ersetzt man $A \rightarrow \alpha_1 B \alpha_2$ durch $A \rightarrow \alpha_1 \beta_1 \alpha_2 | \alpha_1 \beta_2 \alpha_2 | \ldots | \alpha_1 \beta_r \alpha_2$, so ändert sich die von der Grammatik erzeugte Sprache nicht.*

## Lemma 75

*Sei* $G = (V, \Sigma, P, S)$ *kontextfrei, sei* $A \rightarrow A\alpha_1 | A\alpha_2 | \ldots | A\alpha_r$ *die Menge der linksrekursiven* $A$*-Produktionen (alle* $\alpha_i \neq \epsilon$*, die Produktion* $A \rightarrow A$ *kommt o.B.d.A. nicht vor), und seien* $A \rightarrow \beta_1 | \beta_2 | \ldots | \beta_s$ *die restlichen* $A$*-Produktionen (ebenfalls alle* $\beta_i \neq \epsilon$*).*

*Ersetzen wir alle* $A$*-Produktionen durch*

$$
\begin{aligned}
A &\rightarrow \beta_1 | \ldots | \beta_s | \beta_1 A' | \ldots | \beta_s A' \\
A' &\rightarrow \alpha_1 | \ldots | \alpha_r | \alpha_1 A' | \ldots | \alpha_r A' ,
\end{aligned}
$$

*wobei* $A'$ *ein neues Nichtterminal ist, so ändert sich die Sprache nicht, und die neue Grammatik enthält keine linksrekursive* $A$*-Produktion mehr.*

**Beweis:**

Von $A$ lassen sich vor der Transformation alle Zeichenreihen der Form

$$(\beta_1|\beta_2|\dots|\beta_s)(\alpha_1|\alpha_2|\dots|\alpha_r)^+$$

ableiten.

Dies ist auch nach der Transformation der Fall. Während vor der Transformation alle Zeichenreihen der obigen Form von rechts her aufgebaut werden, werden sie danach von links nach rechts erzeugt.

Die Umkehrung gilt ebenso. $\square$

### Satz 76

*Zu jeder kontextfreien Grammatik $G$ gibt es eine äquivalente Grammatik in Greibach-Normalform.*

Beweis:

Sei o.B.d.A. $G = (V, \Sigma, P, S)$ mit $V = \{A_1, \ldots, A_m\}$ in Chomsky-Normalform und enthalte keine nutzlosen Variablen.

Bemerkung: Im folgenden Algorithmus werden ggf neue Variablen hinzugefügt, die Programmvariable $m$ ändert sich dadurch entsprechend!

Beweis:

**for** $k = 1, \ldots, m$ **do**
  **for** $j = 1, \ldots, k-1$ **do**
    **for all** $(A_k \to A_j\alpha) \in P$ **do**
      ersetze die Produktion gemäß der Konstruktion
      in Lemma 74
    **od**
  **od**
  **co** die rechte Seite keiner $A_{k'}$-Produktion, $k' < k$, beginnt
    nun noch mit einer Variablen $A_j$, $j < k$ **oc**

  ersetze alle linksrekursiven $A_k$-Produktionen gemäß der
  Konstruktion in Lemma 75
  **co** die rechte Seite keiner $A_{k'}$-Produktion, $k' \leq k$, beginnt
    nun noch mit einer Variablen $A_j$, $j \leq k$ **oc**
**od**

□

## Korollar 77

*Sei $G$ eine kontextfreie Grammatik. Es gibt einen Algorithmus, der eine zu $G$ äquivalente Grammatik in Greibach-Normalform konstruiert, deren rechte Seiten jeweils höchstens zwei Variablen enthalten.*

## Beweis:

Klar! $\square$