



Vorlesungsmitschrift:  
**“Formale Sprachen”**  
Prof. Dr. J. Duske  
Institut für Informatik, Hannover  
(WS 95/96)

26. Februar 1997

ge-TeX-ed von Sven Meyer  
(email: [sm@stochastik.uni-hannover.de](mailto:sm@stochastik.uni-hannover.de))  
(WWW: [www.stochastik.uni-hannover.de/~sm](http://www.stochastik.uni-hannover.de/~sm))

Für diejenigen, die sich die Mühe machen, dieses Skript zu lesen und dabei auf Tippfehler stoßen:  
Bitte schickt mir, nachdem Ihr auf meiner Homepage nachgesehen und Euch vergewissert habt, daß dieser Fehler  
noch immer existiert, eine email, damit ich ihn berichtigen kann.

Mit bestem Dank,    Sven.

# Inhaltsverzeichnis

|           |   |           |
|-----------|---|-----------|
| <b>1</b>  | <b>Grundlegende Definitionen</b>                              | <b>1</b>  |
| <b>2</b>  | <b>Rechtslineare Sprachen und endliche Automaten</b>          | <b>3</b>  |
| <b>3</b>  | <b>Rechtslineare Sprachen und reguläre Mengen</b>             | <b>7</b>  |
| <b>4</b>  | <b>Das Iterationslemma für reguläre Sprachen</b>              | <b>10</b> |
| <b>5</b>  | <b>Abschlußigenschaften regulärer Sprachen</b>                | <b>11</b> |
| <b>6</b>  | <b>Kontextfreie Grammatiken und Sprachen</b>                  | <b>12</b> |
| <b>7</b>  | <b>Transformation kontextfreier Grammatiken</b>               | <b>14</b> |
| <b>8</b>  | <b>Chomsky–Normalform, Homomorphie kontextfreier Sprachen</b> | <b>18</b> |
| <b>9</b>  | <b>Der Iterationssatz für kontextfreie Sprachen</b>           | <b>21</b> |
| <b>10</b> | <b>Rekursivität und die Greibach–Normalform</b>               | <b>23</b> |
| <b>11</b> | <b>Kontextfreie Sprachen und Kellerautomaten</b>              | <b>27</b> |
| <b>12</b> | <b>Abgeschlossenheitseigenschaften kontextfreier Sprachen</b> | <b>30</b> |
| <b>13</b> | <b>Indizierte Grammatiken und Ableitungsmodi</b>              | <b>32</b> |
| 13.1      | Ableitung bzgl. indizierter Grammatiken . . . . .             | 32        |
| 13.2      | Festlegung der Ableitungsmodi . . . . .                       | 33        |
| <b>14</b> | <b>Eigenschaften V–indizierter Sprachen</b>                   | <b>35</b> |
| <b>15</b> | <b>Lineare Sprache</b>  | <b>41</b> |
| <b>16</b> | <b>Selbsteinbettende Grammatiken</b>                          | <b>43</b> |
| <b>17</b> | <b>Semilineare Mengen und der Satz von Parikh</b>             | <b>45</b> |
| <b>18</b> | <b>Literaturliste</b>   | <b>48</b> |

# Kapitel 1

## Grundlegende Definitionen

$x_i$ : Buchstaben, z.B.:  $X = \{a, b, c\}$

Eine Folge  $x_{i_1}x_{i_2}\dots x_{i_k}$ ,  $k \geq 1$  von Buchstaben aus  $X$  heißt **nichtleeres Wort über  $X$** .

Sei  $w = x_{i_1}\dots x_{i_k}$  ein Wort, dann:

–  $k$  ist die Länge von  $w$ :  $|w| = k$

–  $w^R = x_{i_k}\dots x_{i_1}$  ist das Spiegelbild von  $w$  (reverse)

$X^+$ : Menge aller nichtleeren Wörter über  $X$ .

Seien  $w = x_{i_1}\dots x_{i_k}$  und  $v = y_{j_1}\dots y_{j_l}$  aus  $X^+$ .

Hintereinanderschreiben von  $w$  und  $v$ :  $w \circ v = x_{i_1}\dots x_{i_k}y_{j_1}\dots y_{j_l} \in X^+$ . (**Produkt** von  $w$  und  $v$ ,  $\circ$  ist assoziativ)

$(X^+, \circ)$ : Halbgruppe (freie Halbgruppe ber  $X$ )

Hinzufügen eines Einselementes  $e$  zu  $X^+$ :  $X^* = X^+ \cup \{e\}$

– Erweiterung von  $\circ$  auf  $X^*$ :  $\forall w \in X^* : w \circ e = e \circ w = w$ .

$(X^*, \circ)$ : **Monoid** (freies Monoid ber  $X \cong$  HG mit 1-Element)

$e$ : **leeres Wort**

Festlegung:  $|e| = 0$   
 $e^R = e$

Es gilt:  $|w \circ v| = |w| + |v|$

Es sei:  $X^+ := (X^+, \circ)$ ;  $X^* := (X^*, \circ)$ ;  $wv := w \circ v$

**Definition 1.1** (Sprache)

Sei  $X$  ein Alphabet. Eine Teilmenge  $L \subseteq X^*$  heißt **Sprache über  $X$** .

**Beispiel 1.2**  $X = \{a, b, c\}$

$L = \{a^n b^n | n \geq 0\}$  ist Sprache über  $X$ . ( $a^0 = b^0 = e$ )

**Definition 1.3** (Produktionssystem)

Ein **Produktionssystem** ist ein Paar  $(A, P)$ . Hierbei ist  $A$  ein Alphabet und  $P$  eine endliche Menge geordneter Wortpaare über  $A$ .

Schreibweise:  $(\alpha, \beta) \in P$  wird  $\alpha \rightarrow \beta$  geschrieben.

Benutzung des Produktionssystems:

— Seien  $u, v \in A^*$ . Dann  $u \Rightarrow v$  g.d.w.  $\exists u_1, u', v_1 \in A^*$  mit  $u = u_1 u' u_2$  und  $v = u_1 v_1 u_2$  und  $u' \rightarrow v_1 \in P$ .

— Sei  $k \geq 0$ .  $u \xRightarrow{k} v$  g.d.w.  $\exists u_0, u_1, \dots, u_k \in A^*$  und  $u = u_0$ ,  $v = u_k$  und  $u_{i-1} \Rightarrow u_i$  für  $i \in [1 : k]$ .

Sprechweise:  $v$  läßt sich in  $k$  Schritten aus  $u$  ableiten.

$u_0, u_1, \dots, u_k$  heißt **Ableitung von  $v$  aus  $u$** .

Schreibweise:  $u_0 \Rightarrow u_1 \Rightarrow \dots \Rightarrow u_k$ .

$u \xRightarrow{*} v$  g.d.w.  $\exists k \geq 0$  mit  $u \xRightarrow{k} v$ .

$u \xRightarrow{+} v$  g.d.w.  $\exists k > 0$  mit  $u \xRightarrow{k} v$ .

**Definition 1.4** (Grammatik)

Eine **Grammatik** ist ein 4-Tupel  $G = (V, X, S, P)$  mit:

1.  $V$  ist Alphabet (Variablenmenge, Nicht-Terminale)
2.  $X$  ist Alphabet (Konstantenmenge, Terminale)
3.  $V \cap X = \emptyset$
4.  $S \in V$  (Startsymbol oder Startvariable)
5.  $P$  ist endliche Menge von Paaren  $(u, v)$  mit  $u \in (V \cup X)^* V (V \cup X)^*$  und  $v \in (V \cup X)^*$  (Produktionen von  $G$ )

Es gilt:  $G$  legt ein Produktionssystem  $(A, P)$  mit  $A = V \cup X$  fest.  $\Rightarrow, \xRightarrow{k}, \xRightarrow{*}, \xRightarrow{+}$  s.o.

$\alpha \in (V \cup X)^*$  heißt **Satzform**, wenn gilt:  $S \xRightarrow{*} \alpha$ .

$w \in X^*$  heißt **Satz**, wenn gilt:  $S \xRightarrow{*} w$ .

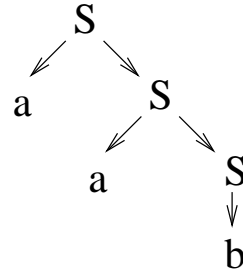
Dann:  $L(G) = \{w | w \in X^* \text{ und } S \xRightarrow{*} w\}$

### Beispiel 1.5

$X = \{a, b, c\}$ ,  $L = \{a^n b \mid n \geq 0\}$  ist Sprache über  $X$ .

Es ist  $L = L(G)$  mit  $G = (\{S\}, X, S, P)$  und  $P = \{S \rightarrow b, S \rightarrow aS\}$ .

Ableitung:  $S \Rightarrow aS \Rightarrow a^2 S \Rightarrow a^2 b$ .



### Beispiel 1.6

$X = \{a, b, c\}$ ,  $L = \{a^n b^n \mid n \geq 0\}$

Es ist  $L = L(G)$  mit  $G = (\{S\}, X, S, P)$  und  $P = \{S \rightarrow e, S \rightarrow aSb\}$

### Beispiel 1.7

$X = \{a, b, c\}$ ,  $L = \{a^n b^n c^n \mid n \geq 0\}$ .

Es ist  $L = L(G)$  mit  $G = (V, X, S, P)$ ,  $V = \{S, A, B\}$  und  $P = \{S \rightarrow e, S \rightarrow abc, S \rightarrow aAbc, Ab \rightarrow bA, Ac \rightarrow Bbcc, bB \rightarrow Bb, aB \rightarrow aaA, aB \rightarrow aa\}$

$S \Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \Rightarrow aBb^2c^2 \Rightarrow aaAb^2c^2 \xRightarrow{2} a^2b^2Ac^2 \Rightarrow a^2b^2Bbc^3 \xRightarrow{2} a^2Bb^3c^3 \Rightarrow a^3b^3c^3$

### Beispiel 1.8

$X = \{a, b, c\}$ ,  $L = \{ww^R \mid w \in \{a, b\}^*\}$ .

Es ist  $L = L(G)$  mit  $G = (\{S\}, X, S, P)$  und  $P = \{S \rightarrow e, S \rightarrow aSa, S \rightarrow bSb\}$ .

### Beispiel 1.9

$X = \{a, b, c\}$ ,  $L = \{ww \mid w \in \{a, b\}^*\}$ .

Es ist  $L = L(G)$  mit  $G = (V, X, S, P)$ ,  $V = (S, A, B, C, D)$  und

$P = \{S \rightarrow CD, C \rightarrow aCA, C \rightarrow bCB, AD \rightarrow aD, BD \rightarrow bD, Aa \rightarrow aA, Ab \rightarrow bA, Ba \rightarrow aB, Bb \rightarrow bB, C \rightarrow e, D \rightarrow e\}$

Bsp. für eine Ableitung:  $S \Rightarrow CD \Rightarrow aCAD \Rightarrow abCBAD \Rightarrow abBAD \Rightarrow abBaD \Rightarrow abaBD \Rightarrow ababD \Rightarrow abab$ .

# Kapitel 2

## Rechtslineare Sprachen und endliche Automaten

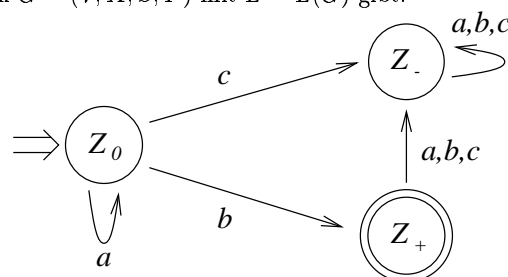
### Definition 2.1 (rechtslinear)

Eine Grammatik  $G = (V, X, S, P)$  heißt **rechtslinear**, wenn jede Produktion von der Art  $A \rightarrow aB$  oder  $A \rightarrow b$  mit  $A, B \in V, a \in X, b \in X \cup \{e\}$  ist.

Eine Sprache  $L \subseteq X^*$  heißt **rechtslinear**, wenn es eine rechtslineare Grammatik  $G = (V, X, S, P)$  mit  $L = L(G)$  gibt.

$L = \{a^n b | n \geq 0\}$  ist rechtslinear.

Gesucht: Vorrichtung, in die Wörter über  $X = \{a, b, c\}$  eingegeben werden können und die zwischen den Wörtern aus  $L$  und denjenigen aus  $X^* \setminus L$  unterscheidet.

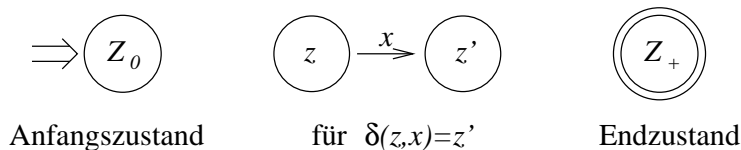


### Definition 2.2 (DFA)

Ein **deterministischer, endlicher Automat** ist ein 5-Tupel  $A = (Z, X, \delta, z_0, F)$ . Hierbei ist:

1.  $Z$  eine endliche Menge (Zustandsmenge von  $A$ )
2.  $X$  ein Alphabet (Eingabealphabet)
3.  $\delta : Z \times X \rightarrow Z$  (Überföhrungsfunktion)
4.  $z_0 \in Z$  (Anfangszustand)
5.  $F \subseteq Z$  (Menge der Endzustände)

### Bemerkung 2.3 Angabe von $A$ durch ein Zustandsüberföhrungsdiagramm:



Allgemein soll folgende Situation vorliegen:

- Auf einen Eingabeband stehen die Buchstaben eines Eingabewortes  $w = a_1 a_2 \dots a_n, a_i \in X, i \in [1 : n], n \geq 0$ .
- Ein Lesekopf soll die Buchstaben des Bandes von links nach rechts lesen können.
- Erweiterung von  $\delta : Z \times X \rightarrow Z$  zu  $\bar{\delta} : Z \times X^* \rightarrow Z$  folgendermaßen:

1.  $\forall z \in Z : \bar{\delta}(z, e) = z$
2.  $\forall z \in Z, \forall u \in X^*, \forall x \in X : \bar{\delta}(z, ux) = \delta(\bar{\delta}(z, u), x)$

Es gilt:  $\forall z \in Z, \forall x \in X : \bar{\delta}(z, x) = \delta(\bar{\delta}(z, e), x) = \delta(z, x)$   
 $\forall z \in Z, \forall u, v \in X^* : \bar{\delta}(z, uv) = \bar{\delta}(\bar{\delta}(z, u), v)$

### Definition 2.4 (Sprache eines DFA)

$A = (Z, X, \delta, z_0, F)$  sei ein deterministischer, endlicher Automat. Die **von  $A$  akzeptierte Sprache** ist die Wortmenge  $L(A) = \{w | w \in X^* \text{ und } \bar{\delta}(z_0, w) \in F\}$ .

### Bemerkung 2.5 $e \in L(A) \iff z_0 \in F$ .

### Definition 2.6 (NFA)

Ein **nichtdeterministischer, endlicher Automat** ist ein 5-Tupel  $B = (Z, X, \beta, z_0, F)$ . Hierbei sind  $Z, X, z_0, F$  wie in 2.2 und  $\beta : Z \times X \rightarrow \mathbb{P}(Z)$  ist Überföhrungsfunktion von  $B$ .

Interpretation: Ist  $\beta(z, x) = \{z_1, \dots, z_k\}$ , dann: Ausgehend von dem Zustand  $z$  und dem Einlesen von  $x$  kann einer der Zustände  $z_i, i \in [1 : k]$  angenommen werden ( $k \geq 0$ ).

Erweiterung von  $\beta : Z \times X \rightarrow \mathbb{P}(Z)$  zu einer Funktion  $\bar{\beta} : \mathbb{P}(Z) \times X^* \rightarrow \mathbb{P}(Z)$  durch:

1.  $\forall z \in Z : \bar{\beta}(\{z\}, e) = \{z\}$
2.  $\forall z \in Z, u \in X^*, v \in X : \bar{\beta}(\{z\}, uv) = \bigcup_{z' \in \bar{\beta}(\{z\}, u)} \beta(z', v)$
3.  $\forall Y \in \mathbb{P}(Z), u \in X^* : \bar{\beta}(Y, u) = \bigcup_{z \in Y} \bar{\beta}(\{z\}, u)$

Es gilt:  $\forall z \in Z, \forall x \in X : \bar{\beta}(\{z\}, x) = \beta(z, x)$   
 $\forall Y \in \mathbb{P}(Z), \forall u, v \in X^* : \bar{\beta}(Y, uv) = \bar{\beta}(\bar{\beta}(Y, u), v)$

Interpretation:  $\bar{\beta}(Y, v)$  ist die Menge aller Zustände, die man erreichen kann, wenn man in einem Zustand aus  $Y$  startet und  $v$  einliest.

### Definition 2.7 (Sprache eines NFA)

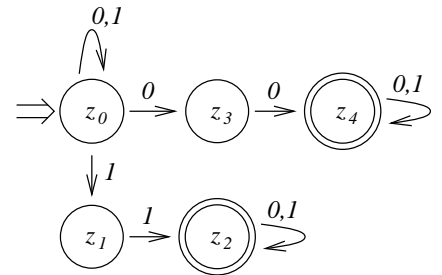
$B = (V, X, \beta, z_0, F)$  sei ein nichtdeterministischer, endlicher Automat. Die von  $B$  akzeptierte Sprache ist die Wortmenge  $L(B) = \{w | w \in X^* \text{ und } \bar{\beta}(\{z_0\}, w) \cap F \neq \emptyset\}$ .

**Bemerkung 2.8**  $e \in L(B) \iff z_0 \in F$ .

### Beispiel 2.9

$B = (Z, X, \beta, z_0, F)$  mit  $X = \{0, 1\}$ ,  
 $Z = \{z_i | i \in [0 : 4]\}$ ,  $F = \{z_2, z_4\}$  und

|                                |                                |
|--------------------------------|--------------------------------|
| $\beta(z_0, 0) = \{z_0, z_3\}$ | $\beta(z_0, 1) = \{z_0, z_1\}$ |
| $\beta(z_1, 0) = \emptyset$    | $\beta(z_1, 1) = \{z_2\}$      |
| $\beta(z_2, 0) = \{z_2\}$      | $\beta(z_2, 1) = \{z_2\}$      |
| $\beta(z_3, 0) = \{z_4\}$      | $\beta(z_3, 1) = \emptyset$    |
| $\beta(z_4, 0) = \{z_4\}$      | $\beta(z_4, 1) = \{z_4\}$      |



$L(B) = \{w | w \in \{0, 1\}^* \text{ und } w \text{ enthalt als Teilwort } 00 \text{ oder } 11\}$ .

Vereinbarung: endlicher Automat = endlicher Akzeptor

DFA = deterministischer, endlicher Automat (deterministic finite automaton)  
 NFA = nichtdeterministischer, endlicher Automat (non-deterministic finite automaton)

### Satz 2.10 ( $L(\text{NFA}) \subseteq L(\text{DFA})$ )

$L \subseteq X^*$  werde von einem NFA  $B = (Z, X, \beta, z_0, F)$  erkannt. Dann kann man einen DFA  $A = (Z', X, \delta, z'_0, F')$  konstruieren fur den  $L = L(A)$  gilt.

Bew.: Setze  $Z' = \mathbb{P}(Z)$ ,  $z'_0 = \{z_0\}$ ,  $F' = \{Y | Y \in \mathbb{P}(Z) \text{ und } Y \cap F \neq \emptyset\}$ .

$\forall Y \in \mathbb{P}(Z), x \in X : \delta(Y, x) = \bar{\beta}(Y, x)$

Es gilt folgendes:  $\forall u \in X^* : \bar{\delta}(z'_0, u) = \bar{\beta}(\{z_0\}, u)$

Induktion uber  $n = |u|$ .

$n = 0$ :  $u = e$  und  $\bar{\delta}(z'_0, e) = z'_0 = \{z_0\} = \bar{\beta}(\{z_0\}, e)$

$n \rightarrow n + 1$ :  $u = u'x, |u'| = n, x \in X$ .

$\bar{\delta}(z'_0, u) = \bar{\delta}(z'_0, u'x) = \delta(\bar{\delta}(z'_0, u'), x) = \delta(\bar{\beta}(\{z_0\}, u'), x) = \bar{\beta}(\bar{\beta}(\{z_0\}, u'), x) = \bar{\beta}(\{z_0\}, u'x) = \bar{\beta}(\{z_0\}, u)$

Damit folgt:  $w \in L(A) \iff \bar{\delta}(z'_0, w) \in F' \iff \bar{\beta}(\{z_0\}, w) \in F' \iff \bar{\beta}(\{z_0\}, w) \cap F \neq \emptyset \iff w \in L(B)$  □

### Satz 2.11 ( $L(\text{NFA}) = L(\text{DFA})$ )

Eine Sprache  $L \subseteq X^*$  wird genau dann von einem NFA akzeptiert, wenn sie von einem DFA akzeptiert wird.

### Satz 2.12 (Rechtslineare Sprachen werden von NFAs erkannt)

Es gelte  $L = L(G)$  fur eine rechtslineare Grammatik  $G = (V, X, S, P)$ . Dann kann man einen NFA  $B = (Z, X, \beta, z_0, F)$  mit  $L(B) = L(G)$  konstruieren.

Bew.: Setze  $Z = V \cup \{A\}$ ,  $A$  sei Hilfselement, das nicht in  $V$  vorkommt.

$z_0 = S$

$B_1 \rightarrow e, B_2 \rightarrow e, \dots, B_k \rightarrow e$  seien die  $e$ -Produktionen von  $G$ .

$F = \{B_1, B_2, \dots, B_k, A\}$

Setze  $\beta$  folgendermaen:

$\forall B, C \in V$  und  $\forall a \in X : A \in \beta(B, a)$  falls  $B \rightarrow a \in P$

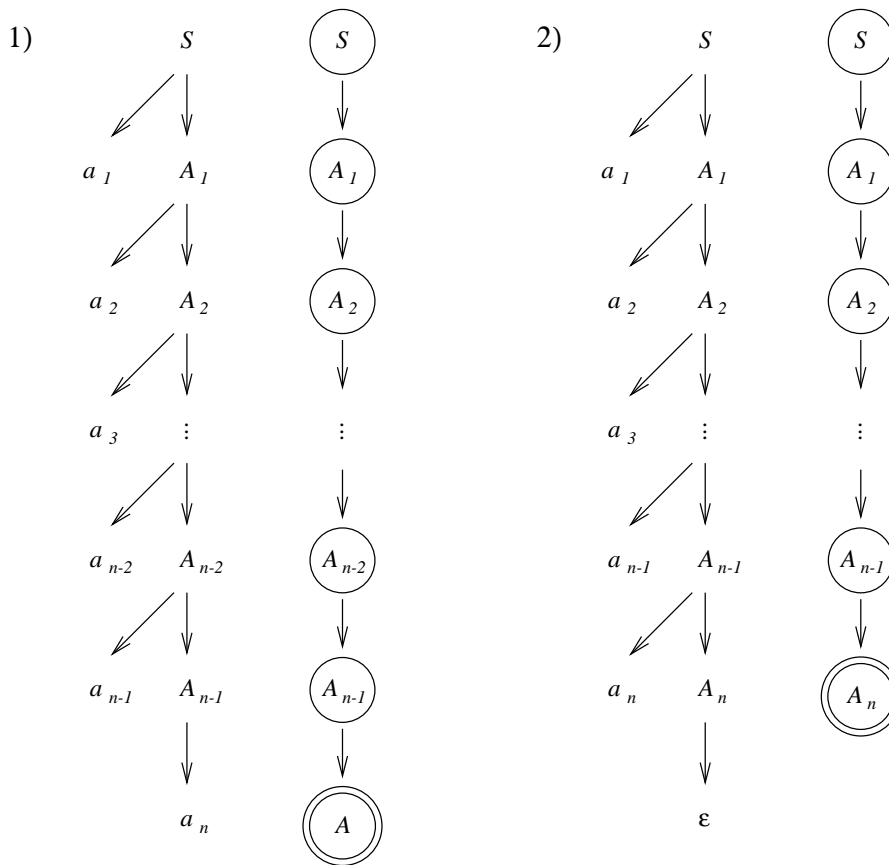
$C \in \beta(B, a)$  falls  $B \rightarrow aC \in P$

$\forall a \in X : \beta(A, a) = \emptyset$

a)  $L(G) \subseteq L(B)$ , dazu:

Sei  $e \in L(G)$ , dann ist  $S \rightarrow e \in P$ , also ist  $S \in F$ , also ist  $e \in L(B)$ .

Sei  $w = a_1 a_2 \dots a_n \in L(G)$ ,  $n \geq 1$ . Dann:



In 1) gilt:  $A_{n-1}$  ist aus  $\overline{\beta}(\{S\}, a_1 \dots a_{n-1})$  und  $A \in \beta(A_{n-1}, a_n)$ .

In 2) gilt:  $A_n$  ist bereits aus  $\overline{\beta}(\{S\}, a_1 \dots a_n)$

also ist  $w \in L(B)$ .

b)  $L(B) \subseteq L(G)$  entsprechend. □

### Lemma 2.13

Ist  $G = (V, X, S, P)$  eine rechtslineare Grammatik, für die  $e \notin L(G)$  gilt, dann gibt es eine rechtslineare Grammatik  $G' = (V', X, S', P')$  mit  $L(G') = L(G) \cup \{e\}$ .

Bew.:  $S'$  sei ein neues Zeichen,  $S' \notin V \cup X$ .

$V' = V \cup \{S'\}$

$P' = P \cup \{S' \rightarrow e\} \cup \{S' \rightarrow \alpha \mid S \rightarrow \alpha \in P\}$  □

### Satz 2.14 (Sprachen von DFAs sind rechtslinear)

Es gelte  $L = L(A)$  für einen DFA  $A = (Z, X, \delta, z_0, F)$ . Dann kann man eine rechtslineare Grammatik  $G = (V, X, S, P)$  mit  $L(G) = L(A)$  konstruieren.

Bew.: Setze  $G$  mit  $V = Z$  und  $S = z_0$  und

$B \rightarrow aC \in P$ , falls  $\delta(B, a) = C$

$B \rightarrow a \in P$ , falls  $\delta(B, a) \in F$

Es gilt  $L(G) = L(A) \setminus \{e\}$ .

Ist  $e \notin L(A)$ , dann ist  $L(G) = L(A)$

Ist  $e \in L(A)$ , dann ist  $L(G) = L(A) \cup \{e\} = L(G')$  für eine geeignete rechtslineare Grammatik  $G'$ . □

### Beispiel 2.15

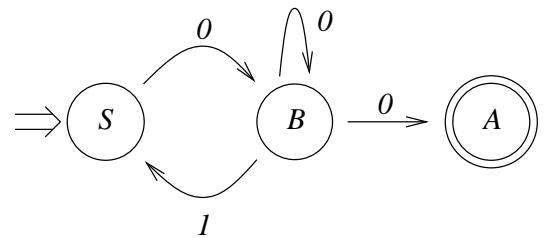
$G = (\{S, B\}, \{0, 1\}, S, P)$  mit  $P = \{S \rightarrow 0B, B \rightarrow 0B, B \rightarrow 1S, B \rightarrow 0\}$

Konstruktion eines NFA's  $B$ :

$B = (\{S, B, A\}, \{0, 1\}, \beta, S, \{A\})$

$\beta$ :

|                           |                           |
|---------------------------|---------------------------|
| $\beta(S, 0) = \{B\}$     | $\beta(S, 1) = \emptyset$ |
| $\beta(B, 0) = \{B, A\}$  | $\beta(B, 1) = \{S\}$     |
| $\beta(A, 0) = \emptyset$ | $\beta(A, 1) = \emptyset$ |



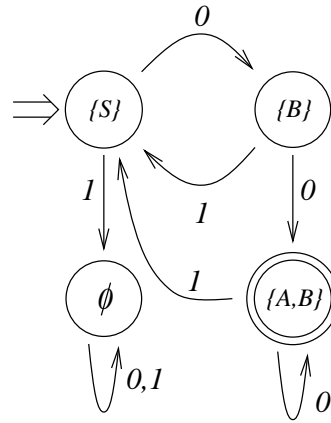
Konstruktion eines DFA's  $\tilde{A}$ :

$\tilde{A} = (Z, \{0, 1\}, \delta, \{S\}, F)$  aus  $B$  mit  $L(\tilde{A}) = L(B)$ .

$Z = \{\emptyset, \{S\}, \{A\}, \{B\}, \{A, S\}, \{A, B\}, \{B, S\}, \{A, B, S\}\}$ .

$F = \{\{A\}, \{A, S\}, \{A, B\}, \{A, B, S\}\}$ .

$\delta(\{S\}, 0) = \{B\}$        $\delta(\{S\}, 1) = \emptyset$        $\delta(\emptyset, 0) = \emptyset$   
 $\delta(\{B\}, 0) = \{A, B\}$      $\delta(\{B\}, 1) = \{S\}$        $\delta(\emptyset, 1) = \emptyset$   
 $\delta(\{A, B\}, 0) = \{A, B\}$     $\delta(\{A, B\}, 1) = \{S\}$



# Kapitel 3

## Rechtslineare Sprachen und reguläre Mengen

Sei  $X$  ein Alphabet. Betrachte folgende Operationen auf  $\mathbb{P}(X^*)$ :  
Seien  $L, L_1, L_2 \in \mathbb{P}(X^*)$ , dann:

1.  $L_1 \cup L_2 = \{w | w \in L_1 \text{ oder } w \in L_2\}$  (Summe von  $L_1$  und  $L_2$ )
  2.  $L_1 \cdot L_2 = \{uv | u \in L_1 \text{ und } v \in L_2\}$  (Produkt von  $L_1$  und  $L_2$ )
  3.  $L^* = \bigcup_{k=0}^{\infty} L^k$  (Iteration von  $L$ )
- hierbei ist  $L^0 := \{e\}$  und  $L^k := L^{k-1} \cdot L$

**Bemerkung 3.1**  $\emptyset^* = \bigcup_{k=0}^{\infty} \emptyset^k = \{e\}$ .

Sei  $X = \{x_1, \dots, x_n\}$ , dann ist  $X^* = \{x_1, \dots, x_n\}^*$ .

$\emptyset, \{x_1\}, \dots, \{x_n\}$  heißen die **elementaren Sprachen über  $X$** .

**Definition 3.2** (regulär)

Eine Sprache/Menge  $L \in \mathbb{P}(X^*)$  heißt regulär, wenn sie, ausgehend von den elementaren Sprachen durch endlichmalige Anwendung der Operationen  $\cup, \cdot, ^*$  erhalten werden kann.

**Satz 3.3** ( $L$ (DFA) ist regulär)

Gilt  $L = L(A)$  für einen DFA  $A = (Z, X, \delta, z_0, F)$ , dann ist  $L$  eine reguläre Menge (Sprache).

Bew.: Wegen  $L(A) = \bigcup_{z \in F} \{u | \bar{\delta}(z_0, u) = z\}$  kann  $F$  einelementig angenommen werden.

Setze weiter o.B.d.A.  $Z = \{1, 2, \dots, m\}$  und  $z_0 = 1$ .

Sprechweise: Seien  $i, j \in Z$  und sei  $u = u_{i_1} u_{i_2} \dots u_{i_\alpha} \in X^*$  mit  $\bar{\delta}(i, u) = j$ .

Dann heißen  $\bar{\delta}(i, x_{i_1} \dots x_{i_\beta})$  mit  $\beta \in [1 : \alpha - 1]$  die bei der Überführung von  $i$  nach  $j$  durchlaufenen Zwischenzustände.

Für  $i, j \in [1 : m]$  und  $k \in [0 : m]$  setze man  $L_{ij}^{(k)} = \{u | \delta(i, u) = j \text{ und alle durchlaufenen Zwischenzustände sind aus } [1 : k]\}$ .

Speziell:  $k = 0$ :  $u \in L_{ij}^{(0)}$ . Es werden keine Zwischenzustände durchlaufen, also  $u = e$  und  $i = j$  oder  $u = x \in X$ .

Es genügt z.Z.:  $L_{1l}^{(m)}$  ist regulär für  $l \in [1 : m]$ .

Es wird gezeigt:  $\forall i, j \in [0 : m] \forall k \in [0 : m]$  ist  $L_{ij}^{(k)}$  regulär.

Dazu: (Induktion über  $k$ ):

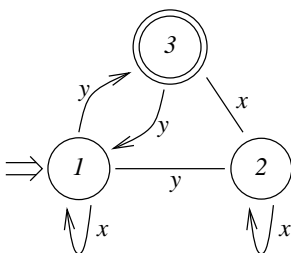
$k = 0$ : Jedes  $\alpha \in L_{ij}^{(0)}$  ist elementare Sprache oder  $e$ , also ist  $L_{ij}^{(0)}$  regulär.

$k > 0$ : Es gelte  $\forall i, j \in [1 : m]$  und  $l \in [0 : k - 1]$ , daß  $L_{ij}^{(k)}$  regulär ist.

Wegen  $L_{ij}^{(k)} = L_{ik}^{(k-1)} \cup L_{ij}^{(k-1)} (L_{kk}^{(k-1)})^* L_{kj}^{(k-1)}$  ist  $L_{ij}^{(k)}$  regulär. □

**Beispiel 3.4**

Sei  $A = (Z, X, \delta, z_0, F)$  folgendermaßen:



Um  $L_{13}^{(3)}$  zu berechnen, benötigen wir folgende Identitäten:  
 $\forall L, L_1, L_2, L_3 \subseteq X^*$  gilt:

- (1)  $L \cdot \{e\} = \{e\} \cdot L = L$
- (2)  $L_1 \cdot (L_2 \cup L_3) = L_1 \cdot L_2 \cup L_1 \cdot L_3$   
 $(L_2 \cup L_3) \cdot L_1 = L_2 \cdot L_1 \cup L_3 \cdot L_1$
- (3)  $L^* = \{e\} \cup L \cdot L^* = \{e\} \cup L^* \cdot L$

Es gilt:  $L_{13}^{(3)} = L_{13}^{(2)} \cup L_{13}^{(2)} \cdot (L_{33}^{(2)})^* \cdot L_{33}^{(2)} = L_{13}^{(2)} \cdot (\{e\} \cup (L_{33}^{(2)})^* \cdot L_{33}^{(2)}) = L_{13}^{(2)} \cdot (L_{33}^{(2)})^*$

Es müssen  $L_{13}^{(2)}$  und  $L_{33}^{(2)}$  bestimmt werden:

Bestimmung von  $L_{13}^{(2)} = L_{13}^{(1)} \cup L_{12}^{(1)} \cdot (L_{22}^{(1)})^* \cdot L_{23}^{(1)}$ :

Es ist  $L_{13}^{(1)} = L_{13}^{(0)} \cup L_{11}^{(0)} \cdot (L_{11}^{(0)})^* \cdot L_{13}^{(0)} = (\{e\} \cup L_{11}^{(0)} \cdot (L_{11}^{(0)})^*) \cdot L_{13}^{(0)} = (L_{11}^{(0)})^* \cdot L_{13}^{(0)} = x^* \cdot y$

$L_{12}^{(1)} = L_{11}^{(0)} \cdot (L_{11}^{(0)})^* \cdot L_{12}^{(0)} \cup L_{12}^{(0)} = (L_{11}^{(0)} \cdot (L_{11}^{(0)})^* \cup \{e\}) \cdot L_{12}^{(0)} = (L_{11}^{(0)})^* \cdot L_{12}^{(0)} = x^* \cdot \emptyset = \emptyset$

Also ist  $L_{13}^{(2)} = x^* \cdot y$ .

Bestimmung von  $L_{33}^{(2)} = L_{33}^{(1)} \cup L_{32}^{(1)} \cdot (L_{22}^{(1)})^* \cdot L_{23}^{(1)}$ , dazu:

$L_{33}^{(1)} = L_{33}^{(0)} \cup L_{31}^{(0)} \cdot (L_{11}^{(0)})^* \cdot L_{13}^{(0)} = \{e\} \cup yx^*y$

$L_{32}^{(1)} = L_{32}^{(0)} \cup L_{31}^{(0)} \cdot (L_{11}^{(0)})^* \cdot L_{12}^{(0)} = x \cup yx^*\emptyset = x$

$L_{22}^{(1)} = L_{22}^{(0)} \cup L_{21}^{(0)} \cdot (L_{11}^{(0)})^* \cdot L_{12}^{(0)} = \{e, x\} \cup yx^*\emptyset = \{e, x\}$

$L_{23}^{(1)} = L_{23}^{(0)} \cup L_{21}^{(0)} \cdot (L_{11}^{(0)})^* \cdot L_{13}^{(0)} = \emptyset \cup yx^*y = yx^*y$

$\Rightarrow L_{33}^{(2)} = e \cup yx^*y \cup x \cdot \{e, x\}^* yx^*y = e \cup yx^*y \cup xx^*yx^*y = e \cup (e \cup xx^*)yx^*y = e \cup x^*yx^*y$

$\Rightarrow L_{13}^{(2)} \cdot (L_{33}^{(2)})^* = x^*y(e \cup x^*yx^*y)^* = x^*y(x^*yx^*y)^*$

**Satz 3.5** ( $L$  regulär  $\Rightarrow L = L(\text{DFA})$ )

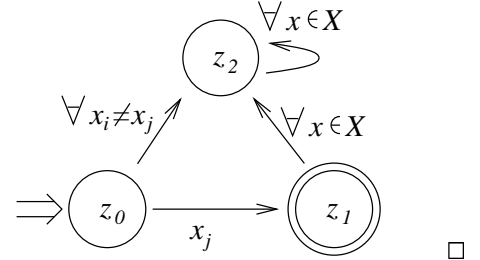
Ist  $L \subseteq X^*$  regulär, dann gibt es einen DFA  $A = (Z, X, \delta, z_0, F)$  mit  $L = L(A)$ .

**Lemma 3.6**

Jede elementare Sprache über  $X$  wird von einem DFA erkannt.

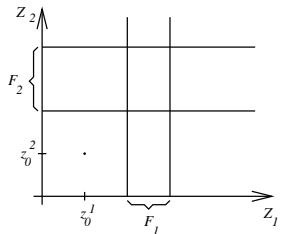
Bew.: Für  $L = \emptyset$  wähle einen DFA, der keine Endzustände besitzt.

Ist  $L = \{x_j\}$ , dann wähle angegebenen DFA:



**Lemma 3.7**

Können  $L_1 \subseteq X^*$  und  $L_2 \subseteq X^*$  von DFA's erkannt werden, so auch  $L_1 \cup L_2$ .



Bew.:  $L_1$  werde von  $A_1 = (Z_1, X, \delta_1, z_0^1, F_1)$  und  $L_2$  werde von  $A_2 = (Z_2, X, \delta_2, z_0^2, F_2)$  akzeptiert.

Setze  $A_3 = (Z_1 \times Z_2, X, \delta_3, (z_0^1, z_0^2), F_3)$  mit  $\delta_3((z_1, z_2), x) = (\delta_1(z_1, x), \delta_2(z_2, x))$  und  $F_3 = (F_1 \times Z_2) \cup (Z_1 \times F_2)$ .

$u \in L_1 \cup L_2 \Rightarrow u \in L_1 \vee u \in L_2$ , also  $\overline{\delta_1}(z_0^1, u) \in F_1 \vee \overline{\delta_2}(z_0^2, u) \in F_2$ , also ist  $(\overline{\delta_1}(z_0^1, u), \overline{\delta_2}(z_0^2, u)) \in F_3$ , also  $\overline{\delta_3}((z_0^1, z_0^2), u) \in F_3$ , also  $u \in L(A_3)$ .

Aus  $u \in L(A_3)$  folgt  $u \in L_1 \cup L_2$  (entsprechend)

**Lemma 3.8**

Können  $L_1 \subseteq X^*$  und  $L_2 \subseteq X^*$  von DFA's erkannt werden, dann auch  $L_1 \cdot L_2$ .

Bew.:  $L_1 = L(A_1), L_2 = L(A_2)$ , analog wie oben.

O.B.d.A.  $Z_1 \cap Z_2 = \emptyset$

Setze  $A = (Z, X, \delta, z_0, F)$  mit  $F = \mathbb{P}(Z_1 \cup Z_2)$  und  $z_0 = \{z_0^1\}$ .

$\delta$  werde folgendermaßen festgelegt:

$$\forall z \in Z_1, \quad \forall x \in X : \quad \delta(\{z\}, x) = \begin{cases} \delta_1(z, x) & \text{falls } z \in Z_1 \setminus F_1 \\ \{\delta_1(z, x), \delta_2(z_0^2, x)\} & \text{falls } z \in F_1 \end{cases}$$

$$\forall z \in Z_2, \quad \forall x \in X : \quad \delta(\{z\}, x) = \{\delta_2(z, x)\}$$

$$\forall Y \in Z, \quad \forall x \in X : \quad \delta(Y, x) = \bigcup_{z \in Y} \delta(z, x)$$

$F$  werde folgendermaßen festgelegt:

a)  $e \notin L_2$ : Dann ist  $F = \{Y \mid Y \in Z \text{ und } Y \cap F_2 \neq \emptyset\}$

b)  $e \in L_2$ : Dann ist  $F = \{Y \mid Y \in Z \text{ und } Y \cap (F_1 \cup F_2) \neq \emptyset\}$

Es ist  $L(A) = L_1 \cdot L_2$  zu beweisen:

Es liege a) vor. Sei  $w \in L_1 \cdot L_2$ , also  $w = u \cdot v$  mit  $u \in L_1, v \in L_2, v \neq e$ .

Es ist dann also  $\overline{\delta_1}(\{z_0^1\}, u) \in F_1$  und  $\overline{\delta_2}(\{z_0^2\}, v) \in F_2$ . Damit gilt:

$\overline{\delta}(\{z_0^1\}, u)$  enthält ein Element aus  $F_1$  und dann (wegen  $v \neq e$ )

$\overline{\delta}(\overline{\delta}(\{z_0^1\}, u), v) = \overline{\delta}(\{z_0\}, uv)$  enthält ein Element aus  $F_2$ .

b) mit a) trivial

Sei umgekehrt  $w \in L(A)$ .  $u_1 u_2 \dots u_k$  seien genau die Präfixe von  $w$  mit

$\overline{\delta_1}(\{z_0^1\}, u_i) \in F_1, i \in [1 : k]$ .

Da  $\overline{\delta}(z_0, w)$  ein Element aus  $F_2$  enthält,

gilt  $k \geq 1$  und  $\exists i \in [1 : k]$  mit  $w = u_i \cdot v_i$  und  $\overline{\delta_2}(z_0^2, v_i) \in F_2$ .

Daher ist  $L(A) \subseteq L_1 \cdot L_2$ .

$$L^* = \bigcup_{k=0}^{\infty} L^k = \{e\} \cup L \cup L^2 \cup L^3 \cup \dots$$

**Lemma 3.9**

Kann  $L \subseteq X^*$  von einem DFA erkannt werden, dann auch  $L^*$ .

Bew.:  $L$  werde von  $A^* = (Z^*, X, \delta^*, z_0, F^*)$  erkannt.

Setze  $A' = (Z', X, \delta', c_0, F')$  mit  $Z' = \{c_0\} \cup \mathbb{P}(Z)$  und  $c_0 \notin Z^*$ .

$c_0$  ist ein neuer (Start-) Zustand.

$\delta'$  werde folgendermaßen definiert:

$$\begin{aligned} \forall z \in Z, \forall x \in X : \quad \delta'(\{z\}, x) &= \begin{cases} \{\delta(z, x)\} & \text{falls } z \in Z \setminus F \\ \{\delta(z, x), \delta(z_0, x)\} & \text{falls } z \in Z \end{cases} \\ \forall x \in X : \quad \delta'(\{c_0\}, x) &= \delta'(\{z_0\}, x) \\ \forall Y \in \mathbb{P}(Z), \forall x \in X : \quad \delta'(Y, x) &= \bigcup_{z \in Y} \delta(\{z\}, x) \\ F' &= \{c_0\} \cup \{Y \mid Y \in \mathbb{P}(Z) \text{ mit } Y \cap F \neq \emptyset\} \end{aligned}$$

□

### Satz 3.10

Für eine Sprache  $L \subseteq X^*$  sind folgende Aussagen gleichwertig:

- (a)  $L$  ist eine rechtslineare Sprache.
- (b)  $L$  ist eine reguläre Menge.
- (c)  $L$  wird von einem DFA erkannt.
- (d)  $L$  wird von einem NFA erkannt.

# Kapitel 4

## Das Iterationslemma für reguläre Sprachen

### Satz 4.1 (Iterationslemma)

$L$  sei eine reguläre Sprache über dem Alphabet  $X$ . Es gibt ein  $k > 0$ , so daß gilt:

Jedes  $w \in L$  mit  $|w| \geq k$  läßt sich so als  $w = x \cdot y \cdot z$  mit  $y \neq e, |x \cdot y| \leq k$  schreiben, daß für alle  $i \geq 0$  gilt:

$x \cdot y^i \cdot z \in L$ .

Bew.:  $A = (Z, X, \delta, z_0, F)$  sei ein DFA mit  $L = L(A)$  und  $|Z| = k$ .

Sei  $w = a_1 \dots a_m \in L$  mit  $|w| \geq k$ .

Betrachte die Zustände  $\bar{\delta}(\{z_0\}, a_1 \dots a_i), i \in [0 : k]$ . (Das sind  $k + 1$  Zustände!)

Es existiert ein  $0 \leq p < r \leq k$  mit  $\bar{\delta}(\{z_0\}, a_1 \dots a_p) = \bar{\delta}(\{z_0\}, a_1 \dots a_r)$ .

Setze  $x = a_1 \dots a_p; y = a_{p+1} \dots a_r (\neq e); z = a_{r+1} \dots a_m$ . □

### Beispiel 4.2

$L = \{a^n \cdot b^n | n \geq 0\}$  ist keine reguläre Sprache.

Bew.: Ann.:  $L$  ist regulär.

Wähle  $k$  wie im Iterationslemma und betrachte das Wort  $a^k \cdot b^k \in L$ .

Schreibe  $a^k \cdot b^k = x \cdot y \cdot z$  mit  $|x \cdot y| \leq k, y \neq e$  und  $x \cdot y^2 \cdot z \in L$  ergibt sich ein Widerspruch, da  $x \cdot y^2 \cdot z$  mehr  $a$ 's als  $b$ 's enthält. □

### Beispiel 4.3 $X = \{0\}$ . $L = \{0^{n^2} | n \geq 1\}$ ist keine reguläre Sprache.

Bew.: Ann.:  $L$  ist regulär.

Wähle  $k$  wie im Iterationslemma. Setze  $L \ni 0^{k^2} = x \cdot y \cdot z$  mit  $1 \leq |y| \leq k$ .

Es gilt  $x \cdot y^i \cdot z \in L \quad \forall i \geq 0$ . Wähle  $i = 2$ . Dann ist  $x \cdot y^2 \cdot z \in L$ .

Es ist  $k^2 = |x \cdot y \cdot z| < |x \cdot y^2 \cdot z| \leq k^2 + k < (k + 1)^2$ . □

### Beispiel 4.4 $X = \{0, 1\}$ . $L = \{w | w \in 1 \cdot X^* \text{ und } w \text{ ist Dualdarstellung einer Primzahl}\}$ ist keine reguläre Sprache.

Bew.: Um dies zu zeigen, benötigt man folgendes:

1) Es gibt unendlich viele Primzahlen

2) Für jede Primzahl  $p > 2$  gilt:  $2^{p-1} \equiv 1 \pmod{p}$ . (Spezialfall des kleinen Fermatschen Satzes)

Ann.:  $L$  ist regulär. Wähle  $k$  wie im Iterationslemma.

Sei  $w$  die Dualdarstellung einer Primzahl  $p$  mit  $p > 2^k$ .

Es ist  $|w| > k$ . Schreibe  $w = x \cdot y \cdot z$  wie im Iterationslemma.

Dann müssen auch  $x \cdot y^i \cdot z, i \geq 0$  Dualdarstellungen einer Primzahl sein.

Seien  $x, y, z$  die Dualdarstellungen von  $n_x, n_y, n_z$ .

Falls  $x = e$  bzw.  $z = e$ , dann setze  $n_x = 0$  bzw.  $n_z = 0$ .

Wähle  $i = p$ :  $x \cdot y^p \cdot z$  muß Dualdarstellung einer Primzahl  $q$  sein.

Es ist  $p < q$ . Es gilt:  $q = n_x 2^{|z|+p|y|} + n_y 2^{|z|} \underbrace{(1 + 2^{|y|} + \dots + 2^{(p-1)|y|})}_S + n_z$

Es gilt  $2^{p-1} \equiv 1 \pmod{p}$ , also auch  $2^{(p-1)|y|} \equiv 1 \pmod{p}$ , also  $2^{p|y|} \equiv 2^{|y|} \pmod{p}$

Dann gilt:  $(2^{|y|} - 1)S = (2^{|y|} - 1)(1 + 2^{|y|} + 2^{2|y|} + \dots + 2^{(p-1)|y|}) =$   
 $2^{|y|} + 2^{2|y|} + \dots + 2^{(p-1)|y|} + 2^{p|y|}$

$-1 - 2^{|y|} - 2^{2|y|} - \dots - 2^{(p-1)|y|} = 2^{p|y|} - 1$

Wegen  $2^{p|y|} \equiv 2^{|y|} \pmod{p}$  gilt also:  $(2^{|y|} - 1)S = 2^{p|y|} - 1 \equiv 2^{|y|} - 1 \pmod{p}$ , also  $p \mid (2^{|y|} - 1)S - (2^{|y|} - 1) = \underbrace{(2^{|y|} - 1)(S - 1)}_{< 2^k}$

Wegen  $1 \leq |y| \leq k$  gilt:  $2 \leq 2^{|y|} \leq 2^k \leq p$  und damit gilt:  $p \nmid 2^{|y|} - 1$ .

Also muß  $p \mid (S - 1)$  gelten, also  $S \equiv 1 \pmod{p}$ .

Es ist  $q = n_x \cdot 2^{|z|+p|y|} + n_y \cdot 2^{|z|} \cdot S + n_z = n_x \cdot 2^{|z|} \cdot \underbrace{2^{p-1}}_{\equiv 1} \cdot 2^{p|y|} + n_y \cdot 2^{|z|} \cdot \underbrace{S}_{\equiv 1} + n_z \equiv \underbrace{n_x \cdot 2^{|z|+|y|} + n_y \cdot 2^{|z|} + n_z}_p \pmod{p}$

Also  $q \equiv p \pmod{p}$ , also  $p \mid q, p < q$ . Widerspruch! □

# Kapitel 5

## Abschlußeigenschaften regulärer Sprachen

### Satz 5.1

Seien  $L, L_1, L_2 \subseteq X^*$  reguläre Sprachen. Dann sind auch  $L_1 \cup L_2, L_1 \cdot L_2, L^*$  reguläre Sprachen.

### Satz 5.2 ( $\bar{L}$ ist regulär)

Ist  $L \subseteq X^*$  eine reguläre Sprache, dann ist auch  $X^* \setminus L$  eine reguläre Sprache.

Bew.: Sei  $L = L(A)$  für einen DFA  $A = (Z, X, \delta, z_0, F)$ .

$\bar{L} = X^* \setminus L$  wird dann von  $\bar{A} = (Z, X, \delta, z_0, Z \setminus F)$  erkannt. □

### Satz 5.3

Sind  $L_1, L_2$  reguläre Sprachen, dann ist auch  $L_1 \cap L_2$  eine reguläre Sprache.

Bew.: Es ist  $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ . □

### Definition 5.4 (Substitution)

Seien  $X, Y$  Alphabete. Eine **Substitution** ist eine Abbildung  $f : X \rightarrow \mathbb{P}(Y^*)$ .

$f$  heißt **reguläre Substitution**, wenn  $\forall x \in X$  gilt:  $f(x)$  ist regulär.

Erweiterung von  $f$  auf  $X^*$ :

$$(1) f(e) = \{e\}$$

$$(2) f(w \cdot x) = f(w) \cdot f(x) \quad w \in X^*, x \in X$$

Erweiterung von  $f$  auf Sprachen  $L \subseteq X^*$ :

$$f(L) = \bigcup_{w \in L} f(w)$$

z.B.:  $X = \{0, 1\}, Y = \{a, b\}, f(0) = \{a\}, f(1) = \{b\}^*$  ( $f$  ist reguläre Substitution)

Es ist  $f(010) = a \cdot b^* \cdot a$ .

Ist  $L = 0^* \cdot (0 \cup 1) \cdot 1^*$ , dann ist  $f(L) = a^* \cdot (a \cup b^*) \cdot (b^*)^* = (a^* \cdot a \cup a^* \cdot b^*) \cdot b^* = a^* \cdot a \cdot b^* \cup a^* \cdot b^* \cdot b^* = a^* \cdot b^*$  regulär.

### Satz 5.5

Seien  $X, Y$  Alphabete und  $f : X \rightarrow \mathbb{P}(Y^*)$  eine reguläre Substitution. Ist  $L \subseteq X^*$  regulär, dann ist auch  $f(L) \subseteq Y^*$  regulär.

Bew.: (Induktion über den Aufbau regulärer Mengen aus  $X^*$ ).

- $f(\emptyset) = \emptyset$  und  $f(x), x \in X$  sind regulär.
- $L_1, L_2 \subseteq X^*$ , so daß  $f(L_1), f(L_2)$  regulär sind. Dann sind auch:
  - $f(L_1 \cup L_2) = f(L_1) \cup f(L_2)$  und
  - $f(L_1 \cdot L_2) = f(L_1) \cdot f(L_2)$  regulär.
- Ist  $L \subseteq X^*$  so, daß  $f(L)$  regulär ist, dann ist auch

$$f(L^*) = f\left(\bigcup_{i=0}^{\infty} L^i\right) = \bigcup_{i=0}^{\infty} f(L^i) = \bigcup_{i=0}^{\infty} (f(L))^i = (f(L))^* \text{ regulär.}$$

Sei  $h : X^* \rightarrow Y^*$  Homomorphismus (festgelegt durch die Werte auf den Buchstaben von  $X$ ). Dieser legt eine reguläre Substitution  $h' : X \rightarrow \mathbb{P}(Y^*)$  mit  $h'(x) = \{h(x)\} \quad \forall x \in X$  fest. □

### Korollar 5.6

Ist  $h : X^* \rightarrow Y^*$  ein Homomorphismus<sup>1</sup> und  $L \subseteq X^*$  eine reguläre Sprache, dann ist auch  $h(L)$  regulär.

### Satz 5.7

Sei  $h : X^* \rightarrow Y^*$  ein Homomorphismus und sein  $L \subseteq Y^*$  regulär. Dann ist

$h^{-1}(L) := \{w \mid w \in X^* \text{ und } h(w) \in L\}$  regulär.

Bew.: Es wird ein DFA  $A' = (Z, X, \delta', z_0, F)$  konstruiert, der  $h^{-1}(L)$  akzeptiert.

Wenn  $A'$  ein  $x \in X$  einliest, wird die Arbeitsweise von  $A$  bei Eingabe von  $h(x)$  simuliert, d.h.:

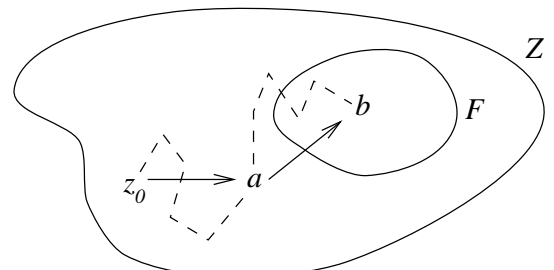
$$\forall z \in Z, \forall x \in X : \delta'(z, x) = \bar{\delta}(\{z\}, h(x))$$

Es gilt (Induktion über  $|w|$ ):

$$\forall z \in Z, \forall w \in X^* : \delta'(z, w) = \bar{\delta}(\{z\}, h(w)), \text{ also:}$$

$$w \in h^{-1}(L) \Leftrightarrow h(w) \in L \Leftrightarrow \bar{\delta}(\{z_0\}, h(w)) \in F$$

$$\Leftrightarrow \delta'(\{z_0\}, w) \in F \Leftrightarrow w \in L(A')$$



□

<sup>1</sup>m.E. eine reguläre Substitution

# Kapitel 6

## Kontextfreie Grammatiken und Sprachen

### Definition 6.1 (kontextfrei, linear kontextfrei)

Eine Grammatik  $G = (V, X, S, P)$  heißt **kontextfrei**, wenn alle Produktionen von der Form  $A \rightarrow \alpha$ ,  $A \in V, \alpha \in (V \cup X)^*$  sind. Kommt in  $\alpha$  jeweils höchstens eine Variable vor, dann heißt  $G$  **linear kontextfrei**.

Eine Sprache  $L \subseteq X^*$  heißt **(linear-) kontextfrei**, wenn es eine (linear-) kontextfreie Grammatik  $G = (V, X, S, P)$  mit  $L = L(G)$  gibt.

Die Sprache  $L = \{a^n \cdot b^n \mid n \geq 0\}$  ist linear kontextfrei, aber nicht regulär.

### Satz 6.2

Jede reguläre Sprache ist linear kontextfrei. Es gibt linear-kontextfreie Sprachen, die nicht regulär sind.

### Beispiel 6.3

$G = (V, X, S, P)$  mit  $V = \{S, A, B\}$ ,  $X = \{a, b\}$  und  
 $P = \{S \rightarrow bA, S \rightarrow aB, A \rightarrow a, A \rightarrow aS, A \rightarrow bAA, B \rightarrow b, B \rightarrow bS, B \rightarrow aBB\}$

Beh.:  $\forall w \in X^+$ :

- (a)  $S \xrightarrow{*} w$  g.d.w.  $w$  besitzt die gleiche Anzahl von  $a$ 's und  $b$ 's.
- (b)  $A \xrightarrow{*} w$  g.d.w.  $w$  besitzt ein  $a$  mehr als  $b$ 's.
- (c)  $B \xrightarrow{*} w$  g.d.w.  $w$  besitzt ein  $b$  mehr als  $a$ 's.

Bew.: (durch Induktion über  $|w|$ ):

- $|w| = 1$  : — aus  $S$  kann kein Wort der Länge 1 abgeleitet werden.  
— aus  $A$  kann nur  $a$  als Wort der Länge 1 abgeleitet werden.  
— aus  $B$  kann nur  $b$  als Wort der Länge 1 abgeleitet werden.

Die Behauptung gelte für alle  $w \in \{a, b\}^+$  mit  $|w| \leq k - 1$

Sei  $w \in \{a, b\}^+$  mit  $|w| = k$ .

a) – Es gelte  $S \xrightarrow{*} w$ . Die Ableitung beginne mit  $S \Rightarrow aB$ . Dann gilt:

$$B \xrightarrow{*} w_1 \text{ mit } w = a \cdot w_1 \text{ und } |w_1| = k - 1.$$

Die Ableitung beginne mit  $S \Rightarrow b \cdot A \dots$  (entsprechend)

– Es gelte umgekehrt:  $w$  besitze die gleiche Anzahl von  $a$ 's und  $b$ 's.

Dann ist  $w = a \cdot w_1$  und  $w_1$  besitzt ein  $b$  mehr als  $a$ 's. Also können wir:

$$S \Rightarrow aB \Rightarrow a \cdot w_1 = w$$

$$\text{oder } w = b \cdot w_1 \dots \text{ (entsprechend)}$$

b) – Es gelte  $A \xrightarrow{*} w$ .

Die Ableitung beginne mit  $A \Rightarrow a \cdot S$  (klar)

Die Ableitung beginne mit  $A \Rightarrow b \cdot A \cdot A$  (auch klar)

– Es gelte umgekehrt:  $w$  besitze ein  $a$  mehr als  $b$ 's. Dann:

$$w = a \cdot w_1 \text{ und } S \xrightarrow{*} w_1, \text{ also } A \Rightarrow aS \xrightarrow{*} a \cdot w_1 = w.$$

$$w = b \cdot w_1. w_1 \text{ hat dann zwei } a\text{'s mehr als } b\text{'s. Also } w_1 = w'_1 \cdot w''_1 \text{ und } w'_1, w''_1$$

besitzen jeweils ein  $a$  mehr als  $b$ 's. Dann sind  $w'_1, w''_1$  aus  $A$  herleitbar.

$$A \Rightarrow b \cdot A \cdot A \Rightarrow b \cdot w'_1 \cdot A \Rightarrow b \cdot w'_1 \cdot w''_1 = w$$

c) analog

### Beispiel 6.4

Betrachte die Sprache  $\tilde{L} = \{w \cdot w \mid w \in \{a, b\}^*\}$ .  $L$  sei das Komplement von  $\tilde{L}$ .<sup>1</sup>

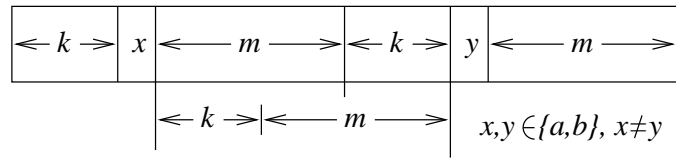
$L$  wird von der kontextfreien Grammatik  $G = (V, X, S, P)$  mit  $V = \{S, A, S_a, S_b\}$ ,  $X = \{a, b\}$  und den Produktionen  $P = P_1 \cup P_2$  erzeugt mit:

$$P_1 = \{S \rightarrow a|b, S \rightarrow A \cdot A \cdot a|A \cdot A \cdot b, A \rightarrow A \cdot A \cdot A, A \rightarrow a|b\}^2$$

Ein Wort gerader Länge ( $\neq \epsilon$ ), das sich nicht in der Form  $w \cdot w$ ,  $w \in \{a, b\}^*$  schreiben läßt, läßt sich folgendermaßen zerlegen:

<sup>1</sup>Alle Wörter in  $\tilde{L}$  haben eine gerade Länge

<sup>2</sup>Mit  $P_1$  kann man genau die Wörter ungerader Länge über  $\{a, b\}$  ableiten.



Setze  $P_2 = \{S \rightarrow S_x \cdot S_y$  mit  $x, y \in \{a, b\}$  und  $x \neq y$ ,  
 $S_z \rightarrow x \cdot S_z \cdot y$  mit  $x, y, z \in \{a, b\}$ ,  
 $S_z \rightarrow z$  mit  $z \in \{a, b\}\}$

# Kapitel 7

## Transformation kontextfreier Grammatiken

Zwei Grammatiken  $G$  und  $G'$  heißen **äquivalent**, wenn  $L(G) = L(G')$  gilt.

### Definition 7.1 ( $e$ -Bedingung)

Eine kontextfreie Grammatik  $G = (V, X, S, P)$  erfüllt die  $e$ -**Bedingung**, wenn für alle Produktionen  $A \rightarrow \alpha \in P$  gilt:

- (1) Ist  $\alpha = e$ , dann ist  $A = S$ .
- (2) Ist  $S \rightarrow e \in P$ , dann kommt  $S$  in keiner rechten Seite einer Produktion vor.

### Satz 7.2 (Entfernen von $e$ -Produktionen)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik. Es gibt eine zu  $G$  äquivalente kontextfreie Grammatik  $G' = (V', X, S', P')$ , die die  $e$ -Bedingung erfüllt.

Bew.: Setze  $\tilde{V} = \{A \mid A \in V \text{ und } A \xrightarrow{+}_G e\}$ .

$P'$  werde folgendermaßen festgelegt:

- (a) Sei  $A \rightarrow \alpha_0 B_1 \alpha_1 B_2 \alpha_2 \dots B_k \alpha_k \in P$  mit  $B_i \in \tilde{V}$  für  $i \in [1 : k]$ ,  $k \geq 0$  und in keinem  $\alpha_j$ ,  $j \in [0 : k]$  trete ein Buchstabe aus  $\tilde{V}$  auf. Dann sind in  $P'$  alle Produktionen  $A \rightarrow \alpha_0 X_1 \alpha_1 X_2 \alpha_2 \dots X_k \alpha_k$  mit  $X_i \in \{B_i, e\}$ ,  $i \in [1 : k]$ , für die  $\alpha_0 X_1 \alpha_1 X_2 \alpha_2 \dots X_k \alpha_k \neq e$  ist.
- (b) Falls  $S \in \tilde{V}$ . Dann enthält  $P'$  die Produktionen  $S' \rightarrow e$  und  $S' \rightarrow S$ , wobei  $S'$  eine neue Variable ist. Im Fall (b) setze  $V' = V \cup \{S'\}$ , sonst  $V' = V$ ,  $S' = S$ .

$G' = (V', X, S', P')$  erfüllt die  $e$ -Bedingung. Es bleibt noch  $L(G) = L(G')$  zu zeigen:

Dazu wird zunächst bewiesen:

$$\forall A \in V \quad \forall \alpha \in (V \cup X)^* : A \xrightarrow{*}_{G'} \alpha \text{ g.d.w. } A \xrightarrow{*}_G \alpha.$$

Bew. (Induktion über die Anzahl  $n$  der Ableitungsschritte):

- (1) Es gelte  $A \xrightarrow{n}_{G'} \alpha$ :

$$\underline{n = 0} : A = \alpha \text{ und es gilt } A \xrightarrow{0}_G \alpha.$$

$$\underline{n = 1} : \text{Es ist } A \rightarrow \alpha \in P', \text{ also } A \rightarrow \alpha_0 X_1 \alpha_1 X_2 \alpha_2 \dots X_k \alpha_k \text{ gemäß (a) und}$$

$$A \rightarrow \alpha_0 B_1 \alpha_1 B_2 \alpha_2 \dots B_k \alpha_k \in P \text{ mit } B_i \xrightarrow{+}_G e \text{ für } i \in [1 : k], \text{ also gilt } A \xrightarrow{+}_G \alpha.$$

$$\underline{n \rightarrow n + 1} : \text{Gelte } A \xrightarrow{n}_{G'} \alpha_1 \xrightarrow{*}_G \alpha. \text{ Es gilt dann } A \xrightarrow{*}_G \alpha_1 \text{ (nach IV).}$$

$$\text{Sei } \alpha_1 = \beta_1 B \beta_2, \alpha = \beta_1 \beta \beta_2 \text{ und } B \rightarrow \beta \in P'.$$

$$\text{Es ist } \beta \neq e \text{ und aufgrund der IV gilt: } B \xrightarrow{*}_G \beta, \text{ zusammen: } A \xrightarrow{*}_G \alpha.$$

- (2) Es gelte  $A \xrightarrow{n}_G \alpha$ .

$$\underline{n = 0} : \text{(klar)}$$

$$\underline{n = 1} : \text{Es ist } A \rightarrow \alpha \in P, \alpha \neq e, \text{ und damit } A \rightarrow \alpha \in P'.$$

$$\underline{n \rightarrow n + 1} : \text{Gelte } A \xrightarrow{1}_G \alpha_1 \xrightarrow{n}_G \alpha. \text{ Schreibe } \alpha_1 = \beta_1 A_1 \beta_2 A_2 \dots \beta_m A_m \beta_{m+1} \text{ so, daß gilt:}$$

$$A_i \xrightarrow{n_i}_G \gamma_i, n_i < n, i \in [1 : m] \text{ und } \alpha = \beta_1 \gamma_1 \beta_2 \gamma_2 \dots \beta_m \gamma_m \beta_{m+1}.$$

$$\text{Ist } \gamma_i \neq e, \text{ dann gilt } A_i \xrightarrow{*}_{G'} \gamma_i \text{ (nach IV)}$$

Ist  $\gamma_i = e$ , dann streiche  $A_i$  aus  $\alpha_1$ . Man erhält so eine Produktion

$$A \rightarrow \alpha'_1 \in P' \text{ und es gilt: } A \xrightarrow{1}_{G'} \alpha'_1 \xrightarrow{*}_{G'} \alpha.$$

Ist  $e \notin L(G)$ , dann gilt:

$$L(G') = \{w \mid w \in X^+ \text{ und } S \xrightarrow{*}_{G'} w\} = \{w \mid w \in X^+ \text{ und } S \xrightarrow{*}_G w\} = L(G).$$

Ist  $e \in L(G)$ , dann gilt:

$$L(G') = \{e\} \cup \{w \mid w \in X^+ \text{ und } S \xrightarrow{*}_{G'} w\} = \{e\} \cup \{w \mid w \in X^+ \text{ und } S \xrightarrow{*}_G w\} = L(G). \quad \square$$

### Beispiel 7.3

$G$  enthalte folgende Produktionen:  $S \rightarrow aSbS|bSaS|e$ .

$P'$  enthält die Produktionen:

$$S' \rightarrow S|e$$

$$S \rightarrow aSbS|aSb|abS|ab$$

$$S \rightarrow bSaS|baS|bSa|ba$$

## Algorithmus 7.4 ( $L(G)=\emptyset?$ )

Eingabe :  $G = (V, X, S, P)$  sei kontextfrei.

Ausgabe : „JA“, falls  $L(G) \neq \emptyset$ , „NEIN“ sonst.

Methode : (1)  $V_0 := \emptyset$  und  $i := 1$ .

(2)  $V_i := \{A \mid A \rightarrow \alpha \in P, \alpha \in (V_{i-1} \cup X)^*\} \cup V_{i-1}$ .

(3) Falls  $V_i \neq V_{i-1}$ , dann  $i := i + 1$  und weiter bei (2), sonst  $V_e := V_i$

(4) Falls  $S \in V_e$ : „JA“, sonst: „NEIN“

## Satz 7.5

Algorithmus 7.4 ergibt „JA“ genau dann, wenn  $S \xRightarrow{*} w$  für ein  $w \in X^*$  gilt.

Bew.: Verabredung: Sei  $i$  so, daß sich im Algorithmus  $V_e = V_i$  ergibt ( $V_i = V_{i-1}$ ).

Setze  $\forall j > i : V_j := V_i$ .

Es wird folgendes durch Induktion über  $l$  bewiesen:  $\forall l : A \in V_l$ , dann  $A \xRightarrow{*} w$  für ein  $w \in X^*$ .

$l=1$  : (klar)

$l \rightarrow l+1$  :  $A \in V_{l+1}$ .

Falls  $A \in V_l$ . (klar)

Falls  $A \in V_{l+1} \setminus V_l$ , dann gibt es  $A \rightarrow X_1 \dots X_k$  mit  $X_j \in V_l$  oder  $X_j \in X$ ,  $j \in [1 : k]$ , also  $\forall j \in [1 : k] \exists w_j \in X^*$  mit

$X_j \xRightarrow{*} w_j$ .

Dann:  $A \Rightarrow X_1 \dots X_k \xRightarrow{*} w_1 X_2 \dots X_k \xRightarrow{*} w_1 \dots w_k = w \in X^*$

Also: Wenn Algorithmus 7.4 „JA“ ergibt, dann ist  $S \in V_e = V_i$ , also  $\exists w \in X^* : S \xRightarrow{*} w, w \in X^*$ .

Umkehrung: Durch Induktion über  $n$  wird gezeigt: Falls  $A \xRightarrow{n} w, w \in X^*$  gilt, dann ist  $A \in V_e$ .

$n=1$  :  $A \in V_1 \subseteq V_e$ .

$n \rightarrow n+1$  :  $A \xRightarrow{1} X_1 \dots X_k \xRightarrow{n} w$ .

Dann gibt es  $X_j \xRightarrow{n_j} w_j, n_j \leq n, j \in [1 : k]$  mit  $w = w_1 \dots w_k$ .

Falls  $X_j \in V$ , dann ist laut IV  $X_j \in V_e = V_{i-1}$ , also  $A \in V_i = V_e$ .

Gilt also  $S \xRightarrow{*} w, w \in X^*$ , dann ist  $S \in V_e$ , also gibt der Algorithmus „JA“ aus. □

## Korollar 7.6

Gegeben sei eine kontextfreie Grammatik  $G = (V, X, S, P)$ . Es ist entscheidbar, ob  $L(G) = \emptyset$  ist.

## Definition 7.7 (Unerreichbare Zeichen)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik. Ein Zeichen  $a \in V \cup X$  heißt **unerreichbar**, falls  $a$  in keiner Satzform bzgl.  $G$  vorkommt, sonst **erreichbar**.

## Algorithmus 7.8 (Entfernen von unerreichbaren Buchstaben)

Eingabe :  $G = (V, X, S, P)$  sei kontextfrei.

Ausgabe :  $G' = (V', X', S, P')$  mit (1)  $L(G) = L(G')$

(2)  $\forall a \in (V' \cup X') \exists \alpha, \beta \in (V' \cup X')^*$  mit  $S \xRightarrow{*}_{G'} \alpha a \beta$ .

Methode : (1) Setze  $V_0 := \{S\}$  und  $i := 1$ .

(2) Setze  $V_i := \{a \mid a \in V \cup X \text{ und } A \rightarrow \alpha a \beta \text{ und } A \in V_{i-1}\} \cup V_{i-1}$

(3) Falls  $V_i \neq V_{i-1}$ , dann  $i := i + 1$ , weiter bei (2)

Sonst  $V' = V_i \cap V$

$X' = V_i \cap X$

$P' =$  alle Produktionen aus  $P$ , die nur Buchstaben aus  $V_i$  enthalten.

$G = (V', X', S, P')$

Es gilt:  $S \xRightarrow{*}_G \alpha a \beta$  g.d.w.  $a \in V' \cup X'$ . □

## Definition 7.9 (Nutzlose Zeichen)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik. Ein  $a \in V \cup X$  heißt **nutzlos**, wenn es keine Ableitung der Form  $S \xRightarrow{*} w_1 a w_2 \xRightarrow{*} w_1 v w_2 \in X^*$  gibt.

## Algorithmus 7.10 (Entfernen nutzloser Buchstaben)

Eingabe :  $G = (V, X, S, P)$  kontextfrei mit  $L(G) \neq \emptyset$ .

Ausgabe :  $G' = (V', X', S, P')$  kontextfrei mit  $L(G) = L(G')$  und kein  $a \in V' \cup X'$  ist nutzlos.

Methode : (1) Anwendung von 7.4 zur Bestimmung von  $V_e$ .

Setze  $G_1 = (V_e, X, S, P_1)$  mit  $P_1 =$  alle Produktionen aus  $P$ , die nur Buchstaben aus  $V_e \cup X$  enthalten.

(Wegen  $L(G) \neq \emptyset$  ist  $S \in V_e$ .)

(2) Anwendung von 7.8 auf  $G_1$  ergibt  $G' = (V', X', S, P')$ .

## Satz 7.11

$G'$  enthält keine nutzlosen Buchstaben und es ist  $L(G) = L(G')$ .

Bew.: Ann.:  $A \in V'$  ist nutzlos, d.h. es gibt keine Wörter  $\alpha, w, \beta \in X'^*$ , so daß  $S \xRightarrow{*}_G \alpha A \beta \xRightarrow{*}_{G'} \alpha w \beta$  gilt.

1.Fall:  $S \xRightarrow{*}_G \alpha A \beta$  ist für keine Wörter  $\alpha, \beta \in X'^*$  möglich.

$A$  ist bzgl.  $G'$  erreichbar, also:  $\exists w_1, w_2 \in (X' \cup V')^*$  mit  $S \xrightarrow{G'}^* w_1 A w_2$ .

Jede Produktion aus  $G'$  kommt auch in  $G_1$  vor, also gilt auch:  $S \xrightarrow{G_1}^* w_1 A w_2$ .

Jede in  $w_1$  oder  $w_2$  vorkommende Variable kann bzgl.  $G_1$  in ein Wort über  $X$  abgeleitet werden, also:

$$(*) S \xrightarrow{G_1}^* \alpha'_1 A \alpha'_2 \text{ mit } \alpha'_1, \alpha'_2 \in X^*.$$

Alle Buchstaben aus  $\alpha'_1, \alpha'_2$  sind bzgl.  $G_1$  erreichbar, also  $\alpha'_1, \alpha'_2 \in X'^*$ .

Alle in (\*) angewendeten Produktionen sind auch Produktionen in  $P'$ , also  $S \xrightarrow{G'}^* \alpha'_1 A \alpha'_2$ .

(Widerspruch, da es  $\alpha, \beta$  doch gibt!)

**2.Fall:**  $S \xrightarrow{G'}^* \alpha A \beta$  ist für geeignete  $\alpha, \beta \in X'^*$  möglich, aber  $A \xrightarrow{G'}^* w$  gilt für kein  $w \in X'^*$ .

Also gilt auch  $S \xrightarrow{G_1}^* \alpha A \beta$  und aus  $A \in V_e$  folgt  $A \xrightarrow{G_1}^* w$  für ein geeignetes  $w \in X^*$ .

Die Ableitung bzgl.  $G_1$  ist auch Ableitung bzgl.  $G'$ , da alle Buchstaben erreichbar sind.

Somit verbleibt noch zu zeigen:

Ann.:  $x \in X'$  nutzlos (entsprechend) □

### Beispiel 7.12

$G = (\{S, A, B\}, \{a, b\}, S, P)$  mit  $P = \{S \rightarrow a|A, A \rightarrow AB, B \rightarrow b\}$ .

**Bestimmung von  $G_1$ :**

$$V_1 = \{S, B\}$$

$$V_2 = \{S, B\} = V_e \quad G_1 = (\{S, B\}, \{a, b\}, S, \{S \rightarrow a, B \rightarrow b\})$$

**Bestimmung von  $G'$ :**

$$V_0 = \{S\}$$

$$V_1 = \{S, a\}$$

$$V_2 = \{S, a\} \quad G' = (\{S\}, \{a\}, S, \{S \rightarrow a\})$$

**Definition 7.13** (Singular)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik. Eine Produktion  $A \rightarrow B \in P$  mit  $A, B \in V$  heißt **singuläre Produktion von  $G$** .

**Algorithmus 7.14** (Entfernen singulärer Produktionen)

Eingabe :  $G = (V, X, S, P)$  kontextfrei mit  $e$ -Bedingung.

Ausgabe :  $G' = (V, X, S, P')$  kontextfrei mit:

(a)  $G'$  enthält keine singulären Produktionen

(b)  $L(G) = L(G')$

Methode : (1) Für jedes  $A \in V$  konstruiere  $V_A$  folgendermaßen:

(a)  $V_0 = \{A\}$  und  $i = 1$ .

(b)  $V_i = \{C \mid B \rightarrow C \in P \text{ und } B \in V_{i-1}\} \cup V_{i-1}$

(c) Falls  $V_i \neq V_{i-1}$ , dann setze  $i := i + 1$ , weiter bei (b), sonst:  $V_A := V_i$ .

(2) Konstruktion von  $P'$ : Falls  $B \rightarrow \alpha \in P$  nicht singular, dann sei  $A \rightarrow \alpha \in P'$  für alle  $A$  mit  $B \in V_A$ .

### Bemerkung 7.15

Gilt  $A \xrightarrow{G}^* B$ , dann können bei dieser Ableitung wegen der  $e$ -Bedingung für  $G$  nur singuläre Produktionen angewendet werden. Also

$$V_A = \{B \mid A \xrightarrow{G}^* B\}.$$

$G'$  besitzt keine singulären Produktionen.

Es ist  $L(G) = L(G')$

$G'$  erfüllt die  $e$ -Bedingung.

### Beispiel 7.16

Anwendung des vorangehenden Algorithmus auf folgende Produktionsmenge:

$$\begin{array}{l} E \rightarrow E + T \mid T \\ T \rightarrow T * F \mid F \\ F \rightarrow (E) \mid a \end{array}$$

In Schritt (1) ergibt sich:  $V_E = \{E, T, F\}, V_T = \{T, F\}, V_F = \{F\}$ .

In Schritt (2) erhält man für  $P'$ :

$$\begin{array}{l} E \rightarrow E + T \mid T * F \mid (E) \mid a \\ T \rightarrow T * F \mid (E) \mid a \\ F \rightarrow (E) \mid a \end{array}$$

**Definition 7.17** (zyklenfrei, eigentlich)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik.

(a)  $G$  heißt **zyklenfrei**, wenn  $A \xrightarrow{G}^+ A$  für kein  $A \in V$  gilt.

(b)  $G$  heißt **eigentlich**, wenn  $G$  der  $e$ -Bedingung genügt, keine singulären Produktionen und keine nutzlosen Buchstaben enthält.

### Bemerkung 7.18

(i)  $G$  enthält keine singuläre Produktion und erfüllt die  $e$ -Bedingung  $\implies G$  ist zyklenfrei.

(ii) Ist  $G$  eigentlich, dann ist  $G$  auch zyklenfrei.

**Satz 7.19** (Konstruktion einer eigentlichen Grammatik)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik mit  $L(G) \neq \emptyset$ . Dann kann man eine eigentliche, kontextfreie Grammatik  $G'$  mit  $L(G) = L(G')$  konstruieren.

- Bew.: (a) Konstruiere  $G_1$  mit  $\epsilon$ -Bedingung und  $L(G_1) = L(G)$ .  
(b) Konstruiere  $G_2$  ohne singuläre Produktionen und  $L(G_2) = L(G_1)$ .  
(c) Konstruiere  $G'$  ohne nutzlose Buchstaben mit  $L(G') = L(G_2)$ .

□

# Kapitel 8

## Chomsky–Normalform, Homomorphie kontextfreier Sprachen

Chomsky, Avram Noam, \* 1928

Studium: Linguistik, Mathematik, Philosophie

seit 1955 Prof. für Sprachwissenschaften am MIT

Führte den Begriff der *Typ-0-* oder *Phrasenstrukturgrammatik* ein.

**Definition 8.1** (Chomsky–Normalform)

Eine kontextfreie Grammatik  $G = (V, X, S, P)$  ist in **Chomsky–Normalform**, wenn sie der  $\epsilon$ -Bedingung genügt, und jede Produktion, außer evtl.  $S \rightarrow \epsilon$ , von der Form  $A \rightarrow BC$  oder  $A \rightarrow a$  ist, wobei  $A, B, C \in V$  und  $a \in X$  sind.

**Satz 8.2** (Konstruktion der CNF)

Jede kontextfreie Sprache kann durch eine kontextfreie Grammatik in Chomsky–Normalform (CNF) erzeugt werden.

Bew.: Sei  $L$  eine kontextfreie Sprache.

Sei  $G_1 = (V, X, S, P)$  eine kontextfreie Grammatik, die der  $\epsilon$ -Bedingung genügt, die keine singulären Produktionen enthält und für die  $L = L(G_1)$  gilt, also:

- Die rechten Seiten aller Produktionen aus  $P$  sind  $\neq \epsilon$  (außer evtl.  $S \rightarrow \epsilon$ )
- Falls die rechte Seite einer Produktion die Länge 1 hat, ist diese eine Konstante.

Sei  $A \rightarrow X_1 X_2 \dots X_m, m \geq 2$ , eine Produktion aus  $P$ .

Falls  $X_i$  die Konstante  $a$  ist, führe eine neue Variable  $C_a$  und Produktion  $C_a \rightarrow a$  ein. Ersetze  $X_i$  in der obigen Produktion durch  $C_a$ .

Behandle alle Produktionen aus  $P$ , deren rechte Seiten länger als 1 sind, auf diese Weise.

$V'$ : so erhaltene (neue) Variablenmenge (Erweiterung von  $V$ )

$P'$ : so erhaltene (neue) Produktionenmenge zusammen mit allen  $A \rightarrow a \in P, |a| \leq 1$ .

Setze  $G_2 = (V', X, S, P')$ . Es ist  $L(G_1) = L(G_2)$ .

Es wird jetzt  $G_2$  folgendermaßen modifiziert:

Sei  $A \rightarrow B_1 \dots B_m, |m| \geq 3$  eine Produktion aus  $P'$ .

Führe (für diese Produktion) neue Variablen  $D_1, D_2, \dots, D_{m-2}$  und ersetze  $A \rightarrow B_1 \dots B_m$  durch folgende Menge von Produktionen:  $\{A \rightarrow B_1 D_1, D_1 \rightarrow B_2 D_2, \dots, D_{m-3} \rightarrow B_{m-2} D_{m-2}, D_{m-2} \rightarrow B_{m-1} B_m\}$

$V''$ : so erhaltene (neue) Variablenmenge.

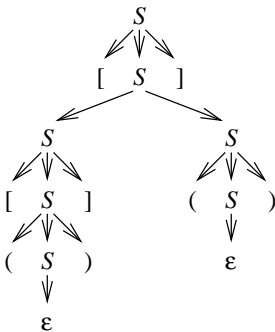
$P''$ : so erhaltene (neue) Produktionenmenge.

Setze  $G_3 = (V'', X, S, P'')$ . Es gilt  $L(G_3) = L(G_2) = L(G_1) = L$  und  $G_3$  ist in CNF. □

Für jedes  $n \geq 1$  sei  $X_n$  das folgende Alphabet:  $X_n = \{a_1, a'_1, a_2, a'_2, \dots, a_n, a'_n\}$ .

$D_n$  sei die durch die Grammatik  $G_D^n = (\{S\}, X_n, S, \{S \rightarrow \epsilon, S \rightarrow SS, S \rightarrow a_1 S a'_1, \dots, S \rightarrow a_n S a'_n\})$  erzeugte Sprache.

Ableitungsbaum für das Wort  $[[()]]()$ :



Bew.: Sei  $L \subseteq X^*$  eine kontextfreie Sprache.

Zunächst wird  $e \notin L$  angenommen.

Sei  $G = (V, X, S_0, P)$  eine Grammatik in CNF mit  $L = L(G)$ .

Sei  $X = \{a_1, \dots, a_m\}$  und seien  $A_i \rightarrow B_i C_i, i \in [1 : r]$ , alle Produktionen aus  $P$  mit Variablen auf der rechten Seite.

Betrachte  $X_{m+r} = \{a_1, a'_1, \dots, a_m, a'_m, a_{m+1}, a'_{m+1}, \dots, a_{m+r}, a'_{m+r}\}$

Definiere einen Homomorphismus  $h : \begin{cases} X_{m+r} \rightarrow X^* \\ a_i \mapsto \begin{cases} a_i & \text{für } i \in [1 : m] \\ e & \text{sonst.} \end{cases} \end{cases}$

Konstruiere folgende Grammatik  $G_1 = (V, X_{m+r}, S_0, P_1)$  mit

$P_1 = \{A \rightarrow aa' \mid A \rightarrow a \in P\} \cup \{A \rightarrow aa'a'_{m+i} C_i \mid A \rightarrow a \in P, i \in [1 : r]\} \cup \{A_i \rightarrow a_{m+i} B_i \mid i \in [1 : r]\}$

Es ist  $L(G_1)$  eine reguläre Sprache  $R$ .

Es wird gezeigt:  $L = h(D_{m+r} \cap R)$ .

$L \subseteq h(D_{m+r} \cap R)$ .

Sei  $A = v_0 \xrightarrow{G} v_1 \xrightarrow{G} \dots \xrightarrow{G} v_t = v$  eine Ableitung eines Wortes  $v \in X^*$  aus  $A \in V$ .

Durch Induktion über  $t$  wird gezeigt:  $\exists w \in D_{m+r}$  mit  $A \xrightarrow{G_1}^* w$  und  $h(w) = v$ .

$t = 1$ :  $A \xrightarrow{G} a$ . Es gilt  $A \xrightarrow{G_1} aa'$  und  $h(aa') = a$ .

$t \rightarrow t+1$ :  $A = v_0 \xrightarrow{G} v_1 \xrightarrow{G} \dots \xrightarrow{G} v_{t+1}$ .

Im ersten Schritt möge die Produktion  $A_i \rightarrow B_i C_i$  angewendet worden sein.

Weiter gibt es Wörter  $v'_1, v'_2$  mit  $B_i \xrightarrow{G}^{\leq t} v'_1, C_i \xrightarrow{G}^{\leq t} v'_2$  und  $v_{t+1} = v'_1 \cdot v'_2$ .

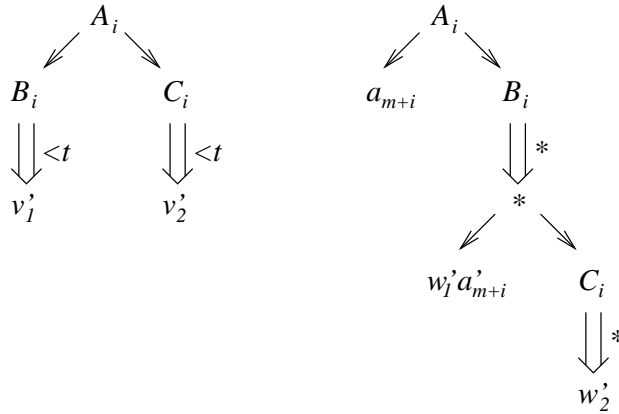
Aufgrund der Induktionsvoraussetzung  $\exists w'_1, w'_2 \in D_{m+r}$  mit  $B_i \xrightarrow{G_1}^* w'_1, C_i \xrightarrow{G_1}^* w'_2$  und  $h(w'_1) = v'_1$  und  $h(w'_2) = v'_2$ .

O.B.d.A.  $B_i \xrightarrow{G_1}^{\leq t} w'_1$  ist Linksableitung. Die letzte angewendete Produktion in dieser Ableitung sei

$D \rightarrow dd'$ . Ersetze diese Produktion durch  $D \rightarrow dd'a'_{m+i} C_i$ .

Dann ist bzgl.  $G_1$  folgendes möglich:  $A_i \xrightarrow{G_1} a_{m+i} B_i \xrightarrow{G_1}^* a_{m+i} w'_1 a'_{m+i} C_i \xrightarrow{G_1}^* a_{m+i} w'_1 a'_{m+i} w'_2$ .

Es ist  $a_{m+i} w'_1 a'_{m+i} w'_2 \in D_{m+r}$  und es gilt  $h(a_{m+i} w'_1 a'_{m+i} w'_2) = h(w'_1) h(w'_2) = v'_1 v'_2 = v_{t+1}$ .



$h(D_{m+r} \cap R) \subseteq L$ .

Durch Induktion über  $u$  wird folgendes gezeigt: Falls  $A = w_0 \xrightarrow{G_1} w_1 \xrightarrow{G_1} \dots \xrightarrow{G_1} w_u = w$  mit  $w \in D_{m+r}$  gilt, dann gilt

auch  $A \xrightarrow{G}^* h(w)$ .

$u = 1$ :  $A \xrightarrow{G_1} w$ , also  $A \xrightarrow{G_1} aa', aa' \in D_{m+r}, A \rightarrow a \in P$ .

$u \rightarrow u+1$ : Sei  $A = w_0 \xrightarrow{G_1} w_1 \xrightarrow{G_1} \dots \xrightarrow{G_1} w_{u+1} = w, w \in D_{m+r}$ .

Die erste angewendete Produktion kann nicht von der Form  $A \rightarrow aa'$  sein, und auch nicht von der Form  $A \rightarrow aa'a'_{m+i} C_i$  (dann wäre  $w \notin D_{m+r}$ ).

Also: Die erste angewendete Produktion ist  $A = A_i \rightarrow a_{m+i} B_i$ .

Es gibt also eine Ableitung  $\tilde{w}_1 = B_i \xrightarrow{G_1} \tilde{w}_2 \xrightarrow{G_1} \tilde{w}_3 \xrightarrow{G_1} \dots \xrightarrow{G_1} \tilde{w}_{u+1}$  und  $w = a_{m+i} \tilde{w}_{u+1}$ .

Es ist  $w = a_{m+i} \tilde{w}'_k a'_{m+i} \tilde{v}$  mit  $\tilde{w}, \tilde{v} \in D_{m+r}$ . Es gibt ein  $k$  mit  $2 \leq k \leq u$ , so daß  $\tilde{w}_k = \tilde{w}'_k a'_{m+i} C_i$  gilt.

Weiterhin muß bei  $\tilde{w}_{k-i} \xrightarrow{G_1} \tilde{w}_k$  die Produktion  $B \rightarrow bb'a'_{m+i} C_i$  angewendet worden sein.

Hier ist  $B \rightarrow b \in P$ .

Wende nicht  $B \rightarrow bb'ba'_{m+i} C_i$  sondern  $B \rightarrow bb'$  an.

Man erhält eine Ableitung von  $\tilde{w}$  aus  $B_i$  mit  $< u$  Schritten.

Entsprechend gibt es eine Ableitung von  $\tilde{v}$  aus  $C_i$  mit  $< u$  Schritten.

Aufgrund der IV gilt dann:  $B_i \xrightarrow{G}^* h(\tilde{w}), C_i \xrightarrow{G}^* h(\tilde{v})$ , also insgesamt:

$$A = A_i \xrightarrow{G} B_i C_i \xrightarrow{G}^* h(\tilde{w}) \cdot h(\tilde{v}) = h(\tilde{w} \cdot \tilde{v}) = h(w).$$

Ist  $e \in L$ , dann ist  $L \setminus \{e\}$  auch kontextfrei. Es ist  $L \setminus \{e\} = h(D_{m+r} \cap R)$ .

Hier muß gelten  $e \notin R$ . Es ist dann  $L = h(D_{m+r} \cap (R \cup \{e\}))$ .

$R \cup \{e\}$  ist wiederum regulär. □

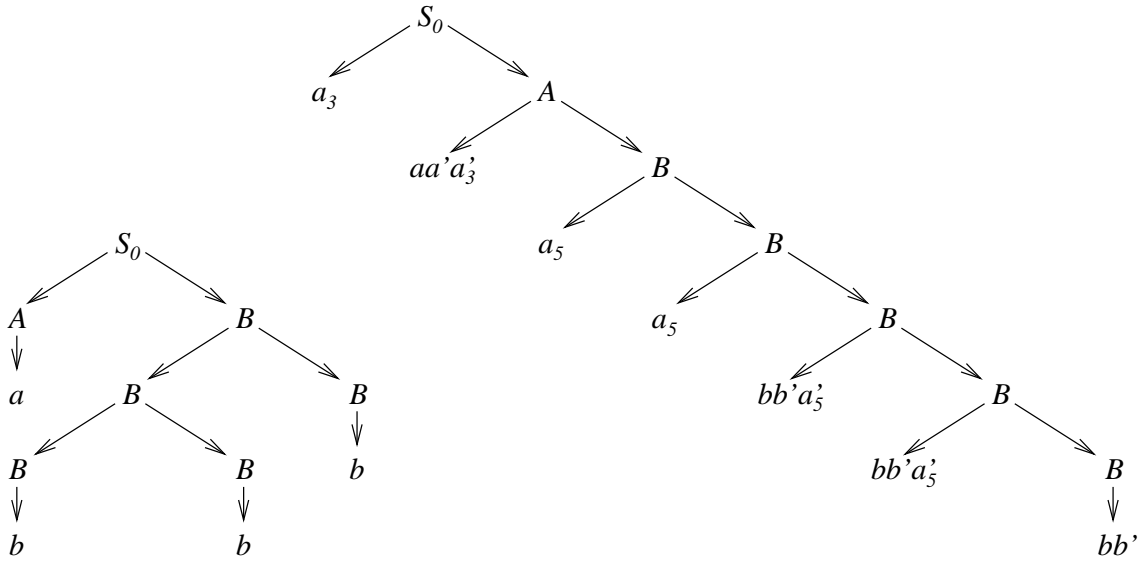
### Beispiel 8.4

$G = (\{S, A, B\}, \{a, b\}, S_0, P)$  mit

$$P = \left\{ \begin{array}{ll} 1. S_0 \rightarrow AB & A \rightarrow a \\ 2. A \rightarrow AA & B \rightarrow b \\ 3. B \rightarrow BB & \end{array} \right\} \quad L(G) = \{a^i b^j \mid i, j \geq 1\}$$

$m = 2, r = 3$ .

Konstruktion von  $G_1 = (\{S_0, A, B\}, X_5, S_0, P_1)$ .  $X_5 = \{a, a', b, b', a_3, a'_3, a_4, a'_4, a_5, a'_5\}$ .



Ableitungsbaum von  $abb$ .

Ausgehend von der nebenstehenden Ableitung wird eine Ableitung von  $a_3aa'a'_3a_5bb'a'_5bb'a'_5bb'$  angegeben.

$$P_1 = \{A \rightarrow aa', B \rightarrow bb'\} \cup \{A \rightarrow aa'a'_3B, A \rightarrow aa'a'_4A, A \rightarrow aa'a'_5B, B \rightarrow bb'a'_3B, B \rightarrow bb'a'_4A, B \rightarrow bb'a'_5B\} \cup \{S_0 \rightarrow a_3A, A \rightarrow a_4A, B \rightarrow a_5B\}$$

# Kapitel 9

## Der Iterationssatz für kontextfreie Sprachen

### Satz 9.1 (Iterationssatz)

$L$  sei eine kontextfreie Sprache. Es gibt eine ganze Zahl  $k \geq 1$ , so daß sich jedes  $z \in L$  mit  $|z| \geq k$  in der Form  $z = uvwxy$  schreiben läßt, wobei  $vx \neq \epsilon$ ,  $|vwx| \leq k$  und  $uw^iwx^i y \in L$  für alle  $i \geq 0$ .

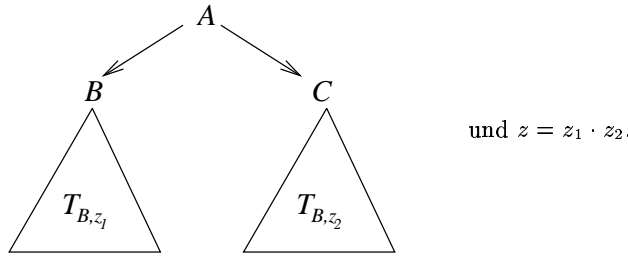
Bew.:  $G = (V, X, S, P)$  sei eine Grammatik in CNF mit  $L = L(G)$ .

Für alle  $A \in V$  setze  $G_A = (V, X, A, P)$ . (Es ist  $L = L(G_S)$ ).

Es gilt: Sei  $z \in L(G_A)$  und  $T_{A,z}$  ein Ableitungsbaum von  $z$  bzgl.  $G_A$ . Hat jeder Weg von der Wurzel zu einem Blatt in  $T_{A,z}$  eine Länge  $\leq n$ , dann gilt  $|z| \leq 2^{n-1}$ .

$n \equiv 1$ :  $T_{A,z}$   $A \rightarrow a = z$  oder  $S \rightarrow \epsilon = z$ .

$n \rightarrow n + 1$ :  $T_{A,z}$ .



Jeder Weg von  $A$  zu einem Blatt hat eine Länge  $\leq n + 1$ , also jeder Weg von  $B$  oder  $C$  zu einem Blatt hat eine Länge  $\leq n$ , daher  $|z_1|, |z_2| \leq 2^{n-1}$  und  $|z| \leq 2 \cdot 2^{n-1} = 2^n$ .

Sei  $V = \{A_1, A_2, \dots, A_n\}$ . Setze  $k = 2^n$ . Sei  $z \in L(G)$  mit  $|z| \geq k$  und  $T = T_{S,z}$  ein Ableitungsbaum für  $z$  bzgl.  $G$ .

Sei  $p$  ein Weg mit maximaler Länge, die  $\geq n + 1$  ist. (Wegen  $|z| > 2^{n+1}$  hat  $p$  eine Länge, die  $\geq n + 1$  ist.)

Sei  $A_m = S, A_{m-1}, \dots, A_2, A_1, a$  die Folge der Knoten in  $p$  ausgehend von  $S$ . Es ist  $m \geq n + 1$ . Unter den  $A_{n+1}, A_n, \dots, A_2, A_1$  kommt eine Variable doppelt vor:  $A = A_i = A_j$  mit  $n + 1 \geq i > j \geq 1$ .

$T_i$  sei Teilbaum von  $T$  mit Wurzel  $A_i$ .

Jeder Weg in  $T_i$  von der Wurzel zu einem Blatt hat eine Länge  $\leq i$ , da sonst  $p$  nicht maximal in  $T$  wäre, also:  $T_i$  ist Ableitungsbaum eines Wortes  $z_1$  mit  $|z_1| \leq 2^{i-1} \leq 2^n = k$ .

$T_j$  sei Teilbaum von  $T$  mit Wurzel  $A_j$ .  $T_j$  ist Ableitungsbaum eines Wortes  $z_2$  und es gilt:  $z_1 = z_3 z_2 z_4 = vwx$ .

$T_i$  muß mit  $A_i \rightarrow A_{i+1}B$  oder  $A_i \rightarrow BA_{i+1}$  beginnen. Also kann nicht  $z_3 z_4 = vx = \epsilon$  gelten.

Es gilt also  $A = A_i \xrightarrow{*} \underbrace{z_3 A_j z_4}_{v A_j x} = z_3 A z_4$  und  $A = A_j \xrightarrow{*} z_2 = w$ .

Weiter gilt:  $S \xrightarrow{*} u A_i y = u A_j y$ . □

### Beispiel 9.2

$L = \{a^{n^2} | n \geq 1\}$  ist keine kontextfreie Sprache.

Ann.:  $L$  ist kontextfrei. Wähle  $k$  wie im Iterationslemma.

Setze  $n = k$  (dann ist  $n^2 \geq k$ )

Zerlege  $a^{n^2} = uvwxy$  wie im Iterationslemma,  $vx \neq \epsilon$ ,  $|vwx| \leq k$ ,  $1 \leq |vx| \leq k$ .

Es ist  $uw^2wx^2y \in L$  aufgrund der Annahme.

Es ist dann:  $k^2 < |uw^2wx^2y| \leq k^2 + k$

Die nächste auf  $k^2$  folgende Quadratzahl ist  $(k + 1)^2 = k^2 + 2k + 1 > k^2 + k$  (Widerspruch!)

### Beispiel 9.3

$L = \{a^n b^n c^n | n \geq 1\}$  ist keine kontextfreie Sprache.

Ann.:  $L$  ist kontextfrei. Wähle  $k$  wie im Iterationslemma.

$a^k b^k c^k = uvwxy$  wie im Iterationslemma.

In  $vwx$  können nicht gleichzeitig  $a$ 's,  $b$ 's und  $c$ 's vorkommen.

Es ist  $uvw y \in L$  aufgrund der Annahme.

1. In  $vwx$  kommen nur  $a$ 's und  $b$ 's vor.

Dann kommen in  $uvw y$   $k$   $c$ 's vor. (Widerspruch!)

2. In  $vwx$  kommen nur  $b$ 's oder  $c$ 's vor.

Dann kommen in  $uvw y$   $k$   $a$ 's vor. (Widerspruch!)

### Beispiel 9.4

$L = \{a^i b^j c^j \mid j \leq i\}$  ist nicht kontextfrei.

Ann.:  $L$  ist kontextfrei. Wähle  $k$  wie oben.

Schreibe  $a^k b^k c^k = uvwxy$  wie im Iterationslemma.

Falls  $vwx$  nur  $a$ 's oder  $b$ 's enthält, dann führt  $uwy \in L$  auf einen Widerspruch, da  $uwy$  weniger  $a$ 's oder  $b$ 's als  $c$ 's enthält.

Falls  $vwx$  nur  $b$ 's oder  $c$ 's enthält, dann führt  $uv^2wx^2y \in L$  auf einen Widerspruch.

### Beispiel 9.5

$L = \{a^i b^j c^l \mid i < j < l\}$  ist nicht kontextfrei.

Ann.:  $L$  ist kontextfrei. Wähle  $k$  wie oben.

Schreibe  $a^k b^{k+1} c^{k+2} = uvwxy$  wie oben.

Falls  $vwx$  nur  $a$ 's oder  $b$ 's enthält, dann führt  $uv^iwx^i y \in L$  für hinreichend großes  $i$  auf einen Widerspruch.

Falls  $vwx$  nur  $b$ 's oder  $c$ 's enthält, dann:

(1)  $vx$  enthält ein  $b$ , dann führt  $uwy \in L$  auf einen Widerspruch.

(höchstens so viele  $b$ 's wie  $a$ 's.)

(2)  $vx$  enthält ein  $c$  (und keine  $b$ 's), dann führt  $uwy \in L$  ebenfalls auf einen Widerspruch.

# Kapitel 10

## Rekursivität und die Greibach–Normalform

### Satz 10.1

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik und  $A \rightarrow \alpha B \beta, B \in V$  sei eine Produktion aus  $P$ .  $B \rightarrow \gamma_1 | \gamma_2 | \dots | \gamma_k$  seine alle  $B$ -Produktionen aus  $G$ .

Setze  $P' = (P \setminus \{A \rightarrow \alpha B \beta\}) \cup \{A \rightarrow \alpha \gamma_1 \beta | \alpha \gamma_2 \beta | \dots | \alpha \gamma_k \beta\}$  und  $G' = (V, X, S, P')$ . Dann ist  $L(G) = L(G')$ .

### Beispiel 10.2

$G = (\{S\}, \{a, b\}, S, P)$  und  $P = \{S \rightarrow aSS|b\}$ .

Ersetze in  $S \rightarrow aSS$  das linke  $S$  durch alle rechten Seiten von  $S$ -Produktionen.

$P' = \{S \rightarrow aaSSS|abS|b\}$ .

### Definition 10.3 (Rekursivität)

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik.

$A \in V$  heißt **rekursiv**, wenn  $A \xrightarrow[G]{\neq} \alpha A \beta$  mit  $\alpha, \beta \in (V \cup X)^*$  gilt.

$A$  heißt **linksrekursiv**, wenn hierbei  $\alpha = e$  ist.

$A$  heißt **rechtsrekursiv**, wenn hierbei  $\beta = e$  gilt.

$G$  heißt **linksrekursiv** bzw. **rechtsrekursiv**, wenn es mindestens eine Variable in  $V$  gibt, die linksrekursiv bzw. rechtsrekursiv ist.

### Satz 10.4

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik und  $A \rightarrow A\alpha_1|A\alpha_2|\dots|A\alpha_m|\beta_1|\dots|\beta_n$  seien alle  $A$ -Produktionen von  $G$ , wobei kein  $\beta_i$  mit  $A$  beginnt.

Setze  $G' = (V \cup \{A'\}, X, S, P')$ . Hierbei ist  $A'$  eine neue Variable und  $P'$  entsteht aus  $P$  durch Ersetzen der  $A$ -Produktionen durch die Produktionen

$A \rightarrow \beta_1 | \dots | \beta_n | \beta_1 A' | \dots | \beta_n A'$

$A' \rightarrow \alpha_1 | \dots | \alpha_m | \alpha_1 A' | \dots | \alpha_m A'$

Dann ist  $L(G) = L(G')$ .

Bew.: (1)  $L(G) \subseteq L(G')$

Sei  $w \in L(G)$  und  $S \xrightarrow[*]{lG} w$  sei eine Linksableitung von  $w$  bzgl.  $G$ .

Ersetze jede in der Linksableitung auftretende Folge maximaler Länge der Form

$$(*) \quad vA\gamma \xrightarrow[G]{\Rightarrow} vA\alpha_{i_1}\gamma \xrightarrow[G]{\Rightarrow} vA\alpha_{i_2}\alpha_{i_1}\gamma \xrightarrow[G]{\Rightarrow} \dots \xrightarrow[G]{\Rightarrow} vA\alpha_{i_k}\dots\alpha_{i_2}\alpha_{i_1}\gamma \xrightarrow[G]{\Rightarrow} v\beta_{i_1}\alpha_{i_k}\dots\alpha_{i_2}\alpha_{i_1}\gamma \quad (k \geq 1)$$

durch  $vA\gamma \xrightarrow[G']{\Rightarrow} v\beta_1 A'\gamma \xrightarrow[G']{\Rightarrow} v\beta_1 \alpha_{i_k} A'\gamma \xrightarrow[G']{\Rightarrow} \dots \xrightarrow[G']{\Rightarrow} v\beta_1 \alpha_{i_k} \dots \alpha_{i_2} A'\gamma \xrightarrow[G']{\Rightarrow} v\beta_1 \alpha_{i_k} \dots \alpha_{i_2} \alpha_{i_1} \gamma$ .

Man erhält so eine Ableitung (nicht notwendig Linksableitung) von  $w$  bzgl.  $G'$ . Also ist  $L(G) \subseteq L(G')$ .

(2)  $L(G') \subseteq L(G)$

Sei  $w \in L(G')$  und  $S \xrightarrow[*]{rG'} w$  sei eine Rechtsableitung bzgl.  $G'$ .

Ersetze jede in dieser Ableitung vorkommende Teilfolge

$$\gamma A v \xrightarrow[G']{\Rightarrow} \gamma \beta_1 A' v \xrightarrow[G']{\Rightarrow} \gamma \beta_1 \alpha_{i_k} A v \xrightarrow[G']{\Rightarrow} \dots \xrightarrow[G']{\Rightarrow} \gamma \beta_1 \alpha_{i_k} \dots \alpha_{i_2} A' v \xrightarrow[G']{\Rightarrow} \gamma \beta_1 \alpha_{i_k} \dots \alpha_{i_2} \alpha_{i_1} v \text{ durch } (*).$$

Man erhält so eine Ableitung (nicht notwendig Rechtsableitung) von  $w$  bzgl.  $G$ . Also ist  $L(G') \subseteq L(G)$ . □

### Beispiel 10.5

Betrachte  $G = (V, X, S, P)$  mit  $P = \{E \rightarrow E + T | T, T \rightarrow T * F | F, F \rightarrow (E) | \alpha\}$

Anwendung der Konstruktion des vorangehenden Satzes auf die  $E$ - und  $T$ -Produktionen:

$$P' = \left\{ \begin{array}{l|l} E \rightarrow T & TE' \\ E' \rightarrow +T & +TE' \\ T \rightarrow F & FT' \\ T' \rightarrow *F & *FT' \\ F \rightarrow (E) & \alpha \end{array} \right\}$$

$$G' = (V \cup \{E', T'\}, X, S, P')$$

## Algorithmus 10.6 (Elimination von Linksrekursivität)

Eingabe :  $G = (V, X, S, P)$  kontextfreie Grammatik ohne singuläre Produktionen und ohne  $\epsilon$ -Produktionen

Ausgabe :  $G' = (V', X, S, P')$  kontextfrei mit  $L(G) = L(G')$ , und keine Variable von  $G'$  ist linksrekursiv.

Methode : (1)  $V = \{A_1, \dots, A_n\}$  und  $i := 1$

(1a)  $G_i := G$

(2)  $A_i \rightarrow A_i\alpha_1 | \dots | A_i\alpha_m | \beta_1 | \dots | \beta_p$  seine die  $A_i$ -Produktionen von  $G_i$ . Ersetze diese durch:

$A_i \rightarrow \beta_i | \dots | \beta_p | \beta_1 A'_i | \dots | \beta_p A'_i \quad A'_i \rightarrow \alpha_1 | \dots | \alpha_m | \alpha_1 A'_i | \dots | \alpha_m A'_i$

( $A'_i$  neu eingeführtes Symbol)

(2a)  $G'_i$  ist die so aus  $G_i$  erhaltene Grammatik.

(3) Falls  $i = n$ , setze  $G' = G'_i$ , stop

sonst  $i := i + 1$  und  $j := 1$

(3a)  $G'_i := G'_{i-1}$

(4) Jede Produktion der Form  $A_i \rightarrow A_j\alpha$  aus  $G'_i$  werde ersetzt durch  $A_i \rightarrow \beta_1\alpha | \dots | \beta_m\alpha$ , wobei

$A_j \rightarrow \beta_1 | \dots | \beta_m$  alle  $A_j$ -Produktionen aus  $G'_i$  sind.

(4a)  $G'_i$  ist die aus  $G'_i$  erhaltene Grammatik.

(5) Falls  $j = i - 1$ , setze  $G_i := G'_i$  und gehe zu (2)

sonst setze  $j := j + 1$  und gehe zu (4).

## Satz 10.7

$G'$  aus 10.6 ist kontextfrei, es gilt  $L(G) = L(G')$  und  $G'$  besitzt keine linksrekursiven Variablen.

Bew.:  $L(G) = L(G')$  (siehe 10.1 und 10.6)

Durch Induktion über  $i$  wird folgendes gezeigt: Wird (2) mit  $i$  und dann (4) mit  $i + 1$  für  $j = 1 \dots i$  durchlaufen, dann gilt:

a) Nach Durchlauf von (2) haben alle  $A_i$ -Produktionen rechte Seiten, die mit einer Konstanten oder einem  $A_k$  mit  $k > i$  beginnen.

b) Nach Durchlauf von (4) für  $j = 1 \dots i$  haben alle  $A_{i+1}$ -Produktionen rechte Seiten, die mit einer Konstanten oder einem  $A_k$  mit  $k > j$  beginnen.

$i = 1$ :

a) In (2) beginnen alle  $\beta_i$  mit einer Konstanten oder einem  $A_k$  mit  $k > 1$ . Das gilt dann auch nach Ausführung von (2) für alle  $A_1$ -Produktionen.

b)  $i + 1 = 2$ . Die rechten Seiten aller  $A_1$ -Produktionen beginnen nach a) mit einer Konstanten oder einem  $A_k$  mit  $k > 1$ .

Das gilt dann nach Ausführung von (4) für  $j = 1$  auch für alle  $A_2$ -Produktionen.

$i \rightarrow i + 1$ :

a) Vor Ausführung von (2) mit  $i + 1$  gilt, daß alle  $\beta_1 \dots \beta_p$  mit einer Konstanten oder einem  $A_k$  mit  $k > i + 1$  beginnen (siehe Induktionsvoraussetzung). Das gilt dann auch nach Ausführung von (2) für alle  $A_{i+1}$ -Produktionen.

b) (4) wird mit  $i + 2$  für  $j = 1 \dots i + 1$  durchlaufen.

$j = 1$ : Die rechten Seiten aller  $A_1$ -Produktionen beginnen mit einer Konstanten oder einem  $A_k$  mit  $k > 1$ . Dies gilt dann auch nach Ausführung von (4) für  $j = 1$  für die rechten Seiten aller  $A_{i+2}$ -Produktionen.

$j \rightarrow j + 1$ : Vor Ausführung von (4) mit  $j + 1$  beginnen die rechten Seiten aller  $A_{i+2}$ -Produktionen mit einer Konstanten oder einem  $A_k$  mit  $k \geq j + 1$ .

Es gilt (siehe a) ): Die rechten Seiten aller  $A_{j+1}$ -Produktionen beginnen mit einer Konstanten oder einem  $A_k$  mit  $k > j + 1$ . Das gilt dann auch nach Ausführung von (4) mit  $j + 1$  für alle rechten Seiten aller  $A_{i+2}$ -Produktionen.

In  $G'$  ist kein  $A_i$  linksrekursiv, dazu:

Ann.:  $A_i$  ist bzgl.  $G'$  linksrekursiv und es gilt  $A_i = w_0 \xrightarrow{G'} w_1 \xrightarrow{G'} \dots \xrightarrow{G'} w_m = A_i\alpha, m \geq 1$ .

Alle  $w_j, j \in [0, m]$ , müssen mit einer Variablen aus  $V$  beginnen (da in (2)  $\beta_j \neq \epsilon \quad \forall j$ ). Diese seine  $A_i, A_{k_1}, A_{k_2}, \dots, A_{k_m} = A_i$ .

Es ist  $i < k_1 \leq k_2 \leq \dots \leq k_m = i$ . Da  $m \geq 1 \Rightarrow i < i$  (Widerspruch!)

Es gilt: Vor Ausführung von (2) beginnen die rechten Seiten aller  $A_i$ -Produktionen mit einer Konstanten oder zwei Buchstaben aus  $V \cup X$ .

Also: Die rechten Seiten aller  $A'_i$ -Produktionen beginnen mit einem Buchstaben aus  $V \cup X$ , also:

Keine rechte Seite einer  $A'_i$ -Produktion beginnt mit  $A'_i$ . □

## Korollar 10.8

Jede kontextfreie Sprache  $L \neq \emptyset$  wird von einer eigentlichen, kontextfreien Grammatik erzeugt, die nicht linksrekursiv ist.

## Beispiel 10.9

Betrachte  $G = (\{A, B, C\}, \{a, b\}, A, P)$  mit  $P = \{A \rightarrow BC|a, B \rightarrow CA|Ab, C \rightarrow AB|CC|a\}$ .

Es ergeben sich nach Anwendung von (2) und (4) des Algorithmus folgende Produktionen:

(2)  $i = 1$ :  $A \rightarrow BC|a| \underbrace{BCA'}_{\text{nutzlos!!!}}|aA'$

(4)  $i = 2, j = 1$ :  $B \rightarrow BCB|ab|CA$

(2)  $i = 2$ :  $B \rightarrow ab|CA|abB'|CAB'$   
 $B' \rightarrow Cb|CbB'$

(4)  $i = 3, j = 1$ :  $C \rightarrow BCB|aB|CC|a$

$j = 2$ :  $C \rightarrow CACB|abCB|CAB'CB|abB'CB|ab|CC|a$

(2)  $i = 3$ :  $C \rightarrow abCB|abB'CB|aB|a|abCBC'|abB'CBC'|aBC'|aC'$   
 $C' \rightarrow ACB|AB'CB|C|ACBC'|AB'CBC'|CC'$

### Definition 10.10 (Greibach-Normalform)

Eine kontextfreie Grammatik  $G = (V, X, S, P)$  ist in **Greibach-Normalform** (GNF), wenn  $G$  der  $e$ -Bedingung genügt und jede Produktion aus  $P$ , außer evtl.  $S \rightarrow e$ , von der Form  $A \rightarrow a\alpha, a \in X, \alpha \in V^*$  ist.

Ist  $A$  eine (endliche) Menge, dann heißt eine Relation  $R \subseteq A \times A$  **partielle Ordnung**, wenn gilt:

- (1)  $\forall a \in A : (a, a) \notin R$  (irreflexiv)
- (2)  $\forall a, b, c \in A : (a, b) \in R, (b, c) \in R \Rightarrow (a, c) \in R$  (transitiv)

Eine partielle Ordnung  $R$  heißt **lineare Ordnung**, wenn  $\forall a, b \in A$  gilt:

entweder  $a = b$  oder  $(a, b) \in R$  oder  $(b, a) \in R$ .

Schreibweise für  $(a, b) \in R$  :  $aRb$

### Algorithmus 10.11 (Konstruktion einer linearen aus einer partiellen Ordnung)

Eingabe : Partielle Ordnung  $R$  auf einer endlichen Menge  $A$

Ausgabe : Lineare Ordnung  $R'$  auf  $A$  mit  $R \subseteq R'$ .

Methode :  $R'$  kann angegeben werden durch eine Folge  $a_1, \dots, a_n$  von Elementen aus  $A$ , für die  $a_i R' a_j$  genau dann gilt, wenn  $i < j$  ist, und für die weiter  $A = \{a_1, \dots, a_n\}$  ist.

Konstruktion dieser Folge:

- (1)  $i := 0, A_i := A$  und  $R_i := R$
- (2) Falls  $A_i = \emptyset$  stellt  $a_1, a_2, \dots, a_{i-1}$  die lineare Ordnung  $R'$  dar. Stop.  
Sonst sei  $a_i$  ein Element aus  $A_i$ , für das  $aR_i a_i$  für kein  $a \in A_i$  gilt.
- (3) Setze  $A_{i+1} := A_i \setminus \{a_i\}$  und  $R_{i+1} := R_i \cap (A_{i+1} \times A_{i+1})$  und  $i := i + 1$  und fahre bei (2) fort.  
Die Ausgabe des Algorithmus ist offensichtlich eine lineare Ordnung  $R'$  auf  $A$  mit  $R \subseteq R'$ .

### Beispiel 10.12

$A = \{a, b, c, d\}, R = \{(a, b), (a, c), (a, d), (b, d), (c, d)\}$

$A_1 = \{a, b, c, d\}, R_1 = R, a_1 := a$

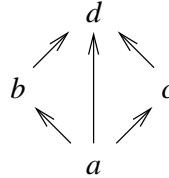
$A_2 = \{b, c, d\}, R_2 = \{(b, d), (c, d)\}, a_2 := b$

$A_3 = \{c, d\}, R_3 = \{(c, d)\}, a_3 := c$

$A_4 = \{d\}, R_4 = \emptyset, a_4 := d$

$A_5 = \emptyset$

lineare Ordnung: a,b,c,d



### Satz 10.13

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik, die nicht linksrekursiv ist. Dann gibt es eine lineare Ordnung  $<$  auf  $V$ , für die gilt: Ist  $A \rightarrow B\alpha$  aus  $P$ , dann ist  $A < B$ .

Bew.: Zunächst Festlegung von  $\bar{R} \subseteq V \times V : A\bar{R}B \iff A \rightarrow B\alpha \in P$

$R$  sei transitive Hülle von  $\bar{R}$ .

$R$  ist irreflexiv, da  $G$  nicht linksrekursiv ist.

Wende den vorangehenden Algorithmus aus  $R$  an.

Setze  $< = R'$ . □

### Algorithmus 10.14 (Transformation in GNF)

Eingabe :  $G = (V, X, S, P)$  sei kontextfrei, nicht linksrekursiv und erfülle  $e$ -Bedingung.

Ausgabe :  $G' = (V', X, S, P')$  kontextfrei in GNF und  $L(G) = L(G')$ .

Methode : (0) Setze  $\bar{G} = (V, X, S, \bar{P})$  mit  $\bar{P} = P \setminus \{S \rightarrow e\}$ .

- (1) Konstruiere eine lineare Ordnung  $<$  auf  $V$  entsprechend 10.11.

Dann sei  $V = \{A_1 < A_2 < \dots < A_n\}$ .

- (2)  $i := n - 1$

- (3) Falls  $i = 0$ , gehe zu (5). Sonst ersetze jede Produktion der Form  $A_i \rightarrow A_j\alpha$  (hierbei muß  $i < j$  sein) durch

$A_i \rightarrow \beta_1\alpha | \dots | \beta_m\alpha$ , wobei  $A_j \rightarrow \beta_1 | \dots | \beta_m$  alle  $A_j$ -Produktionen sind (hierbei beginnen alle  $\beta_1, \dots, \beta_m$  mit einer Konstanten).

- (4)  $i := i - 1$ , weiter bei (3)

- (5) Jetzt beginne die rechten Seiten aller Produktionen mit einer Konstanten. In jeder Produktion  $A \rightarrow aX_1 \dots X_k$  ersetze diejenigen  $X_j$ , die Konstanten sind, durch eine neue Variable  $X'_j$ .

- (6) Für alle in (5) eingeführten  $X'_j$  führe die Produktion  $X'_j \rightarrow X_j$  ein.  $\bar{G}' = (V', X, S, \bar{P}')$  sei die so erhaltene Grammatik.

- (7) Setze  $G' = (V', X, S, P')$  mit  $P' = \bar{P}' \cup \{S \rightarrow e\}$ , falls  $S \rightarrow e$  aus  $P$ ,  $P' = \bar{P}'$  sonst.

### Satz 10.15

Ist  $L$  eine kontextfreie Sprache, dann gibt es eine kontextfreie Grammatik  $G$  in GNF mit  $L = L(G)$ .

Bew.: Für  $i = n - 1, \dots, 1$  zeigt man:

Nach Ausführung von (3) für  $i$  beginnen die rechten Seiten alle  $A_i$ -Produktionen mit einer Konstanten.

Durch (5),(6),(7) wird dann eine Grammatik in GNF erzeugt.

### Beispiel 10.16

Gegeben sei  $G$  mit den Produktionen  $P : E \rightarrow T|TE', E' \rightarrow +T| +TE', T \rightarrow F|FT', T' \rightarrow *F| *FT', F \rightarrow (E)|a$ .

$<$  ist lineare Ordnung auf  $V$  mit:  $E' < E < T' < T < F$ . (Es gilt: Ist  $A \rightarrow B\alpha \in P$ , dann ist  $A < B$ .)

Die rechten Seiten alle  $F$  Produktionen beginnen mit einer Konstanten.

Ausführung von (3) in 10.14 für  $i = 4, 3, 2, 1$  ergibt:

$i = 4$ : Aus  $T \rightarrow FT'$  erhält man durch Ersetzen von  $F$ :  $T \rightarrow (E)a|(E)T'|aT'$ .

$i = 3$ : Die Produktionen  $T' \rightarrow *F|*FT'$  werden nicht geändert.

$i = 2$ : Aus  $E \rightarrow T|TE'$  erhält man durch Ersetzen von  $T$ :  $E \rightarrow (E)a|(E)T'|aT'| (E)E'|aE'| (E)T'E'|aT'E'$ .

$i = 1$ : Die Produktionen  $E' \rightarrow +T|+TE'$  werden nicht geändert.

In (5) wird eine neue Variable  $)'$  eingeführt und die Produktion  $)' \rightarrow )$ . Die resultierende Grammatik in GNF besitzt die Produktionen:

$$\begin{array}{lcl}
 E & \rightarrow & (E)' \quad | \quad a \quad \quad \quad | \quad (E)'T' \quad | \quad aT' \quad | \quad (E)'E' \quad | \quad aE' \quad | \quad (E)'T'E' \quad | \quad aT'E' \\
 E' & \rightarrow & +T \quad \quad \quad | \quad +TE' \\
 T & \rightarrow & (E) \quad \quad \quad | \quad a \quad \quad \quad | \quad (E)T' \quad \quad \quad | \quad aT' \\
 T' & \rightarrow & *F \quad \quad \quad | \quad *FT' \\
 F & \rightarrow & (E) \quad \quad \quad | \quad a \\
 )' & \rightarrow & )
 \end{array}$$

# Kapitel 11

## Kontextfreie Sprachen und Kellerautomaten

### Definition 11.1 (Kellerautomat)

Ein **Kellerautomat** (pda = pushdown automaton) ist ein 7-Tupel  $K = (Z, X, \Gamma, \delta, z_0, \gamma_0, F)$ .

- Hierbei ist:
1.  $Z \neq \emptyset$  eine endliche Menge (Zustandsmenge)
  2.  $X \neq \emptyset$  eine endliche Menge (Eingabealphabet)
  3.  $\Gamma \neq \emptyset$  eine endliche Menge (Kelleralphabet)
  4.  $\delta : Z \times (X \cup \{e\}) \times \Gamma \rightarrow \mathbb{P}_f(Z \times \Gamma^*)$  (Überföhrungsfunktion)
  5.  $z_0 \in Z$  (Anfangszustand)
  6.  $\gamma_0 \in \Gamma$  (initiales Kellerzeichen)
  7.  $F \subseteq Z$  (Menge der Endzustände)

Ein Tripel  $(z, w, \alpha) \in Z \times X^* \times \Gamma^*$  heißt **Konfiguration von  $K$** .

Interpretation:

1. Auf einem Eingabeband steht das Wort  $w \in X^*$ .
2. Das endliche Steuerwerk befindet sich im Zustand  $z$ .
3. Im Keller steht das Wort  $\alpha \in \Gamma^*$ .

Die Arbeitsweise eines pda's wird beschrieben durch eine binäre Relation auf den Konfigurationen von  $K$ .

Es gilt:  $\forall z, z' \in Z, \forall x \in X \cup \{e\}, \forall \gamma \in \Gamma, \forall w \in X^*, \forall \alpha, \beta \in \Gamma^*$  :

$(z, xw, \gamma\alpha) \vdash (z', w, \beta\alpha)$  g.d.w.  $(z', \beta) \in \delta(z, x, \gamma)$

$\beta = e$ : Kellerliste wird gekürzt.

$x = e$ : Vom Eingabeband wurde nichts gelesen:  **$e$ -Überföhrung**.

Wenn die Kellerliste leer ist, ist kein Übergang mehr möglich.

Relation  $\vdash^+$  auf den Konfigurationen von  $K$  ( $\vdash^+$  ist **transitive Hölle** von  $\vdash$ ):

$\kappa \vdash^+ \kappa'$  g.d.w.  $\exists \kappa_0, \kappa_1, \dots, \kappa_r, r \geq 1$  mit  $\kappa = \kappa_0, \kappa' = \kappa_r$  und  $\kappa_i \vdash \kappa_{i+1}$  für  $i \in [0 : r - 1]$ .

Relation  $\vdash^*$  auf den Konfigurationen von  $K$  ( $\vdash^*$  ist **reflexive, transitive Hölle** von  $\vdash$ ):

$\kappa \vdash^* \kappa'$  g.d.w.  $\kappa = \kappa'$  oder  $\kappa \vdash^+ \kappa'$ .

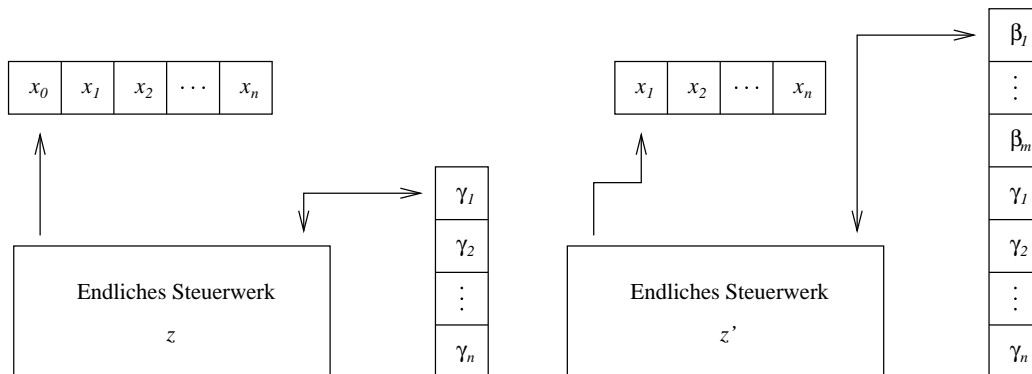
Anfangskonfiguration:  $(z_0, w, \alpha)$  mit  $w \in X^*$

Endkonfiguration:  $(z, e, \alpha)$  mit  $z \in F$

### Definition 11.2 (Sprache eines pda)

Ein Wort  $w \in X^*$  wird **von  $K^1$  akzeptiert**, wenn gilt:  $(z_0, w, \gamma_0) \vdash^* (z, e, \alpha)$  mit  $z \in F$ .

Die von  $K$  **akzeptierte Sprache** ist  $\mathcal{L}(K) = \{w \mid w \text{ wird von } K \text{ akzeptiert}\}$ .



<sup>1</sup>mit Endzuständen

### Beispiel 11.3

pda für die Sprache  $\{a^n b^n | n \geq 0\}$ .

Setze  $K = (\{z_0, z_1, z_2\}, \{a, b\}, \{\gamma_0, a\}, \delta, z_0, \gamma_0, \{z_0\})$  mit

$$\begin{aligned} \delta(z_0, a, \gamma_0) &= \{(z_1, a\gamma_0)\} & \delta(z_1, a, a) &= \{(z_1, aa)\} \\ \delta(z_1, b, a) &= \{(z_2, e)\} & \delta(z_2, b, a) &= \{(z_2, e)\} \\ \delta(z_2, e, \gamma_0) &= \{(z_0, e)\} \end{aligned}$$

Gebe  $w = aabb$  ein. Übergänge:

$$\begin{aligned} (z_0, aabb, \gamma_0) &\vdash (z_1, abb, a\gamma_0) \vdash (z_1, bb, aa\gamma_0) \\ &\vdash (z_2, b, a\gamma_0) \vdash (z_2, e, \gamma_0) \vdash (z_0, e, e) \end{aligned}$$

### Beispiel 11.4

pda  $K$  für die Sprache  $L = \{ww^R | w \in \{a, b\}^+\}$

Setze  $K = (\{z_0, z_1, z_2\}, \{a, b\}, \{\gamma_0, a, b\}, \delta, z_0, \gamma_0, \{z_2\})$  mit:

1.  $\delta(z_0, a, \gamma_0) = \{(z_0, a\gamma_0)\}$
2.  $\delta(z_0, b, \gamma_0) = \{(z_0, b\gamma_0)\}$
3.  $\delta(z_0, a, a) = \{(z_0, aa), (z_1, e)\}$
4.  $\delta(z_0, a, b) = \{(z_0, ab)\}$
5.  $\delta(z_0, b, a) = \{(z_0, ba)\}$
6.  $\delta(z_0, b, b) = \{(z_0, bb), (z_1, e)\}$
7.  $\delta(z_1, a, a) = \{(z_1, e)\}$
8.  $\delta(z_1, b, b) = \{(z_1, e)\}$
9.  $\delta(z_1, e, \gamma_0) = \{(z_2, e)\}$

Durch die Vorschriften 1,2,4,5 und die ersten Alternativen von 3,6 kopiert  $K$  Buchstaben des Eingabewortes in den Keller.  $K$  ist nicht deterministisch.

Die zweiten Alternativen von 3,6 ermöglichen den Übergang in den Zustand  $z_1$ .

Mit 7,8 wird die Kellerliste abgebaut, z.B.:  $w = abba$ . Es können folgende Überführungen durchgeführt werden:

$$\begin{aligned} (z_0, abba, \gamma_0) &\vdash (z_1, bba, a\gamma_0) \\ &\vdash (z_0, ba, ba\gamma_0) \\ &\vdash (z_0, a, bba\gamma_0) & \text{oder} & \vdash (z_1, a, a\gamma_0) \\ &\vdash (z_0, e, abba\gamma_0) & & \vdash (z_1, e, \gamma_0) \\ & & & \vdash (z_2, e, e) \end{aligned}$$

### Satz 11.5

Sei  $K = (Z, X, \Gamma, \delta, z_0, \gamma_0, F)$  ein pda.

Falls  $(z, w, \gamma) \vdash^n (z', e, e)$  mit  $\gamma \in \Gamma$ , dann gilt auch:  $(z, w\bar{w}, \gamma\alpha) \vdash^n (z', \bar{w}, \alpha)$  für alle  $\alpha \in \Gamma^*$  und  $\bar{w} \in X^*$ .

**Definition 11.6** (Akzeptieren mit leerem Keller)

$K = (Z, X, \Gamma, \delta, z_0, \gamma_0, F)$  sei ein pda.

Ein Wort  $w \in X^*$  wird von  $K$  mit einer leeren Kellerliste akzeptiert, wenn gilt:  $(z_0, w, \gamma_0) \vdash^+ (z, e, e)$  mit  $z \in Z$ .

Mit  $L_e(K)$  wird die Menge aller Wörter bezeichnet, die von  $K$  mit leerer Kellerliste akzeptiert werden.

**Satz 11.7** ( $\mathcal{L}(K) \subseteq L_e(K)$ )

Sei  $L = \mathcal{L}(G)$  für einen pda  $K = (Z, X, \Gamma, \delta, z_0, \gamma_0, F)$ .

Dann kann man eine pda  $K'$  konstruieren, so daß  $L_e(K') = \mathcal{L}(K)$  gilt.

Bew.:  $K'$  "simuliere"  $K$ . Falls  $K$  einen Endzustand erreicht, soll  $K'$  fortfahren und einen Zustand  $z_e$  mit einer leeren Kellerliste erreichen.

Schwierigkeit:  $K$  kann eine leere Kellerliste erreichen, ohne in einem Endzustand zu kommen.  $K'$  darf dann das entsprechende Wort nicht akzeptieren.

Daher bekommt  $K'$  eine spezielle Grundmarkierung, die nur entfernt werden kann, wenn  $K'$  den Zustand  $z_e$  erreicht.

$K' = (Z \cup \{z_e, \bar{z}\}, X, \Gamma \cup \{\bar{\gamma}\}, \delta', \bar{z}, \bar{\gamma}, \emptyset)$

$\delta'$  wird folgendermaßen festgelegt:

1.  $\delta'(\bar{z}, e, \bar{\gamma}) = \{(z_0, \gamma_0\bar{\gamma})\}$
2.  $\forall z, z' \in Z, \forall a \in X \cup \{e\}, \forall \gamma \in \Gamma, \forall \alpha \in \Gamma^* : (z', \alpha) \in \delta(z, a, \gamma) \implies (z', \alpha) \in \delta'(z, a, \gamma)$
3.  $\forall z \in Z, \forall \gamma \in \Gamma \cup \{\bar{\gamma}\}$  gilt:  $(z_e, e) \in \delta'(z, e, \gamma)$
4.  $\forall \gamma \in \Gamma \cup \{\bar{\gamma}\}$  gilt:  $(z_e, e) \in \delta'(z_e, e, \gamma)$

□

**Satz 11.8** ( $L_e(K) \subseteq \mathcal{L}(K)$ )

$K = (Z, X, \Gamma, \delta, z_0, \gamma_0, \emptyset)$  sei ein pda. Man kann einen pda  $K'$  konstruieren, so daß  $\mathcal{L}(K') = L_e(K)$  gilt.

Bew.:  $K'$  "simuliert"  $K$  und hat eine spezielle Marke  $\bar{\gamma}$  als Grundmarkierung.

Wenn  $K'$  das Zeichen  $\bar{\gamma}$  kürzen kann, geht er in einen neuen Endzustand  $z_e$  über.

□

**Satz 11.9** ( $L$  kontextfrei  $\implies L = L(\text{pda})$ )

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik. Man kann einen pda  $K$  konstruieren, so daß  $L_e(K) = L(G)$  gilt.

Bew.:  $K$  "simuliert" alle Linksableitungen in  $G$ .  $K = (\{z_0\}, X, V \cup X, \delta, z_0, S, \emptyset)$ .

$\delta$  wird folgendermaßen festgelegt:

1. Ist  $A \rightarrow \alpha \in P$ , dann gilt:  $(z_0, \alpha) \in \delta(z_0, e, A)$
2. Für alle  $a \in X$  gilt  $\delta(z_0, a, a) = \{(z_0, e)\}$

Beh.: Aus  $A \xrightarrow{m} w, w \in X^*$  folgt  $(z_0, w, A) \vdash^* (z_0, e, e)$

$\underline{m=1}$ : Es ist  $A \rightarrow w \in P$ , also gilt  $(z_0, w) \in \delta(z_0, e, A)$ , also  $(z_0, w, A) \vdash^{|w|} (z_0, w, w) \vdash^{|w|} (z_0, e, e)$

$\underline{m \rightarrow m+1}$ : Die Beh. gelte für alle  $m' \leq m$  und es gelte  $A \xrightarrow{m+1} w, w \in X^*$ , d.h.  $A \xrightarrow{m+1} w_1 \xrightarrow{m} w$  mit  $w = u_1 A_1 u_2 A_2 \dots u_l A_l u_{l+1}$  mit  $A_i \in V_i, i \in [1:l]$  und  $u_j \in X^*, j \in [1:l+1]$ .

Es ist  $w = u_1 v_1 u_2 v_2 \dots u_l v_l u_{l+1}$  mit  $v_i \in X^*$  und  $A_i \xrightarrow{m_i} v_i, m_i \leq m$  für  $i \in [1:l]$

Aus der IV und 11.5 folgt:

$$(z_0, w, A) \vdash^{|u_1|} (z_0, w, u_1 A_1 u_2 A_2 \dots u_l A_l u_{l+1}) \vdash^{|u_1|} (z_0, v_1 u_2 \dots u_l v_l u_{l+1}, A_1 u_2 \dots u_l A_l u_{l+1}) \\ \vdash^{|u_1|} (z_0, u_2 \dots u_l v_l u_{l+1}, u_2 \dots u_l A_l u_{l+1}) \vdash^{|u_2|} \dots \vdash^{|u_l|} (z_0, v_l u_{l+1}, A_l u_{l+1}) \vdash^{|u_{l+1}|} (z_0, u_{l+1}, u_{l+1}) \vdash^{|u_{l+1}|} (z_0, e, e)$$

Bew.: Aus  $(z_0, w, A) \vdash^{|u_1|} (z_0, w, u_1 A_1 u_2 A_2 \dots u_l A_l u_{l+1})$  folgt  $A \xrightarrow{*} w$  (klar)

Hieraus folgt dann:  $L_e(K) = L(G)$ . □

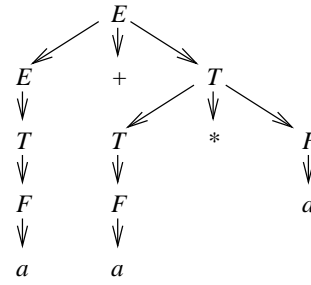
### Beispiel 11.10

$G = (V, X, S, P)$  mit  $V = \{E, T, F\}, X = \{a, (, ), +, *\}, S = E, P = \{E \rightarrow E + T | T, T \rightarrow T * F | F, F \rightarrow (E) | a\}$

1.  $\delta(z_0, e, E) = \{(z_0, E + T), (z_0, T)\}$
2.  $\delta(z_0, e, T) = \{(z_0, T * F), (z_0, T)\}$
3.  $\delta(z_0, e, F) = \{(z_0, (E)), (z_0, a)\}$
4.  $\delta(z_0, x, x) = \{(z_0, e)\} \quad \forall x \in X$

Bei Eingabe von  $a + a * a$  können folgende Überführungen vorgenommen werden:

$(z_0, a + a * a, E) \vdash (z_0, a + a * a, E + T) \vdash (z_0, a + a * a, T + T) \xrightarrow{2} (z_0, a + a * a, F + T) \vdash (z_0, a + a * a, a + T) \vdash (z_0, a * a, T) \vdash (z_0, a * a, T * F) \vdash (z_0, a * a, F * F) \vdash (z_0, a * a, a * F) \xrightarrow{2} (z_0, a, F) \vdash (z_0, a, a) \vdash (z_0, e, e)$   
 ( $K$  ist TOP-DOWN-Analysator)



### Bemerkung 11.11

Für eine kontextfreie Grammatik  $G = (V, X, S, P)$  in GNF kann eine einfacher topdown-Analysator konstruiert werden:  $K = (\{z_0\}, X, V, \delta, z_0, S, \emptyset)$  mit  $(z_0, \alpha) \in \delta(z_0, a, A)$  falls  $A \rightarrow a\alpha \in P$ .

Zum Akzeptieren eines Wortes  $w$  mit  $|w| = n$  benötigt  $K$   $n$  Schritte:  $K$  akzeptiert in **Realzeit**.

### Satz 11.12 ( $L = L(pda) \implies L$ kontextfrei)

$K = (Z, X, \Gamma, \delta, z_0, \gamma_0, F)$  sei ein pda. Man kann eine kontextfreie Grammatik  $G = (V, X, S, P)$  konstruieren, so daß  $L(G) = L_e(K)$  gilt.

Bew.: Setze  $V = \{[z\gamma z'] | z, z' \in Z, \gamma \in \Gamma\} \cup \{S\}$ .  $P$  werde folgendermaßen festgelegt:

1.  $\delta(z, a, \gamma)$  enthalte  $(z', \gamma_1, \dots, \gamma_k), k \geq 1$ . Dann seine alle  $[z\gamma z'] \rightarrow a[z'\gamma_1 z_1][z_1 \gamma_2 z_2] \dots [z_{k-1} \gamma_k z_k]$  für alle Folgen  $z_1, \dots, z_k \in Z$  aus  $P$ .
2.  $\delta(z, a, \gamma)$  enthalte  $(z', e)$ . Dann sei  $[z\gamma z'] \rightarrow a$  aus  $P$ .
3.  $S \rightarrow [z_0 \gamma_0 z]$  ist für alle  $z \in Z$  aus  $P$ .

Beh.: Aus  $[z\gamma z] \xrightarrow{m} w, w \in X^*$  folgt  $(z, w, \gamma) \vdash^* (\tilde{z}, e, e)$ .

Bew.: Induktion über  $m$ :

$\underline{m=1}$ :  $[z\gamma z] \xrightarrow{1} w, w \in X^*$

Es liegt 2. vor, also  $w = a \in X \cup \{e\}$ , und  $\delta(z, a, \gamma)$  enthält  $(\tilde{z}, e)$ , also  $(z, w, \gamma) \vdash (\tilde{z}, e, e)$

$\underline{m \rightarrow m+1}$ : Es gelte  $[z\gamma z] \xrightarrow{m+1} w$ , also  $[z\gamma z] \xrightarrow{m+1} a[z'\gamma_1 z_1][z_1 \gamma_2 z_2] \dots [z_{k-1} \gamma_k z_k], k \geq 1$  und  $\delta(z, a, \gamma)$  enthält  $(z', \gamma_1, \dots, \gamma_k)$ .

Hierbei sind  $z_1, \dots, z_{k-1} \in Z$ . Setze  $z_k := \tilde{z}$ .

Es gilt weiter  $[z'\gamma_1 z_1] \xrightarrow{m_1} w_1, m_1 \leq m, [z_{i-1} \gamma_i z_i] \xrightarrow{m_i} w_i, m_i \leq m, i \in [2:k]$  und  $w = aw_1 \dots w_k$ .

Aufgrund der IV gilt  $(z', w_1, \gamma_1) \xrightarrow{*} (z_1, e, e)$

$$(z_{i-1}, w_i, \gamma_i) \vdash^* (z_i, e, e), i \in [2:k],$$

also zusammen:  $(z, aw_1 \dots w_k, \gamma) \vdash (z', w_1 \dots w_k, \gamma_1 \dots \gamma_k) \vdash (z_1, w_2 \dots w_k, \gamma_2 \dots \gamma_k) \vdash \dots \vdash (z_{k-1}, w_k, \gamma_k) \vdash (\tilde{z}, e, e)$

Entsprechend zeigt man: Aus  $(z, w, \gamma) \vdash (\tilde{z}, e, e)$  folgt  $[z\gamma z] \xrightarrow{*} w$ .

Es folgt:  $S \xrightarrow{*} [z_0 \gamma_0 z] \vdash^* w, w \in X^*$  g.d.w.  $(z_0, w, \gamma_0) \vdash (z, e, e)$ , also  $L(G) = L_e(K)$ . □

### Satz 11.13

$X$  sei ein Alphabet und  $L \subseteq X^*$ . Folgende Aussagen sind gleichwertig:

1.  $L = L(G)$  für eine kontextfreie Grammatik  $G$ .
2.  $L = L(K)$  für einen pda  $K$ .
3.  $L = L_e(K)$  für einen pda  $K$ .

# Kapitel 12

## Abgeschlossenheitseigenschaften kontextfreier Sprachen

### Satz 12.1

Der Durchschnitt zweier kontextfreier Sprachen ist nicht notwendig kontextfrei.

Bew.: Setze  $X = \{a, b, c\}$ . Die Sprachen  $L_1 = \{a^n b^n c^i | n \geq 1 \text{ und } i \geq 0\}$  und  $L_2 = \{a^j b^n c^n | n \geq 1 \text{ und } j \geq 0\}$  sind kontextfrei.

Kontextfreie Grammatiken, die  $L_1$  und  $L_2$  erzeugen sind:

$G_1 = (\{S, T\}, X, S, P_1)$  mit  $P_1 = \{S \rightarrow Sc|T, T \rightarrow aTb|ab\}$  und

$G_2 = (\{S, T\}, X, S, P_2)$  mit  $P_2 = \{S \rightarrow aS|T, T \rightarrow bTc|bc\}$

Die Sprache  $L_1 \cap L_2 = \{a^n b^n c^n | n \geq 1\}$  ist aber nicht kontextfrei. □

### Satz 12.2

$L \subseteq X^*$  sei eine kontextfreie Sprache und  $f : X \rightarrow \mathbb{P}(Y^*)$  sei eine Substitution. Weiter sei  $f(x)$  kontextfrei für alle  $x \in X$ . Dann ist auch  $f(L)$  kontextfrei.

Bew.: Sei  $X = \{x_1, \dots, x_n\}$  und setze  $L_i := f(x_i)$ ,  $i \in [1 : n]$ .

$G = (V, X, S, P)$  sei eine kontextfreie Grammatik mit  $L = L(G)$  und  $G_i = (V_i, Y, S_i, P_i)$  seien kontextfreie Grammatiken mit

$L(G_i) = L_i$ ,  $i \in [1 : n]$

O.B.d.A.: Die Mengen  $V, V_1, \dots, V_n$  seien paarweise disjunkt.

Konstruktion einer kontextfreien Grammatik  $G' = (V', Y, S, P')$  für  $f(L)$ :

1.  $V' = \left(\bigcup_{i=1}^n V_i\right) \cup V$

2.  $h : (V \cup X)^* \rightarrow (V \cup \{S_1, \dots, S_n\})^*$  sei der durch  $h(A) = A$ ,  $\forall A \in V$  und  $h(x_i) = S_i$ ,  $i \in [1 : n]$  festgelegte Homomorphismus.

3. Setze  $P' = \{A \rightarrow h(\alpha) | A \rightarrow \alpha \in P\} \cup \left(\bigcup_{i=1}^n P_i\right)$

also:  $P'$  besteht aus den Produktionen der  $G_i$ ,  $i \in [1 : n]$ , zusammen mit den Produktionen von  $G$ , wobei man in den rechten Seiten dieser Produktionen die  $x_i$  durch die Variablen  $S_i$  ersetzt hat.

Sei  $w = x_{i_1} x_{i_2} \dots x_{i_l} \in L$  und sei weiter  $w_{i_1} \in L_{i_1}, w_{i_2} \in L_{i_2}, \dots, w_{i_l} \in L_{i_l}$ .

Dann gilt:  $S \xrightarrow[G']{*} S_{i_1} S_{i_2} \dots S_{i_l} \xrightarrow[G']{*} w_{i_1} w_{i_2} \dots w_{i_l} \in f(L)$  und daher gilt:  $f(L) \subseteq L(G')$ .

Umgekehrt:  $L(G') \subseteq f(L)$ .

Ist  $A \xrightarrow[G']{*} w$ ,  $A \in V$ ,  $w \in Y^*$  eine Ableitung bzgl.  $G'$ , dann gibt es eine Ableitung  $A \xrightarrow[G]{*} w'$ ,  $w' \in X^*$  bzgl.  $G$  mit  $w \in f(w')$ .

Es gelte  $A \xrightarrow[G']{K} w$ ,  $A \in V$ ,  $w \in Y^*$ .

$k = 1$ :

Dann ist  $A \xrightarrow[G']{1} w \hat{=} A \xrightarrow[G']{1} e$ ,  $a \rightarrow e \in P$  und  $w = e = f(e)$ .

$k \rightarrow k + 1$ :

Es gelte:  $A \xrightarrow[G']{1} w_0 = h(\alpha) \xrightarrow[G']{k} w$  mit  $A \rightarrow \alpha \in P$  und  $w_0 = B_1 \dots B_{l_0} A_1 B_{l_0+1} \dots B_{l_0+l_1} A_2 \dots A_t B_{l_0+\dots+l_{t-1}+1} \dots B_{l_0+\dots+l_t}$   
mit  $A_j \in V$  für  $j \in [1 : t]$ ,  $t \geq 0$  und  $B_i \in \{S_1, \dots, S_n\}$  für  $i \in [1 : l_0 + \dots + l_t]$  und  $l_k \geq 0$  für  $k \in [0 : t]$ .

Es ist dann  $w = u_1 \dots u_{l_0} w_1 u_{l_0+1} \dots u_{l_0+l_1} w_2 \dots w_t u_{l_0+\dots+l_{t-1}+1} \dots u_{l_0+\dots+l_t}$ , wobei gilt:

$B_i \xrightarrow[G_j]{*} u_i$  mit  $B_i = S_j$  für  $i \in [1 : l_0 + \dots + l_t]$  und  $A_j \xrightarrow[G]{k_j} w_j$  mit  $k_j \leq k$  für  $j \in [1 : k]$

Laut IV gilt für  $j \in [1 : t]$ :  $\exists A_j \xrightarrow[G]{*} w'_j, w'_j \in X^*$  mit  $w_j \in f(w'_j)$

Wähle  $b_i \in X$  so, daß  $h(b_i) = B_i$  gilt für  $i \in [1 : l_0 + \dots + l_t]$ . Dann ist  $A \rightarrow \alpha \in P$  mit

$\alpha = b_1 \dots b_{l_0} A_1 b_{l_0+1} \dots b_{l_0+l_1} A_2 \dots A_t b_{l_0+\dots+l_{t-1}+1} \dots b_{l_0+\dots+l_t}$

Es ist  $u_i \in f(b_i)$  für  $i \in [1 : l_0 + \dots + l_t]$ .

Weiter gilt:  $A \xrightarrow[G]{1} \alpha \xrightarrow[G]{*} b_1 \dots b_{l_0} w_1 b_{l_0+1} \dots b_{l_0+l_1} w_2 \dots w_t b_{l_0+\dots+l_{t-1}+1} \dots b_{l_0+\dots+l_t} = w'$  mit  $w \in f(w')$ . □

### Beispiel 12.3

$X = \{0, 1\}$ ,  $Y = \{a, b, c\}$ ,  $f : X \rightarrow \mathbb{P}(Y)$ ,  $f(0) = \{a\}$ ,  $f(1) = \{ww^R \mid w \in \{b, c\}^+\}$ .  
 $L = \{0^n 1^n \mid n \geq 1\}$  ist kontextfrei, somit auch  $f(L) = \{a^n w_1 w_1^R \dots w_n w_n^R \mid w_i \in \{b, c\}^+, i \in [1 : n], n \geq 1\}$ .

### Korollar 12.4

- (a)  $X$  sei ein Alphabet und  $L_1, L_2$  seien kontextfreie Sprachen über  $X$ . Dann gilt:  
 (1)  $L_1 \cup L_2$  ist kontextfrei.  
 (2)  $L_1 \cdot L_2$  ist kontextfrei.  
 (3)  $L_1^*$  ist kontextfrei.  
 (4)  $L_1^+$  ist kontextfrei.  
 (b)  $X$  und  $Y$  seien Alphabete und  $h : X^* \rightarrow Y^*$  sei ein Homomorphismus.  
 Ist  $L \subseteq X^*$  kontextfrei, dann ist auch  $h(L) \subseteq Y^*$  kontextfrei.

Bew.: (a) Setze  $X' = \{1, 2\}$  und  $f : X' \rightarrow \mathbb{P}(X^*)$  mit  $f(1) = L_1$ ,  $f(2) = L_2$ .  
 Dann gilt: (1)  $L_1 \cup L_2 = f(\{1, 2\})$   $\{1, 2\}$  ist kontextfrei  
 (2)  $L_1 \cdot L_2 = f(\{12\})$   $\{12\}$  ist kontextfrei.  
 (3)  $L_1^* = f(\{1\}^*)$   $\{1\}^*$  ist kontextfrei.  
 (4)  $L_1^+ = f(\{1\}^+)$   $\{1\}^+$  ist kontextfrei.  
 (b) Setze  $f : X \rightarrow \mathbb{P}(Y^*)$  mit  $f(x) = \{h(x)\}$ ,  $x \in X$ .  
 Dann ist  $h(L) = f(L)$  kontextfrei.

### Satz 12.5

Das Komplement einer kontextfreien Sprache  $L \subseteq X^*$  ist nicht notwendig kontextfrei.

Bew.: Falls das Komplement jeder kontextfreien Sprache wieder kontextfrei wäre, dann wäre auch für die kontextfreien Sprachen  
 $L_1, L_2$ :  $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$  kontextfrei. (Widerspruch, vergleiche 12.1) □

### Satz 12.6 ( $L \cap R$ ist kontextfrei)

$L \subseteq X^*$  sei kontextfrei,  $R \subseteq X^*$  sei regulär. Dann ist auch  $L \cap R$  kontextfrei.

Bew.:  $L$  werde von dem pda  $K_1 = (Z_K, X, \Gamma, \delta_K, z_K, \gamma_0, F_K)$  erkannt.  
 $R$  werde von dem DFA  $E = (Z_E, X, \delta_E, z_E, F_E)$  erkannt.  
 Aus  $K_1$  und  $E$  wird ein Kellerautomat  $K_2$  konstruiert, der  $L \cap R$  erkennt:  
 $K_2 = (Z_K \times Z_E, X, \Gamma, \delta, (z_K, z_E), \gamma_0, F_K \times F_E)$ , wobei  $\delta$  folgendermaßen definiert ist:  
 $\forall (p, q) \in Z_K \times Z_E, \forall x \in X \cup \{e\}, \forall \gamma \in \Gamma$ :  $(p', \overline{\delta_e}(q, x), \alpha) \in \delta((p, q), x, \gamma)$  g.d.w.  $(p', \alpha) \in \delta_K(p, x, \gamma)$   
 Sei  $w \in L \cap R$ .  $w$  kann in der Form  $w = a_1 a_2 \dots a_n$  mit  $a_i \in X \cup \{e\}$  für  $i \in [1 : n]$  so geschrieben werden, daß gilt:  
 Es gibt eine Folge von Zuständen  $p_0 = z_K, p_1, \dots, p_n$  aus  $Z_K$ , eine Folge von Zuständen  $q_0 = z_E, q_1, \dots, q_n$  aus  $Z_E$  und eine Folge von Wörtern  $\alpha_0 = \gamma_0, \alpha_1, \dots, \alpha_n$  aus  $\Gamma^*$ , mit  $\overline{\delta_E}(q_i, a_{i+1}) = q_{i+1}$   
 $(p_i, a_i a_{i+1} \dots a_n, \alpha_i) \stackrel{1}{K_1} (p_{i+1}, a_{i+1} \dots a_n, \alpha_{i+1})$  für  $i \in [0 : n - 1]$  mit  $q_n \in F_E$  und  $p_n \in F_K$ .  
 Also gilt in  $K_2$ :  $((p_i, q_i), a_{i+1} \dots a_n, \alpha_i) \stackrel{1}{K_2} ((p_{i+1}, q_{i+1}), a_{i+2} \dots a_n, \alpha_{i+1})$  für  $i \in [0 : n - 1]$ , also:  $L \cap R \subseteq L(K_2)$   
 Umkehrung entsprechend. □

### Beispiel 12.7

$L = \{a^m b^n a^m b^n \mid n \geq 1\}$  ist nicht kontextfrei.

Bew.: Ann.:  $L$  ist kontextfrei.

- Wähle  $k$  wie im Iterationslemma und setze  $m = n = k$ .  
 Es gilt:  $a^m b^n a^m b^n = a^k b^k a^k b^k = uvwxy$  mit  $|vwx| < k$ ,  $vx \neq e$ .  
 Es muß dann  $uvw \in L$  sein, also  $uvw = a^{m_1} n^{n_1} a^{m_2} b^{n_2}$  mit  $m_1 = m_2, n_1 = n_2$ .  
 I.  $v \neq e$

- |   |                                       |
|---|---------------------------------------|
| 1. $v$ ist Teilwort der ersten $a$ 's.                    | Dann ist $m_1 < m_2$ . (Widerspruch!) |
| 2. $v$ ist Teilwort der ersten $a$ 's und $b$ 's.         | Dann ist $m_1 < m_2$ . (Widerspruch!) |
| 3. $v$ ist Teilwort der ersten $b$ 's.                    | Dann ist $n_1 < n_2$ . (Widerspruch!) |
| 4. $v$ ist Teilwort der ersten $b$ 's und zweiten $a$ 's. | Dann ist $m_1 > m_2$ . (Widerspruch!) |
| 5. $v$ ist Teilwort der zweiten $a$ 's.                   | Dann ist $m_1 > m_2$ . (Widerspruch!) |
| 6. $v$ ist Teilwort der zweiten $a$ 's und $b$ 's.        | Dann ist $m_1 > m_2$ . (Widerspruch!) |
| 7. $v$ ist Teilwort der zweiten $b$ 's.                   | Dann ist $n_1 > n_2$ . (Widerspruch!) |

II.  $x \neq e$  entsprechend.

### Beispiel 12.8

$L = \{ww \mid w \in \{a, b\}^+\}$  ist nicht kontextfrei.

Bew.: Ann.:  $L$  ist kontextfrei.

Setze  $L' = L \cap \underbrace{a^+ b^+ a^+ b^+}_{\text{regulär}} = \{a^m b^n a^m b^n \mid m, n \geq 1\}$   
 $\implies L'$  ist kontextfrei (Widerspruch!)

# Kapitel 13

## Indizierte Grammatiken und Ableitungsmodi

Ziel: syntaktische Beschreibung nicht-kontextfreier Sprachen.

Interpretation der Anwendung einer kontextfreien Produktion, z.B.:  $A \rightarrow BaC$ :

Aufruf einer Prozedur  $A$ , deren Rumpf aus dem Aufruf einer Prozedur  $B$ , dem Schreiben von 'a' und dem Aufruf einer Prozedur  $C$  besteht.

(I)  $\frac{\textit{procedure } A()}{\textit{begin } B();}$   
 $\frac{\textit{print } 'a';}{C();}$   
 $\textit{end};$

(II)  $\frac{\textit{procedure } A(x)}{\textit{begin } B(x);}$   
 $\frac{\textit{print } 'a';}{C(x);}$   
 $\textit{end};$

Diese Prozeduren haben jedoch keine Parameter.

Erweiterung: Jede Prozedur wird mit einem Parameter versehen.

Aktualparameter sollen Wörter über einem Alphabet  $I$  sein. Diese können bis jetzt jedoch noch nicht modifiziert werden. Es soll eine kellerartige Manipulation zugelassen werden, d.h. am linken Ende eines Wortes soll ein Buchstabe gestrichen oder angefügt werden können.

(IIa)  $\frac{\textit{procedure } D(x)}{\textit{begin } E(fx);}$   
 $\textit{end};$

(IIb)  $\frac{\textit{procedure } D(x)}{\textit{begin if } x = fy \textit{ then } E(y);}$   
 $\frac{\textit{else error};}{\textit{end};}$

$D \rightarrow Ef$

$Df \rightarrow E$

### Definition 13.1 (Indizierte Grammatik)

Eine indizierte Grammatik ist ein 5-tupel  $G = (N, T, I, P, S)$  mit

- (1)  $N, T, I$  sind paarweise disjunkte, endliche Mengen:  
Die Mengen der Nichtterminalen (Variablen), Terminalen (Konstanten) und die Menge der Indizes.
- (2)  $P$  ist eine endliche Menge von Produktionen, wobei jede Produktion von einer der Formen:
  - (a)  $A \rightarrow \alpha$
  - (b)  $A \rightarrow Bf$
  - (c)  $Af \rightarrow B$  mit  $A, B \in N, \alpha \in (N \cup T)^*, f \in I$  ist.
- (3)  $S \in N$  (Startvariable)

## 13.1 Ableitung bzgl. indizierter Grammatiken

Es werden zwei Arten der Parameterübergabe für die Prozedur (II) betrachtet:

„call by Value“

„call by Reference“

Diese legen einen V-Ableitungsmodus ( $\overset{V}{\Longrightarrow}$ ) und einen R-Ableitungsmodus ( $\overset{R}{\Longrightarrow}$ ) fest.

Der Aufruf von (II) im Modus „call by value“ mit Aktualparameter  $\gamma$  bewirkt:

Die Prozeduren  $B, C$  werden ebenfalls mit dem Aktualparameter  $\gamma$  aufgerufen:  $A\gamma \overset{V}{\Longrightarrow} B\gamma a C\gamma$ .

Der Aufruf von (II) im Modus „call by reference“ mit dem Parameter  $\gamma$  bewirkt:

Die Prozedur  $B$  wird ebenfalls mit dem  $\gamma$  aufgerufen, die Prozedur  $C$  jedoch mit einem Parameter  $\gamma'$ , wobei  $\gamma'$  von der Auswertung von  $B$  mit  $\gamma$  abhängt:  $A\gamma \overset{R}{\Longrightarrow} B\gamma a C(\gamma')$ .

## 13.2 Festlegung der Ableitungsmodi

Zunächst werden zwei Operationen  $:_V$  und  $:_R$  definiert:

Sei  $\alpha = u_1 B_1 \beta_1 u_2 B_2 \beta_2 \dots B_k \beta_k u_{k+1}$  mit  $B_i \in N$ ,  $\beta_j \in I^*$  für  $j \in [1 : k]$  und  $u_i \in T^*$  für  $i \in [1 : k + 1]$ ,  $k > 0$ .

Es ist  $\alpha \in (NI^* \cup T)^*$

Sei weiter  $\gamma \in I^*$ . Setze  $A :_V \gamma = u_1 B_1 \beta_1 \gamma u_2 B_2 \beta_2 \gamma \dots B_k \beta_k \gamma u_{k+1}$   
und  $A :_R \gamma = u_1 B_1 \beta_1 \gamma u_2 B_2 \beta_2 \dots B_k \beta_k u_{k+1}$ .

### Definition 13.2 (Ableitungsmodi)

Sei  $w \in T^*$ ,  $A \in N$ ,  $\gamma \in I^*$ ,  $\Theta \in (NI^* \cup T)^*$  und  $f \in I$ .

#### V–Ableitungsmodus

Setze  $u_1 \xrightarrow{V} u_2$  falls

- (a)  $u_1 = wA\gamma\Theta$ ,  $u_2 = w(\alpha :_V \gamma)\Theta$ ,  $A \rightarrow \alpha \in P$  oder
- (b)  $u_1 = wA\gamma\Theta$ ,  $u_2 = wBf\gamma\Theta$ ,  $A \rightarrow Bf \in P$  oder
- (c)  $u_1 = wAf\gamma\Theta$ ,  $u_2 = wB\gamma\Theta$ ,  $Af \rightarrow B \in P$ .

#### R–Ableitungsmodus

- (a)  $u_1 = wA\gamma\Theta$ ,  $u_2 = w(\alpha\Theta :_R \gamma)$ ,  $A \rightarrow \alpha \in P$  oder
- (b) und (c) wie oben.

$\left. \begin{array}{l} \xrightarrow{V+} \\ \xrightarrow{R+} \end{array} \right\}$  transitive Hülle von  $\left\{ \begin{array}{l} \xrightarrow{V} \\ \xrightarrow{R} \end{array} \right\}$ ,  $\left. \begin{array}{l} \xrightarrow{V*} \\ \xrightarrow{R*} \end{array} \right\}$  reflexive, transitive Hülle von  $\left\{ \begin{array}{l} \xrightarrow{V} \\ \xrightarrow{R} \end{array} \right\}$ .

### Definition 13.3 (Die im V–Modus erzeugte Sprache)

$L_V(G) = \{w | w \in T^* \text{ und } S \xrightarrow{V*} w\}$

### Definition 13.4 (Die im R–Modus erzeugte Sprache)

$L_R(G) = \{w | w \in T^* \text{ und } S \xrightarrow{R*} w\}$

Eine Sprache  $L$  heißt **V–indiziert**, wenn  $L = L_V(G)$  für eine indizierte Grammatik  $G$  gilt.

Eine Sprache  $L$  heißt **R–indiziert**, wenn  $L = L_R(G)$  für eine indizierte Grammatik  $G$  gilt.

### Bemerkung 13.5

V– und R–Ableitungen sind Linksableitungen.

$I = \emptyset$ : V–Modus = R–Modus = Linksableitungen kontextfreier Grammatiken.

### Definition 13.6 (Erweiterte, indizierte Grammatik)

Eine **erweiterte, indizierte Grammatik** ist ein 5–tupel  $G = (N, T, I, P, S)$ . Hierbei sind  $N, T, I$  und  $S$  wie in 13.1. Die Produktionen in  $P$  sind von der Form  $Af \rightarrow \beta$ ,  $A \in N$ ,  $f \in I \cup \{e\}$  und  $\beta \in (NI^* \cup T)^*$ .

V– und R–Ableitungsmodus und  $L_V(G)$  und  $L_R(G)$  s.o.

### Satz 13.7

$G$  sei eine erweiterte, indizierte Grammatik. Man kann eine indizierte Grammatik  $G'$  konstruieren, so daß  $L_V(G') = L_V(G)$  und  $L_R(G') = L_R(G)$  gilt.

Bew.:  $\Pi : Af \rightarrow u_1 B_1 \gamma_1 u_2 B_2 \gamma_2 \dots u_n B_n \gamma_n u_{n+1}$  sei eine Produktion aus  $G$ .

(1) Ersetze  $\Pi$  durch

$$\Pi' : Af \rightarrow u_1 \hat{B}_1^{(0)} u_2 \hat{B}_2^{(0)} \dots u_n \hat{B}_n^{(0)} u_{n+1} \text{ und } \Pi'_i : \hat{B}_i^{(0)} \rightarrow B_i \gamma_i \quad \text{für } i \in [1 : n]$$

(2) Falls  $f \neq e$  ersetze  $\Pi'$  durch  $Af \rightarrow \hat{A}$  und  $\hat{A} \rightarrow u_1 \hat{B}_1^{(0)} u_2 \hat{B}_2^{(0)} \dots u_n \hat{B}_n^{(0)} u_{n+1}$

(3) Ersetze  $\Pi'_i, i \in [1 : n]$  durch

$$\begin{aligned} \text{Falls } \gamma_i = f_{n_i} \dots f_1 \text{ ist, dann setze } \hat{B}_i^{(j)} &\rightarrow \hat{B}_i^{(j+1)} f_{j+1}, \quad j \in [0 : n_i - 1] \\ \text{und } \hat{B}_i^{(n_i)} &\rightarrow B_i. \end{aligned}$$

□

### Beispiel 13.8

Sei  $G = (N, T, I, P, S)$  eine erweiterte, indizierte Grammatik mit  $N = \{A, B, C, D, E, S\}$ ,  $T = \{a, b, \$\}$ ,  $I = \{\hat{a}, \hat{b}, \#\}$

$P : S \rightarrow C\hat{a}\# \mid C\hat{b}\# \mid e$

$C \rightarrow C\hat{a}D \mid C\hat{b}D \mid D$

$D\hat{a} \rightarrow DaA$ ,  $D\hat{b} \rightarrow DbB$ ,  $D\# \rightarrow E\#\$$

$A \rightarrow E\hat{a}$ ,  $B \rightarrow E\hat{b}$ ,  $E \rightarrow e$

### Beispiel 13.9 (R–Modus Ableitung)

$x_i \in \{a, b\}, X_i \in \{A, B\}$

$$\begin{aligned} S &\xrightarrow{R} C\hat{x}_1\# \xrightarrow{R} C\hat{x}_2\hat{x}_1\#D \xrightarrow{R} \dots \xrightarrow{R} C\hat{x}_k\hat{x}_{k-1}\dots\hat{x}_1\#D^{k-1} \xrightarrow{R} D\hat{x}_k\hat{x}_{k-1}\dots\hat{x}_1\#D^{k-1} \xrightarrow{R} D\hat{x}_{k-1}\dots\hat{x}_1\#x_k X_k D^{k-1} \\ &\xrightarrow{R} D\hat{x}_{k-2}\dots\hat{x}_1\#x_{k-1} X_{k-1} x_k X_k D^{k-1} \xrightarrow{R} D\#x_1 X_1 x_2 X_2 \dots x_k X_k D^{k-1} \xrightarrow{R} E\#\$x_1 X_1 x_2 X_2 \dots x_k X_k D^{k-1} \\ &\xrightarrow{R} \$x_1 X_1 \#x_2 X_2 \dots x_k X_k D^{k-1} \xrightarrow{R} \$x_1 E\hat{x}_1 \#x_2 X_2 \dots x_k X_k D^{k-1} \xrightarrow{R} \$x_1 x_2 X_2 \hat{x}_1 \# \dots x_k X_k D^{k-1} \\ &\xrightarrow{R} \$x_1 x_2 E\hat{x}_2 \hat{x}_1 \# \dots x_k X_k D^{k-1} \xrightarrow{R} \$x_1 x_2 \dots x_k E\hat{x}_k \dots \hat{x}_1 \# D^{k-1} \xrightarrow{R} \$x_1 x_2 \dots x_k D\hat{x}_k \dots \hat{x}_1 \# D^{k-2} \\ &\xrightarrow{R} \$x_1 x_2 \dots x_k \$x_1 x_2 \dots x_k D\hat{x}_k \dots \hat{x}_1 \# D^{k-3} \xrightarrow{R} (\$x_1 x_2 \dots x_k)^{k-1} D\hat{x}_k \dots \hat{x}_1 \xrightarrow{R} (\$x_1 x_2 \dots x_k)^k E\hat{x}_k \dots \hat{x}_1 \# \\ &\xrightarrow{R} (\$x_1 x_2 \dots x_k)^k \end{aligned}$$

Somit ist  $L_R(G) = \{(\$w)^{|w|} | w \in \{a, b\}^*\}$ .

**Beispiel 13.10** (V-Modus Ableitung)

$$\begin{aligned}
 S &\xrightarrow{V} C\hat{x}_1\# \xrightarrow{V} C\hat{x}_2\hat{x}_1\#D\hat{x}_1\# \xrightarrow{V} C\hat{x}_3\hat{x}_2\hat{x}_1\#D\hat{x}_2\hat{x}_1\#D\hat{x}_1\# \xrightarrow{V^*} D\hat{x}_k\hat{x}_{k-1}\dots\hat{x}_1\#\overbrace{D\hat{x}_{k-1}\dots\hat{x}_1\#\dots D\hat{x}_1\#}^{\Theta} \\
 &\xrightarrow{V} D\hat{x}_{k-1}\dots\hat{x}_1\#x_kX_k\hat{x}_{k-1}\dots\hat{x}_1\#\Theta \xrightarrow{V^*} D\#x_1X_1\#x_2X_2\hat{x}_1\#\dots x_kX_k\hat{x}_{k-1}\dots\hat{x}_1\#\Theta \\
 &\xrightarrow{V^*} D\#x_1X_1\#x_2X_2\hat{x}_1\#\dots x_kX_k\hat{x}_{k-1}\dots\hat{x}_1\#\Theta \xrightarrow{V} E\#\$x_1X_1\#x_2X_2\hat{x}_1\#\dots x_kX_k\hat{x}_{k-1}\dots\hat{x}_1\#\Theta \\
 &\xrightarrow{V} \$x_1X_1\#x_2X_2\hat{x}_1\#\dots x_kX_k\hat{x}_{k-1}\dots\hat{x}_1\#\Theta \xrightarrow{V} \$x_1x_2X_2\hat{x}_1\#\dots x_kX_k\hat{x}_{k-1}\dots\hat{x}_1\#\Theta \xrightarrow{V^*} \$x_1x_2\dots x_k\Theta \\
 &\xrightarrow{V^*} \$x_1x_2\dots x_k\$x_1\dots x_{k-1}\$\dots \$x_1
 \end{aligned}$$

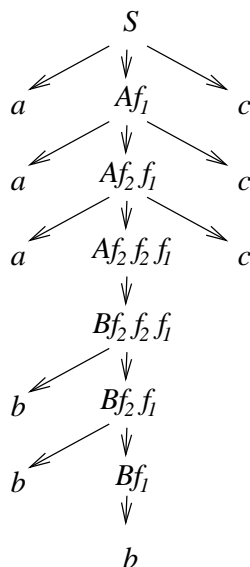
Somit ist  $L_V(G) = \{\$x_1\dots x_k\$x_1\dots x_{k-1}\$\dots \$x_1 \mid x_i \in \{a, b\}, i \in [1:k], k \geq 0\}$ .

# Kapitel 14

## Eigenschaften V-indizierter Sprachen

### Beispiel 14.1

$G = (N, T, I, P, S)$  mit  
 $N = \{S, A, B\}$ ,  
 $T = \{a, b, c\}$ ,  
 $I = \{f_1, f_2\}$   
 $P : S \rightarrow aAf_1c$   
 $A \rightarrow aAf_2c|B$   
 $Bf_1 \rightarrow b, Bf_2 \rightarrow bB$



V-Modus und R-Modus sind identisch, da auf der rechten Seite höchstens eine Variable auftritt.

Ableitung:  $S \Rightarrow aAf_1c \Rightarrow a^2Af_2f_1c^2 \Rightarrow a^3Af_2f_2f_1c^3 \Rightarrow a^3Bf_2f_2f_1c^3 \Rightarrow a^3bBf_2f_1c^3 \Rightarrow a^3b^2Bf_1c^3 \Rightarrow a^3b^3c^3$ .

Es ist  $L_V(G) = L_R(G) = \{a^n b^n c^n | n \geq 1\}$

### Satz 14.2

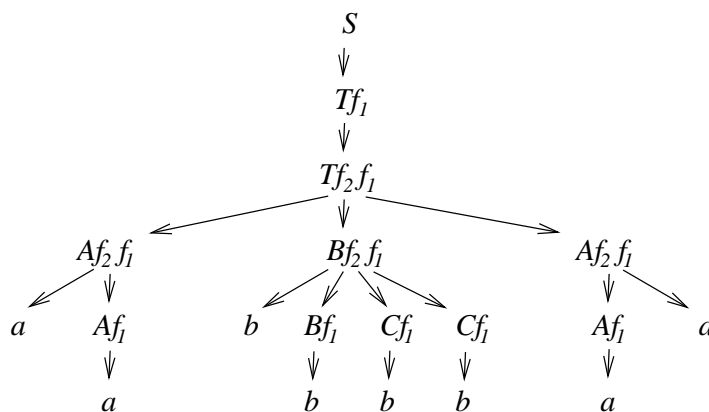
Jede kontextfreie Sprache ist eine V-indizierte Sprache.  
 Es gibt V-indizierte Sprachen, die nicht kontextfrei sind.

Bew.: Eine kontextfreie Grammatik kann als V-indizierte Grammatik mit  $I = \emptyset$  aufgefaßt werden.  
 $\{a^n b^n c^n | n \geq 1\}$  ist V-indiziert, aber nicht kontextfrei. □

### Beispiel 14.3

$G = (N, T, I, P, S)$  mit  $N = \{S, T, A, B, C\}$ ,  $T = \{a, b\}$ ,  $I = \{f_1, f_2\}$  und  
 $P : S \rightarrow Tf_1 \quad Af_1 \rightarrow a \quad Af_2 \rightarrow aA$   
 $T \rightarrow Tf_2|ABA \quad Bf_1 \rightarrow b \quad Bf_2 \rightarrow aBCC$   
 $Cf_1 \rightarrow b \quad Cf_2 \rightarrow bC$

Ableitungsbaum für  $a^2 b^4 a^2$ :



Es gilt für alle  $n \geq 1$ : (1)  $Af_2^{n-1} f_1 \xrightarrow{V^*} w, w \in X^* \iff w = a^n$   
 (2)  $Cf_2^{n-1} f_1 \xrightarrow{V^*} w, w \in X^* \iff w = b^n$   
 (3)  $Bf_2^{n-1} f_1 \xrightarrow{V^*} w, w \in X^* \iff w = b^{n^2}$

(1) und (2): klar.

(3)  $n \equiv 1$ : (klar)

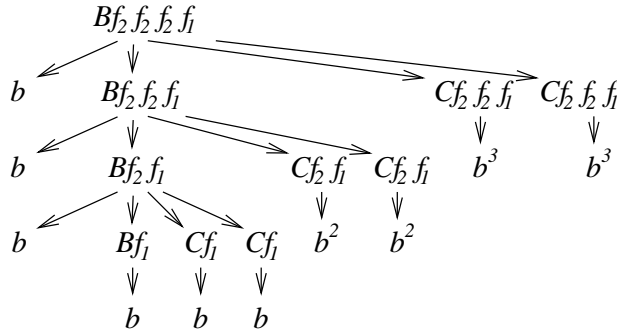
$$n \rightarrow n+1 \quad Bf_2^n f_1 \xrightarrow{V} bBf_2^{n-1} f_1 C f_2^{n-1} f_1 C f_2^{n-1} f_1 \xrightarrow{V^*} bb^{n^2} C f_2^{n-1} f_1 C f_2^{n-1} f_1 \xrightarrow{V^*} bb^{n^2} b^n b^n = b^{n^2+2n+1} = b^{(n+1)^2}$$

Es ist  $L_V(G) = \{a^n b^{n^2} c^n | n \geq 1\}$

Daß sich aus  $Bf_2^{n-1} f_1$  genau  $b^{n^2}$  herleiten läßt, beruht auf folgender Teilung von  $n^2$ :

$$n^2 = n + n^2 - n = n + 2 \binom{n(n-1)}{2} = n + 2 \sum_{i=1}^{n-1} i = n + 2 \cdot 1 + 2 \cdot 2 + \dots + 2 \cdot (n-1)$$

Veranschaulichung für  $n = 4$ :



### Definition 14.4 (V-i pda)

Ein V-indizierter Kellerautomat (V-i pda = V-indexed pda) ist ein 9-tupel  $A = (Z, X, \Gamma_1, \Gamma_2, \delta, z_0, A_0, g_0, F)$ . Hierbei ist:

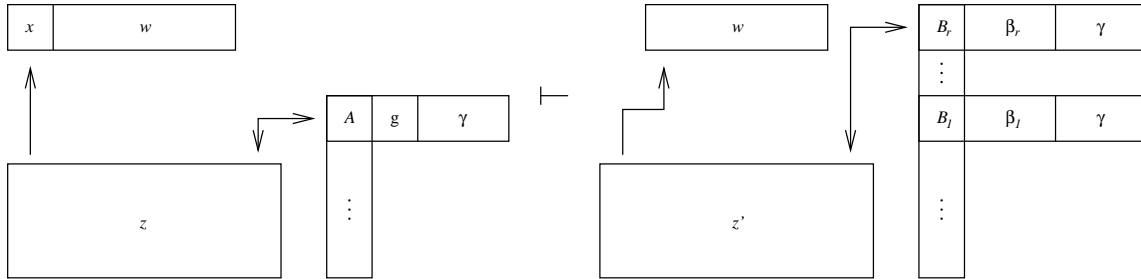
- |   |                          |
|---|--------------------------|
| (1) $Z$ eine endliche Menge   | (Zustandsmenge)          |
| (2) $X$ eine endliche Menge $\neq \emptyset$  | (Eingabealphabet)        |
| (3) $\Gamma_1$ eine endliche Menge  | (Kellersymbole)          |
| (4) $\Gamma_2$ eine endliche Menge  | (Indizes)                |
| (5) $z_0 \in Z$   | (Anfangszustand)         |
| (6) $A_0 \in \Gamma_1$  | (Anfangs-Kellerbelegung) |
| (7) $g_0 \in \Gamma_2 \cup \{e\}$   | (Anfangs-Index)          |
| (8) $F \subseteq Z$   | (Menge der Endzustände)  |
| (9) $\delta : Z \times (X \cup \{e\}) \times \Gamma_1 \times (\Gamma_2 \cup \{e\}) \rightarrow \mathbb{P}_f(Z \times (\Gamma_1 \times \Gamma_2^*))^*$ | (Überföhrungsfunktion)   |

Ein Tupel  $(z, w, \Theta)$ ,  $z \in Z$ ,  $w \in X^*$ ,  $\Theta \in (\Gamma_1 \times \Gamma_2^*)^*$  heißt **Konfiguration von  $K$** .

Relation  $\vdash$  auf den Konfigurationen:

$(z, xw, (A, g\gamma)\Theta) \vdash (z', w, (B_1, \beta_1\gamma) \dots (B_r, \beta_r\gamma)\Theta)$  g.d.w.  $(z', (B_1, \beta_1) \dots (B_r, \beta_r)) \in \delta(z, x, (A, g))$  für

$z, z' \in Z$ ,  $x \in X \cup \{e\}$ ,  $w \in X^*$ ,  $A, B_1, \dots, B_r \in \Gamma_1$ ,  $g \in \Gamma_2 \cup \{e\}$ ,  $\gamma, \beta_1, \dots, \beta_r \in \Gamma_2^*$ ,  $r \geq 0$  und  $\Theta \in (\Gamma_1 \times \Gamma_2^*)^*$



$\vdash^n$ ,  $\vdash^+$ ,  $\vdash^*$ :  $n$ -faches Produkt, transitive und reflexive, transitive Hülle.

$$L(K) = \{w | w \in X^* \text{ und } (z_0, w, (A_0, g_0)) \vdash^* (z, e, \Theta), z \in F\}$$

$$L_e(K) = \{w | w \in X^* \text{ und } (z_0, w, (A_0, g_0)) \vdash^* (z, e, e)\}$$

### Beispiel 14.5

$K = (Z, X, \Gamma_1, \Gamma_2, \delta, z_0, A_0, g_0, G)$  mit  $Z = \{z_0, z_1, z_2, z_3\}$ ,  $X = \{a, b, c\}$ ,  $\Gamma_1 = \{A, A_0\}$ ,  $\Gamma_2 = \{g, g_0\}$ ,  $F = \{z_0\}$ .

$\delta$  ist folgendermaßen definiert:

- (1)  $\delta(z_0, a, (A_0, e)) = \{(z_1, (A_0, g))\}$
- (2)  $\delta(z_1, a, (A_0, e)) = \{(z_1, (A_0, g))\}$
- (3)  $\delta(z_1, b, (A_0, g)) = \{(z_2, (A, e)(A_0, g))\}$
- (4)  $\delta(z_2, b, (A, g)) = \{(z_2, (A, e))\}$
- (5)  $\delta(z_2, e, (A, g_0)) = \{(z_3, e)\}$
- (6)  $\delta(z_3, c, (A_0, g)) = \{(z_3, (A_0, e))\}$
- (7)  $\delta(z_3, e, (A_0, g_0)) = \{(z_0, e)\}$

|  |   |   |
|--|---|---|
| Gebe $a^3 b^3 c^3$ ein: $(z_0, a^3 b^3 c^3, (A_0, g_0))$ | $\vdash$ $(z_1, a^2 b^3 c^3, (A_0, g_0))$ | $\stackrel{2}{\vdash}$ $(z_1, b^3 c^3, (A_0, g_0))$ |
|  | $\vdash$ $(z_2, b^2 c^3, (A_0, g_0))$     | $\stackrel{2}{\vdash}$ $(z_2, c^3, (A_0, g_0))$     |
|  | $\vdash$ $(z_3, c^3, (A_0, g_0))$         | $\stackrel{3}{\vdash}$ $(z_3, e, (A_0, g_0))$       |
|  | $\vdash$ $(z, e, e)$                      |   |

### Satz 14.6

- (a) Sei  $L = L(K)$  für einen V-i pda  $K$ . Dann gibt es einen V-i pda  $K'$  mit  $L = L_e(K')$ .  
 (b) Sei  $L = L_e(K)$  für einen V-i pda  $K$ . Dann gibt es einen V-i pda  $K'$  mit  $L = L(K')$ .

### Satz 14.7

Eine Sprache ist genau dann V-indiziert, wenn sie von einem V-i pda akzeptiert wird.

### Beispiel 14.8

$K_1 = (Z, X, \Gamma_1, \Gamma_2, \delta, z_0, A, g, F)$  mit  $Z = \{z_0, z_1, z_2\}$ ,  $X = \{1, c\}$ ,  $\Gamma_1 = \{A\}$ ,  $\Gamma_2 = \{f, g\}$ ,  $F = \{z_2\}$ .

$\delta$  ist folgendermaßen festgelegt:

- (1)  $\delta(z_0, 1, (A, g)) = \{(z_0, (A, fg))\}$
- (2)  $\delta(z_0, 1, (A, f)) = \{(z_0, (A, ff))\}$
- (3)  $\delta(z_0, c, (A, f)) = \{(z_1, (A, e)(A, f))\}$
- (4)  $\delta(z_1, 1, (A, f)) = \{(z_1, (A, e))\}$
- (5)  $\delta(z_1, 1, (A, g)) = \{(z_2, e)\}$
- (6)  $\delta(z_2, c, (A, f)) = \{(z_1, (A, e)(A, f))\}$

Es ist  $L(K_1) = L_1 = \{1^n(c1^n)^k \mid n \geq 1, k \geq 1\}$

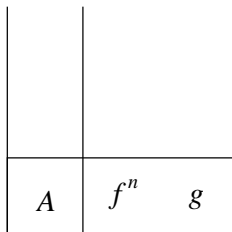
Arbeitsweise von  $K_1$ :

Es werde  $1^n(c1^n)^k$  eingegeben.

Die ersten  $n$  Einsen werden gelesen, dann:

a)

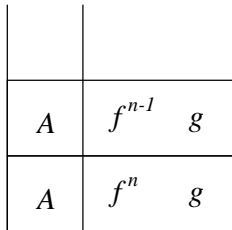
$z_0$



Es wird ein  $c$  gelesen.

b)

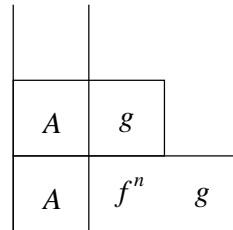
$z_1$



Es werden  $n - 1$  Einsen gelesen:

c)

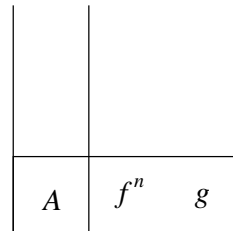
$z_1$



Es wird eine weitere Eins gelesen:

d)

$z_2$



Jetzt kann akzeptiert werden ( $k = 1$ ). Falls  $k > 1$ , wird ein weiteres  $c$  gelesen und die Situation b) hergestellt.

### Beispiel 14.9

$K_2 = (Z, X, \Gamma_1, \Gamma_2, \delta, z_0, A_0, g, F)$  mit  $Z = \{z_0, z_1, z_2, z_3\}$ ,  $X = \{1, c\}$ ,  $\Gamma_1 = \{A_0, A\}$ ,  $\Gamma_2 = \{f, g\}$ ,  $F = \{z_3\}$ .

$\delta$  ist folgendermaßen festgelegt:

- (1)  $\delta(z_0, 1, (A_0, f)) = \{(z_0, (A_0, ff))\}$
- (2)  $\delta(z_0, 1, (A_0, g)) = \{(z_0, (A_0, fg))\}$
- (3)  $\delta(z_0, c, (A_0, f)) = \{(z_0, (A, f)(A_0, f))\}$
- (4)  $\delta(z_0, e, (A, f)) = \{(z_0, (A, e)(A, e))\}$
- (5)  $\delta(z_0, e, (A, g)) = \{(z_1, e)\}$
- (6)  $\delta(z_1, e, (A, g)) = \{(z_2, e)\}$
- (7)  $\delta(z_2, e, (A, f)) = \{(z_2, (A, e)(A, e))\}$
- (8)  $\delta(z_2, 1, (A, g)) = \{(z_2, (A, g))\}$
- (9)  $\delta(z_2, c, (A, g)) = \{(z_2, e)\}$
- (10)  $\delta(z_2, 1, (A_0, f)) = \{(z_2, (A, e))\}$
- (11)  $\delta(z_2, e, (A_0, g)) = \{(z_3, e)\}$

Es ist  $L(K_2) = L_2 = \{1^n \prod_{i=1}^{2^n-1} cw_i | w_i \in 1^*, w_{2^n-1} = 1^n, n \geq 1\}$

Arbeitsweise von  $K_2$ :

Es werden  $1^n \prod_{i=1}^{2^n-1} cw_i, w_i \in 1^*, w_{2^n-1} = 1^n$  eingegeben.

a)

$z_0$

|       |       |     |
|-------|-------|-----|
|       |       |     |
| $A_0$ | $f^n$ | $g$ |

Es wird ein  $c$  gelesen, dann ( $n \geq 1$ ):

b)

$z_0$

|       |       |     |
|-------|-------|-----|
|       |       |     |
| $A$   | $f^n$ | $g$ |
| $A_0$ | $f^n$ | $g$ |

Insgesamt dürfen  $2^n - 1$   $c$ 's gelesen werden.

Es gilt  $2^n - 1 = 2^0 + 2^1 + 2^2 + \dots + 2^{n-1} = \sum_{i=0}^{n-1} 2^i$

Es wird jetzt  $n$ -mal (4) angewendet:

c)

$z_0$

|          |           |     |
|----------|-----------|-----|
|          |           |     |
| $A$      | $g$       |     |
| $A$      | $g$       |     |
| $A$      | $f$       | $g$ |
| $A$      | $f^2$     | $g$ |
| $\vdots$ | $\vdots$  |     |
| $A$      | $f^{n-2}$ | $g$ |
| $A$      | $f^{n-1}$ | $g$ |
| $A_0$    | $f^n$     | $g$ |

Durch Anwendung von (5) und (6) werden die beiden obersten Einträge von  $(A, g)$  gestrichen.

Man gelangt über den Zustand  $z_1$  in den Zustand  $z_2$ .

d)

$z_2$

|          |           |     |
|----------|-----------|-----|
|          |           |     |
| $A$      | $f$       | $g$ |
| $A$      | $f^2$     | $g$ |
| $\vdots$ | $\vdots$  |     |
| $A$      | $f^{n-2}$ | $g$ |
| $A$      | $f^{n-1}$ | $g$ |
| $A_0$    | $f^n$     | $g$ |

Mit Hilfe von (7) werden für den obersten Exponenten 1 (=oberster Eintrag)  $(A, fg)$   $2^1 = 2$  Einträge  $(A, g)$  erzeugt.

e)

$z_2$

|          |           |     |
|----------|-----------|-----|
|          |           |     |
| $A$      | $g$       |     |
| $A$      | $g$       |     |
| $A$      | $f^2$     | $g$ |
| $\vdots$ | $\vdots$  |     |
| $A$      | $f^{n-2}$ | $g$ |
| $A$      | $f^{n-1}$ | $g$ |
| $A_0$    | $f^n$     | $g$ |

Bei einem obersten Eintrag  $(A, g)$  können im Zustand  $z_2$  mit Hilfe von (8) beliebige viele Einsen eingelesen werden ohne die Kellerinhalte zu ändern.

Dies wird mit Hilfe von (9) durch Lesen eines  $c$ 's und Streichen des obersten Kellereintrages beendet.

f)

$z_2$

|          |           |     |
|----------|-----------|-----|
|          |           |     |
| $A$      | $g$       |     |
| $A$      | $f^2$     | $g$ |
| $\vdots$ | $\vdots$  |     |
| $A$      | $f^{n-2}$ | $g$ |
| $A$      | $f^{n-1}$ | $g$ |
| $A_0$    | $f^n$     | $g$ |

Mit Hilfe von (8) können jetzt wieder beliebig viele Einsen gelesen werden, ohne den Kellerinhalt zu ändern.

Dies wird wieder mit (9) durch Lesen eines  $c$ 's beendet.

g)

|       |          |           |     |
|-------|----------|-----------|-----|
| $z_2$ |          |           |     |
|       | $A$      | $f^2$     | $g$ |
|       | $\vdots$ | $\vdots$  |     |
|       | $A$      | $f^{n-2}$ | $g$ |
|       | $A$      | $f^{n-1}$ | $g$ |
|       | $A_0$    | $f^n$     | $g$ |

Allgemein gilt:  $(z_2, \prod_{i=1}^{2^l} w_i c, (A, f^l g)) \vdash^* (z_2, e, e)$ .

$l = 1$ :  $(z_2, w_1 c w_2 c, (A, f g))$  ist klar.

$l \rightarrow l + 1$ :

$$\left( z_2, \prod_{i=1}^{2^{l+1}} w_i c, (A, f^{l+1} g) \right) \vdash^* \left( z_2, \prod_{i=1}^{2^{l+1}} w_i c, (A, f^l g)(A, f^l g) \right) \quad (7)$$

$$\vdash^* \left( z_2, \prod_{i=2^{l+1}}^{2^{l+1}} w_i c, (A, f^l g) \right) \vdash^* (z_2, e, e) \quad (IV)$$

Daraus ergibt sich:

$$\left( z_0, c \prod_{i=1}^{2^n-2} w_i c, (A_0 f^n g) \right) \vdash^* (z_2, e, (A_0, f^n g))$$

Mit Hilfe von (10) und (11) ergibt sich dann:

$$(z_2, 1^n, (A_0, f^n g)) \vdash^*_{(10)} (z_2, e, (A_0, g)) \vdash^* (z_3, e, e)$$

## Abschlußeigenschaften für V-indizierte Sprachen

$L_1 \cap L_2 = \left\{ 1^n \prod_{i=1}^{2^n-1} c 1^n \mid n \geq 1 \right\}$  ist keine V-indizierte Sprache.

also: Der Durchschnitt zweier V-indizierter Sprachen braucht nicht V-indiziert zu sein.

### Satz 14.10

(a)  $L$  sei eine V-indizierte Sprache. Dann ist auch  $L^*$  V-indiziert.

(b)  $L_1$  und  $L_2$  seien V-indizierte Sprachen, dann sind auch  $L_1 \cup L_2$  und  $L_1 \cdot L_2$  V-indiziert.

### Satz 14.11

(a)  $L \subseteq X^*$  sei eine V-indizierte Sprache und  $R \subseteq X^*$  sei eine reguläre Sprache. Dann ist auch  $L \cap R$  V-indiziert.

(b)  $L \subseteq X^*$  sei eine V-indizierte Sprache und  $h : X^* \rightarrow Y^*$  sei ein Homomorphismus. Dann ist auch  $h(L)$  V-indiziert.

(c)  $L \subseteq Y^*$  sei eine V-indizierte Sprache und  $h : X^* \rightarrow Y^*$  sei ein Homomorphismus. Dann ist auch  $h^{-1}(L)$  V-indiziert.

### Satz 14.12

Jede V-indizierte Sprache ist eine R-indizierte Sprache. Es gibt R-indizierte Sprachen, die nicht V-indiziert sind.

Bew.: Die Sprache  $L_R(G) = \{ (\$w)^{|w|} \mid w \in \{a, b\}^* \}$  (vgl. Beispiel 13.6) ist keine V-indizierte Sprache.

(Kann mit einem Iterationslemma gezeigt werden)

$G = (N, T, I, P, S)$  sei eine indizierte Grammatik.

Wir konstruieren eine indizierte Grammatik  $G' = (N', T, I, P', S)$  für die gilt  $L_V(G) = L_R(G')$ .

Setze  $N' = N \cup \{P_f, C_f \mid f \in I\} \cup \{B_f^P, B_f^C \mid B \in N, f \in I\} \cup \{E\}$

$P'$  wird folgendermaßen festgelegt:

Ist  $A \rightarrow \alpha$ ,  $A \in N$ ,  $\alpha \in (N \cup T)^*$  aus  $P$ , dann ist auch  $A \rightarrow \alpha$  aus  $P'$ .

Die Variablen  $P_f$  und  $C_f$  werden dazu benutzt, den Index  $f$  zu produzieren bzw. zu konsumieren.

(1)  $P_f \rightarrow E f$  und  $C_f f \rightarrow E$  sind für jedes  $f \in I$  in  $P'$ .

Weiter ist  $E \rightarrow e$  aus  $P'$ .

Die Indexproduktionen  $A \rightarrow B f$  und  $A f \rightarrow B$  aus  $P$  führen auf folgende Produktionen in  $P'$ :

(2) Ist  $A \rightarrow B f$  aus  $P$ , dann sind  $A \rightarrow B_f^C f$  und  $B_f^C \rightarrow B C_f$  aus  $P'$ .

(3) Ist  $A f \rightarrow B$  aus  $P$ , dann sind  $A f \rightarrow B_f^P$  und  $B_f^P \rightarrow B P_f$  aus  $P'$ .

Erläuterung:  $G$  besitze die Produktionen:

$A \rightarrow B C$      $B f \rightarrow D$      $D \rightarrow C h$

$C \rightarrow a$     (und weitere).

Dann besitzt  $G'$  die Produktionen (und weitere):

$A \rightarrow B C$      $C \rightarrow a$

(1)  $P_f \rightarrow E f$      $P_h \rightarrow E h$      $E \rightarrow e$

$C_f f \rightarrow E$      $C_h h \rightarrow E$

(2)  $D \rightarrow C_h^C h$      $C_h^C \rightarrow C C_h$

(3)  $B f \rightarrow D_f^P$      $D_f^P \rightarrow D P_f$

□

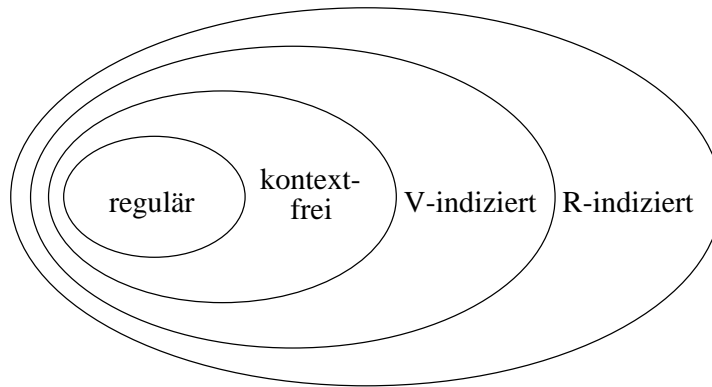
Teil einer V-indizierten Ableitung bzgl.  $G$ :

$A f g \vdash B f g \cdot C f g \vdash D g \cdot C f g \vdash C h g \cdot C f g \vdash a \cdot C f g$

Teil einer R-indizierten Ableitung bzgl.  $G'$ :

$A f g \vdash B f g \cdot C \vdash D_f^P g \cdot C \vdash D g \cdot P_f \cdot C \vdash C_h^C h g \cdot P_f \cdot C \vdash C h g \cdot C_h \cdot P_f \cdot C \vdash a \cdot C_h h g \cdot P_f \cdot C \vdash a \cdot E g \cdot P_f \cdot C \vdash a \cdot e \cdot P_f g \cdot C \vdash a \cdot e \cdot E f g \cdot C \vdash a \cdot e \cdot e \cdot C f g$

# Sprachhierarchie



## Zu den R-indizierten Sprache:

Grammatik  $G = (V, X, S, P)$  (siehe Definition 1.4)

$P =$  endliche Menge von Produktionen  $(u, v)$  und  $u, v \in (V \cup X)^*$  und  $u$  enthält mindestens eine Variable.

Schreibweise:  $u \rightarrow v$

Grammatiken dieses Types heißen Typ-0-Grammatiken (allgemeinster Begriff einer Grammatik)

Erzeugte Sprachen = Type-0-Sprachen (= von Turing-Maschinen akzeptierte Sprachen = rekursiv abzählbare Sprachen)

### Satz 14.13

Die Typ-0-Sprachen sind genau die R-indizierten Sprachen.

### Definition 14.14 (kontext-sensitiv)

Eine Grammatik  $G = (V, X, S, P)$  heißt **kontext-sensitiv** oder **Typ-1-Grammatik**, wenn alle Produktionen, außer evtl.  $S \rightarrow e$ , von der Form  $\alpha A \beta \rightarrow \alpha \gamma \beta$  mit  $A \in V, \alpha, \beta, \gamma \in (V \cup X)^*, |\gamma| \geq 1$  sind. Ist  $S \rightarrow e \in P$ , dann tritt  $S$  in keiner rechten Seite einer Produktion auf.

### Satz 14.15

Jede V-indizierte Sprache ist kontext-sensitiv. Es gibt kontext-sensitive Sprachen, die nicht V-indiziert sind.

# Kapitel 15

## Lineare Sprache

Eine kontextfreie Grammatik  $G = (V, X, S, P)$  heißt **linear**, wenn alle Produktionen von der Form  $A \rightarrow uBv$  oder  $A \rightarrow u$  mit  $A, B \in V$  und  $u, v \in X^*$  sind.

$L = \{a^n b^n | n \geq 1\}$  ist linear kontextfrei, aber nicht rechtslinear, also:

- (a) Jede rechtslineare Sprache ist eine linear kontextfreie Sprache.
- (b) Es gibt linear kontextfreie Sprachen, die nicht rechtslinear sind.

Frage: Ist jede kontextfreie Sprache eine linear-kontextfreie Sprache?

### Definition 15.1 (gsm)

Ein gsm (generalised sequential machine) ist ein 6-Tupel  $S = (Z, X, Y, \delta, \lambda, z_0)$  mit:

- 1)  $Z$  ist eine endliche Menge (Zustandsmenge)
- 2)  $X$  ist ein Alphabet (Eingabealphabet)
- 3)  $Y$  ist ein Alphabet (Ausgabealphabet)
- 4)  $\delta : Z \times X \rightarrow Z$  (Überföhrungsfunktion)
- 5)  $\lambda : Z \times X \rightarrow Y^*$  (Ausgabefunktion)
- 6)  $z_0 \in Z$  (Anfangszustand)

Erweiterung von  $\delta$  zu  $\bar{\delta} : Z \times X^* \rightarrow Z$  durch:

$$\forall z \in Z : \bar{\delta}(z, e) = z$$

$$\forall z \in Z, \forall u \in X^*, \forall x \in X : \bar{\delta}(z, ux) = \bar{\delta}(\bar{\delta}(z, u), x)$$

Erweiterung von  $\lambda$  zu  $\bar{\lambda} : Z \times X^* \rightarrow Y^*$  durch:

$$\forall z \in Z : \bar{\lambda}(z, e) = \epsilon$$

$$\forall z \in Z, \forall u \in X^*, \forall x \in X : \bar{\lambda}(z, xu) = \bar{\lambda}(z, x)\bar{\lambda}(\bar{\delta}(z, x), u)$$

### Definition 15.2 (Übersetzung eines gsm)

$S = (Z, X, Y, \delta, \lambda, z_0)$  sei eine gsm und  $L \subseteq X^*$ . Die **Übersetzung von  $L$  bzgl.  $S$**  ist die Sprache  $S(L) = \{\bar{\lambda}(z_0, w) | w \in L\}$

### Satz 15.3

$S, L$  wie oben. Ist  $L \left\{ \begin{array}{l} \text{regulär} \\ \text{kontextfrei} \end{array} \right\}$ , dann ist auch  $S(L) \left\{ \begin{array}{l} \text{regulär} \\ \text{kontextfrei} \end{array} \right\}$ .

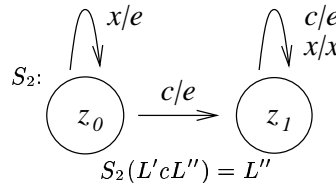
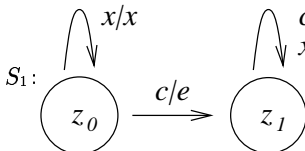
### Satz 15.4

$X$  sei ein Alphabet,  $L', L'' \subseteq X^*$  und  $c \notin X$ . Ist dann  $L'cL''$  eine linear kontextfreie Sprache, dann ist  $L'$  oder  $L''$  regulär.

Bew.:  $G = (V, X \cup \{c\}, S, P)$  sei eine linear kontextfreie Grammatik mit  $L = L(G) = L'cL''$ .

O.B.d.A.: Jede Variable  $A \in V$  kommt in mindestens einer Ableitung eines terminalen Wortes vor.

Man lege zwei gsm's  $S_1 = (\{z_0, z_1\}, X \cup \{c\}, X, \delta_1, \lambda_1, z_0)$  und  $S_2 = (\{z_0, z_1\}, X \cup \{c\}, X, \delta_2, \lambda_2, z_0)$  folgendermaßen fest:



$x \in X$

Es gilt:  $S_1(L'cL'') = L'$

Aufgrund der Annahme über  $G$  gilt:

Aus  $A \in V$ ,  $w \in (X \cup \{c\})^*$  und  $A \xrightarrow{*} w$  folgt:  $w$  enthält höchstens ein  $c$ .

Setze  $V_c = \{A | A \in V, \exists A \xrightarrow{*} w, w \in X^*cX^*\}$

Gilt  $A \xrightarrow{*} uBv$ , dann

(a)  $uv$  enthalten höchstens ein  $c$ .

(b) enthält  $uv$  ein  $c$ , dann ist  $B \notin V_c$

Zerlegung von  $P$  in vier disjunkte Teilmengen:

$P_0 = \{A \rightarrow u | u \in X^*\} \cup \{A \rightarrow uBv | B \in V, u, v \in X^*\}$

$P_1 = \{A \rightarrow uc | u \in X^*\}$

$P_2 = \{A \rightarrow u_1cu_2Bv | B \in V, u_1, u_2, v \in X^*\}$

$$P_3 = \{A \rightarrow uBv_1cv_2 \mid B \in V, u, v_1, v_2 \in X^*\}$$

Es gilt: In einer Ableitung  $A \xRightarrow{*} w, w \in L$ , können Produktionen aus  $P_0$  und genau eine Produktion aus  $P_1 \cup P_2 \cup P_3$  angewendet werden.

Setze  $G_1 = (V, X \cup \{c\}, S, P_0 \cup P_1)$

$$G_2 = (V, X \cup \{c\}, S, P_0 \cup P_2)$$

$$G_3 = (V, X \cup \{c\}, S, P_0 \cup P_3) \text{ und } L_i = L(G_i) \text{ f\u00fcr } i \in [1 : 3].$$

Es gilt:  $L = L_1 \cup L_2 \cup L_3$ .

Beh. 1:  $S_1(L_1)$ ,  $S_1(L_2)$  und  $S_2(L_3)$  sind regul\u00e4re Sprachen

Bew.:  $G'_1 = (V, X, S, P'_1)$  mit  $P'_1 = \{A \rightarrow uB \mid A \rightarrow uBv \in P_0\} \cup \{A \rightarrow u \mid A \rightarrow ucv \in P_1\}$  erzeugt  $S_1(L_1)$ .

$G'_2 = (V, X, S, P'_2)$  mit  $P'_2 = \{A \rightarrow uB \mid A \rightarrow uBv \in P_0\} \cup \{A \rightarrow u_1 \mid A \rightarrow u_1cu_2Bv \in P_2\}$  erzeugt  $S_1(L_2)$ .

$G'_3 = (V, X, S, P'_3)$  mit  $P'_3 = \{A \rightarrow Bv \mid A \rightarrow uBv \in P_0\} \cup \{A \rightarrow v_2 \mid A \rightarrow uBv_1cv_2 \in P_3\}$  erzeugt  $S_2(L_3)$ .

Es ist  $L = L'cL'' = L_1 \cup L_2 \cup L_3$ . Also:

$$(1) L' = S_1(L) = S_1(L_1) \cup S_1(L_2) \cup S_1(L_3)$$

$$(2) L'' = S_2(L) = S_2(L_1) \cup S_2(L_2) \cup S_2(L_3)$$

Insbesondere gilt:  $S_2(L_3) \subseteq L''$

Beh. 2: Es gilt  $L' = S_1(L_1) \cup S_1(L_2)$  oder  $L'' = S_2(L_3)$ .

Bew.: Es ist  $S_1(L_1) \cup S_1(L_2) \subseteq L'$

Sei  $S_1(L_1) \cup S_1(L_2) \not\subseteq L'$ , also  $\exists w \in L' \setminus (S_1(L_1) \cup S_1(L_2))$ . Es ist:

$$(3) w \in L'' \subseteq L'cL'' = L_1 \cup L_2 \cup L_3.$$

Aus  $S_1(wcL'') = w \notin S_1(L_1) \cup S_1(L_2)$  folgt

$$(4) w \in L'' \cap (L_1 \cup L_2) = \emptyset$$

Aus (3) und (4) folgt:  $w \in L'' \subseteq L_3$ , also  $S_2(wcL'') = L'' \subseteq S_2(L_3)$

Wegen (2) folgt:  $L'' = S_2(L_3)$ .

Aus Beh. 1 und Beh. 2 folgt die Aussage des Satzes. □

### Korollar 15.5

$X$  sei ein Alphabet,  $c \notin X$  und  $L \subseteq X^*$ . Ist  $LcL$  linear kontextfrei, dann ist  $L$  regul\u00e4r.

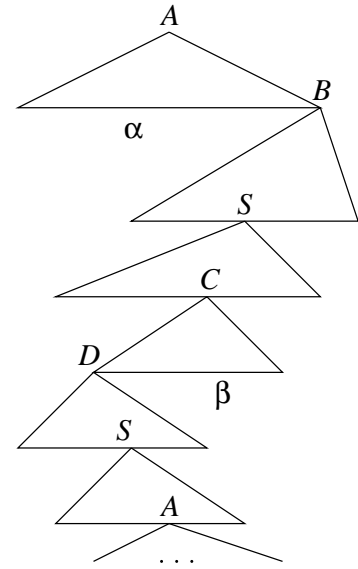
### Satz 15.6

Es gilt kontextfreie Sprache, die nicht linear kontextfrei sind.

Bew.: Betrachte  $G = (\{S, T\}, \{a, b\}, S, P)$  mit  $P = \{S \rightarrow TcT, T \rightarrow aTb \mid e\}$ .

Dann ist  $L(G) = LcL$  mit  $L = \{a^n b^n \mid n \geq 0\}$  □





Es gilt:  $P$  enthält nicht gleichzeitig Produktionen vom Typ (2) und (3), dazu:  
 Ann.:  $P$  enthalte eine Produktion  $A \rightarrow \alpha B$ ,  $C \rightarrow D\beta$  mit  $\alpha, \beta \in (V \cup X)^+$   
 $\exists B \xrightarrow{*} \alpha_1 S \beta_1$ ,  $\alpha_1, \beta_1 \in (V \cup X)^*$ . Da  $G$  reduziert ist, gibt es  
 $S \xrightarrow{*} \alpha_2 C \beta_2$ ,  $\alpha_2, \beta_2 \in (V \cup X)^*$ . Weiter gilt:  
 $\exists D \xrightarrow{*} \alpha_3 S \beta_3$ ,  $\alpha_3, \beta_3 \in (V \cup X)^*$  und  
 $S \xrightarrow{*} \alpha_4 A \beta_4$ ,  $\alpha_4, \beta_4 \in (V \cup X)^*$   
 also:  $A$  ist selbsteinbettend (Widerspruch!)

Es gilt: Wenn  $P$  eine Produktion  $A \rightarrow \alpha B$  vom Typ (2) enthält, dann ist  $\alpha \in X^+$ .

Ann.:  $\alpha \notin X^+$ , damit ist  $\alpha = \beta_1 C \beta_2$

- a) Sei  $\beta_1 \neq e$ , dann ist  $A \rightarrow \beta_1 C \beta_2 B$  vom Typ (1) (Widerspruch!)
- b) Sei  $\beta_1 = e$ , dann ist  $A \rightarrow C \beta_2 B$  sowohl vom Typ (2) als auch vom Typ (3). (Widerspruch!)

Es gilt dementsprechend: Wenn  $P$  eine Produktion  $A \rightarrow B\beta$  vom Typ (3) enthält, dann ist  $\beta \in X^+$ .

Also:  $P$  enthält nur Produktionen vom Typ (2),(4),(5) oder vom Typ (3),(4),(5), also:  $L(G)$  ist regulär.

zu Fall 2: Durch Induktion über  $|V|$  wird gezeigt:

$G = (V, X, S, P)$  sei nicht selbsteinbettend, reduziert und es liege Fall 2 vor, dann folgt:  $L(G)$  ist regulär.

a)  $|V| = 1$  (klar)

b) Die Behauptung gelte für alle  $G = (V, X, S, P)$  wie oben mit  $|V| \leq n$ .

Sei  $G = (V, X, S, P)$  wie oben und  $|V| = n + 1$

Sei  $A \in V$  mit  $\bar{A}A \xrightarrow{*} \alpha S \beta$  (Es ist  $A \neq S$ ).

Setze  $G_1 := (V \setminus \{A\}, X \cup \{A\}, S, P_1)$  mit  $P_1 = \{B \rightarrow \alpha \mid \alpha \in P \text{ und } B \neq A\}$  (Es ist  $L(G_1) \neq \emptyset$ )

Setze  $G_2 := (V \setminus \{S\}, X, A, P_2)$  mit  $P_2 = \{B \rightarrow \alpha \mid B \rightarrow \alpha \in P, B \neq S, \alpha \in (V \setminus \{S\} \cup X)^*\}$  (Es ist  $L(G_2) \neq \emptyset$ )

Reduziere  $G_1$  zu  $G_1^r$  und  $G_2$  zu  $G_2^r$ .

Für  $G_1^r \begin{cases} \text{liegt Fall 1 vor, dann ist } L(G_1) = L(G_1^r) \text{ regulär.} \\ \text{liegt Fall 2 vor, dann ist } L(G_1) = L(G_1^r) \text{ nach IV regulär.} \end{cases}$

entsprechend für  $G_2^r$ .

Es gilt: Eine Ableitung  $S \xrightarrow{*}_G w$  kann aufgeteilt werden in  $S \xrightarrow{*}_{G_1} u_1 A u_2 A u_3 \dots u_n A_n u_{n+1} \xrightarrow{*}_{G_2} \overbrace{u_1 w_1 u_2 w_2 u_3 \dots u_n w_n u_{n+1}}^w$ .

Betrachte folgende Substitution:  $\varphi: X \cup \{A\} \rightarrow \mathbb{P}(X^*)$  mit  $\varphi(x) = \{x\}$  für  $x \in X$  und  $\varphi(A) = L(G_2)$ .

$\varphi$  ist eine reguläre Substitution.

$L(G_1)$  ist eine reguläre Sprache über  $X \cup \{A\}$ .

Es gilt: Reguläre Sprachen sind abgeschlossen gegenüber regulären Substitutionen, also ist  $\varphi(L(G_1)) = L(G)$  regulär. □

# Kapitel 17

## Semilineare Mengen und der Satz von Parikh

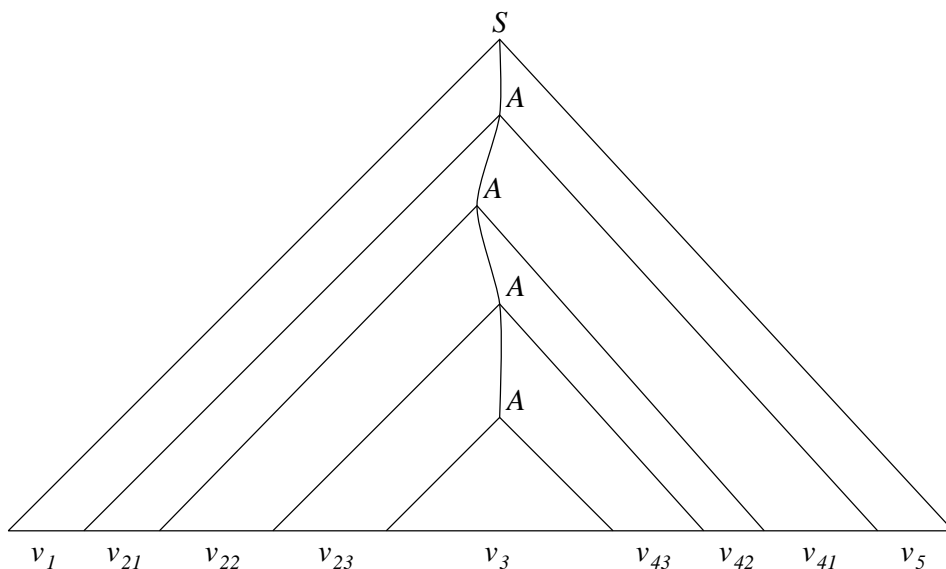
Variante des Iterationsatzes für kontextfreie Sprachen:

**Definition 17.1** (Paring-Lemma)

Für jede kontextfreie Grammatik  $G = (V, X, S, P)$  gibt es eine ganze Zahl  $k \geq 1$ , so daß für jedes  $l \geq 1$  gilt:

Für jedes  $w \in L(G)$  mit  $|w| \geq k^l$  gibt es Wörter  $v_1, v_3, v_5, v_{21}, \dots, v_{2l}, v_{41}, \dots, v_{4l} \in X^*$  und  $A \in V$  mit:

- 1)  $S \xrightarrow{*} v_1 A v_2 \xrightarrow{*} v_1 v_{21} A v_{41} v_5 \xrightarrow{*} v_1 v_{21} v_{22} A v_{42} v_{41} v_5 \xrightarrow{*} v_1 v_{21} \dots v_{2l} A v_{41} \dots v_{4l} v_5$   
 $\xrightarrow{*} v_1 v_{21} \dots v_{2l} v_{3l} v_{4l} \dots v_{41} v_5$ .
- 2)  $v_{2i} v_{4i} \neq \epsilon$  für  $i \in [1 : l]$ .
- 3)  $|v_{21} \dots v_{2l} v_{3l} v_{4l} \dots v_{41} v_5| \leq k^l$ .



Es gilt:  $l = 1$ : Iterationslemma für kontextfreie Grammatiken (Sprachen).

Beweisidee (sonst): In einem Ableitungsbaum eines Wortes  $w$  mit  $|w| \geq k^l$  gibt es einen Weg von der Wurzel zu einem Blatt, der mindestens  $(l + 1)$ -mal eine Variable  $A$  enthält.

$\mathbb{N}$ : Menge der nichtnegativen, ganzen Zahlen ist kommutatives Monoid bzgl.  $+$ .

Sei  $a = (a_1, \dots, a_n) \in \mathbb{N}^n$  und  $b = (b_1, \dots, b_n) \in \mathbb{N}^n$ ,  $n \geq 1$ .

Setze  $a + b := (a_1 + b_1, \dots, a_n + b_n)$  und  $m \cdot a := (ma_1, \dots, ma_n)$ ,  $m \in \mathbb{N}$ .

**Definition 17.2** (semi-lineare Teilmenge, Parikh-Abbildung)

Seien  $a_0, a_1, \dots, a_m$  ( $m \geq 0$ )  $n$ -tupel aus  $\mathbb{N}^n$ . Die Menge  $\{a_0 + n_1 \cdot a_1 + \dots + n_m \cdot a_m \mid n_j \in \mathbb{N} \text{ für } j \in [1 : m]\}$  heißt **lineare Teilmenge von  $\mathbb{N}^n$** .

Die Vereinigung endlich vieler linearer Teilmengen von  $\mathbb{N}^n$  heißt **semi-lineare Teilmenge von  $\mathbb{N}^n$** .

Sei  $X = \{a_1, a_2, \dots, a_n\}$  ein Alphabet.

Die Abbildung  $\Psi : X^* \rightarrow \mathbb{N}^n$  sei folgendermaßen festgelegt:  $\Psi(w) = (|w|_{a_1}, |w|_{a_2}, \dots, |w|_{a_n})$ .

Hierbei ist  $|w|_{a_i}$  die Anzahl des Auftretens von  $a_i$  in  $w$ .

$\Psi$  heißt Parikh-Abbildung.

Es gilt:  $\Psi(w_1 w_2) = \Psi(w_1) + \Psi(w_2) = \Psi(w_2) + \Psi(w_1) = \Psi(w_2 w_1)$ .

Für  $L \subseteq X^*$  setze  $\Psi(L) = \{\Psi(w) \mid w \in L\}$  "Parikh-Bild von  $L$ ", " $\Psi$ -Bild von  $L$ "

### Beispiel 17.3

$L = \{a^n b^n | n \geq 0\}$ , dann ist  $\Psi(L) = \{(n, n) | n \geq 0\} = \{(0, 0) + n(1, 1) | n \in \mathbb{N}\}$  ist (semi-)lineare Teilmenge von  $\mathbb{N}^2$ .

### Definition 17.4 (buchstabenäquivalent)

Zwei Sprachen  $L_1, L_2 \subseteq X^*$  heißen **buchstabenäquivalent**, wenn  $\Psi(L_1) = \Psi(L_2)$  gilt.

### Satz 17.5

Sei  $L \subseteq X^*$ . Dann ist  $\Psi(L)$  genau dann semi-linear, wenn  $L$  buchstabenäquivalent zu einer regulären Sprache ist.

Bew.: Sei  $\Psi(L) \subseteq \mathbb{N}^n$  semilinear, also:

$$\Psi(L) = M_1 \cup M_2 \cup \dots \cup M_l, M_i \text{ linear, } i \in [1 : l], l \geq 0$$

$$\text{Sei } M_i = \{a_{i0} + n_{i1}a_{i1} + \dots + n_{i r_i}a_{i r_i} | n_{ij} \in \mathbb{N}, j \in [1 : r_i], r_i \geq 0\}$$

$$v_{i0}, \dots, v_{i r_i} \text{ seien Wörter aus } X^* \text{ mit } \Psi(v_{ij}) = a_{ij}, j \in [0 : r_i].$$

Setze  $G = (V, X, S, P)$  mit  $V = \{S, A_1, \dots, A_l\}$  und folgenden Produktionen:

$$P : S \rightarrow A_1 \dots A_l$$

$$A_i \rightarrow A_i v_{ij} | v_0 \text{ für } j \in [1 : r_i], i \in [1 : l]$$

$G$  ist semilinear (bis auf die singulären Produktionen), also ist  $L(G)$  regulär und es gilt:  $\Psi(L(G)) = \Psi(L)$

Umgekehrt: Es wird gezeigt: Jede reguläre Sprache hat semilineare  $\Psi$ -Bild.

Es sind  $\Psi(\emptyset) = \emptyset$  und

$$\Psi(\{a_i\}) = \{(0, \dots, 0, 1, 0, \dots, 0)\} \text{ lineare Mengen.}$$

Es wird jetzt gezeigt:  $\Psi(L_1), \Psi(L_2)$  und  $\Psi(L)$  seien semi-linear, dann sind auch

$$(1) \Psi(L_1 \cup L_2)$$

$$(2) \Psi(L_1 \cdot L_2)$$

$$(3) \Psi(L^*) \text{ semi-linear.}$$

zu (1): Sei  $\Psi(L_1) = M_{11} \cup \dots \cup M_{1 l_1}$  und  $\Psi(L_2) = M_{21} \cup \dots \cup M_{2 l_2}$ .

$$M_{1 j_1}, M_{2 j_2} \text{ sind linear für } j_1 \in [1 : l_1], j_2 \in [1 : l_2], l_1, l_2 \geq 0.$$

$$\text{Es ist } \Psi(L_1 \cup L_2) = M_{11} \cup \dots \cup M_{1 l_1} \cup M_{21} \cup \dots \cup M_{2 l_2} \text{ semilinear.}$$

zu (2): Es ist  $\Psi(L_1 \cdot L_2) = \Psi(L_1) + \Psi(L_2) = (M_{11} \cup \dots \cup M_{1 l_1}) + (M_{21} \cup \dots \cup M_{2 l_2}) =$

$$(M_{11} + M_{21}) \cup \dots \cup (M_{11} + M_{2 l_2}) \cup$$

$$(M_{12} + M_{21}) \cup \dots \cup (M_{12} + M_{2 l_2}) \cup$$

$$\vdots$$

$$\vdots$$

$$(M_{1 l_1} + M_{21}) \cup \dots \cup (M_{1 l_1} + M_{2 l_2})$$

$$\text{Es gilt: } M_1, M_2 \text{ linear} \implies M_1 + M_2 \text{ linear.}$$

zu (3):  $\Psi(L)$  sei zunächst linear.

$$\Psi(L) = M \text{ mit } M = \{a_0 + n_1 a_1 + \dots + n_l a_l | n_i \in \mathbb{N}, i \in [1 : l], l \geq 0\}$$

$$\text{Dann ist } \Psi(L^*) = \{0\} \cup \{a_0 + n_0 a_0 + n_1 a_1 + \dots + n_l a_l | n_i \in \mathbb{N}, i \in [0 : l]\} \text{ semi-linear.}$$

$$\text{Sei } \Psi(L) = M_1 \cup \dots \cup M_m, M_i \text{ linear, } i \in [1 : m], m \geq 0$$

$$\text{Induktion: } m=0: \Psi(L) = \emptyset, \text{ also } L = \emptyset \text{ und } \Psi(L^*) = \Psi(\emptyset^*) = \Psi(\{e\}) = \{0\} \text{ linear.}$$

$$m=1: \text{ s.o.}$$

$$m \geq 2: \text{ Es ist } L = L_1 \cup L_2 \cup \dots \cup L_m \text{ mit } \Psi(L_i) = M_i, i \in [1 : m].$$

Die Sprachen  $L_1^*, L_2^*, \dots, L_m^*$  haben semilineare Bilder bzgl.  $\Psi$ , also:

$$\Psi(L_1^* L_2^* \dots L_m^*) = \Psi(L_1^*) + \Psi(L_2^*) + \dots + \Psi(L_m^*) \text{ semilinear.}$$

$$\Psi(L^*) = \Psi((L_1 \cup L_2 \cup \dots \cup L_m)^*) = \Psi(L_1^* L_2^* \dots L_m^*), \text{ also ist } \Psi(L) \text{ semilinear.}$$

□

### Satz 17.6 (Satz von Parikh)

Für jede kontextfreie Sprache  $L$  ist  $\Psi(L)$  semilinear.

(Es ist also jede kontextfreie Sprache buchstabenäquivalent zu einer regulären Sprache).

Bew.: Sei  $G = (V, X, S, P)$  kontextfrei mit  $L = L(G)$ .

$k$  sei die Konstante aus dem Paring-Lemma.

Sei  $\alpha \in (V \cup X)^*$ . Setze  $\nu(\alpha) = \{A | A \in V \text{ und } A \text{ tritt in } \alpha \text{ auf}\}$ .

Erweiterung von  $\nu$  auf Ableitungen: Sei  $\Theta : \alpha_1 \implies \alpha_2 \implies \dots \implies \alpha_n$  eine Ableitung.

$$\text{Setze } \nu(\Theta) = \bigcup_{i=1}^n \nu(\alpha_i)$$

Sei  $U \subseteq V$  mit  $S \in U$ .

$$\text{Setze } L_U = \{w | w \in X^* \text{ und } \exists \Theta : S \xRightarrow{*} w \text{ mit } \nu(\Theta) = U\}.$$

$$\text{Es ist } L = \bigcup_U L_U.$$

Es genügt zu zeigen: Für jedes  $U$  ist  $\Psi(L_U)$  semilinear oder

für jedes  $U$  ist  $\Psi(L_U)$  buchstabenäquivalent zu einer regulären Sprache.

Sei  $|U| = l$ . Setze  $F = \{w | w \in L_U \text{ und } |w| \leq k^l\}$ .

Betrachte die endliche Menge  $\{uw | u, v \in X^*, \exists B \in V, \exists \Theta : B \xRightarrow{*} uBv, \nu(\Theta) \subseteq U, |uw| \leq k^l\}$ .

Die Elemente dieser Menge werden durchnummeriert als  $s_1, s_2, \dots, s_m$ .

Es wird gezeigt:  $L_U$  ist buchstabenäquivalent zu  $F s_1^* \dots s_m^*$ .

1. Beh.:  $\Psi(L_U) \subseteq \Psi(F s_1^* \dots s_m^*)$ .

Bew.: Induktion über  $n = |w|$ ,  $w \in L_U$ .

$$n = 0 : w = e, \text{ also } w = e \in F, \text{ also } w = e \in F s_1^* \dots s_m^*.$$

$$n \rightarrow n + 1 : \text{ Sei } |w| = n + 1.$$

a)  $|w| < k^l$  (klar nach Induktionsvoraussetzung).

b)  $|w| \geq k^l$ . Da  $w \in L_U$ , gibt es eine Ableitung  $\Theta : S \xrightarrow{*} w$  mit  $\nu(\Theta) = U$ .

Da  $|w| \geq k^l$ , kann  $\Theta$  umgeordnet werden in eine Ableitung  $\Theta'$  (s. Paring-Lemma)

$$S \xrightarrow[\Theta_0]{*} v_1 A v_5 \xrightarrow[\Theta_1]{*} v_1 v_{21} A v_{41} v_5 \xrightarrow[\Theta_2]{*} \dots \xrightarrow[\Theta_l]{*} v_1 v_{21} \dots v_{2l} A v_{4l} \dots v_{41} v_5 \xrightarrow[\Theta_{l+1}]{*} v_1 v_{21} \dots v_{2l} v_{3l} v_{4l} \dots v_{41} v_5.$$

$$\text{Es ist } \nu(\Theta) = \bigcup_{i=0}^{l+1} \nu(\Theta_i) = U.$$

Jede Variable  $U \setminus \{A\}$  kommt in mindestens einem  $\nu(\Theta_i)$  vor,  $i \in [0 : l + 1]$ .

Für jede Variable aus  $U \setminus \{A\}$  wähle ein  $i \in [0 : l + 1]$ , so daß  $\nu(\Theta_i)$  diese Variable enthält.

Es gibt genau  $l - 1$  Variablen in  $U \setminus \{A\}$ , also: mindestens ein  $i \in [1 : l]$  wurde nicht ausgewählt. Dieses sei  $i_0$ . wählt. Dieses sei  $i_0$ .

$$\Theta'' : S \xrightarrow{*} v_1 v_{21} \dots v_{2i_0-1} v_{2i_0+1} \dots v_{2l} S v_{4l} \dots v_{4i_0+1} v_{4i_0-1} \dots v_{41} v_5 = w'.$$

Es ist  $\nu(\Theta'') = U$ , also ist  $w' \in L_U$ . Weiter ist  $|w'| < |w| = n + 1$ , also gilt (lt. IV):  $\Psi(w') \in \Psi(F s_1^* \dots s_m^*)$ .

$w' v_{2i_0} v_{4i_0}$  ist Permutation von  $w$ .

Es ist  $v_{2i_0} v_{4i_0} = s_j$  für ein  $j \in [1 : m]$ , also  $s_j^* v_{2i_0} v_{4i_0} \subset s_j^*$  und damit gilt:

$$\Psi(w) = \Psi(w' v_{2i_0} v_{4i_0}) \in \Psi(F s_1^* \dots s_m^* v_{2i_0} v_{4i_0}) \subseteq \Psi(F s_1^* \dots s_m^*).$$

2. Beh.:  $\Psi(F s_1^* \dots s_m^*) \subseteq \Psi(L_U)$ .

Bew.: Sei  $w \in L_U$  und  $i \in [1 : m]$ .

Es ist  $s_i = uv$  und es gibt  $B \in V$  mit  $\Theta : B \xrightarrow{*} uBv$  mit  $\nu(\Theta) \subseteq U$ . (Es ist insbesondere  $B \in U$ .)

Da  $w \in L_U$  gibt es eine Ableitung  $\Theta'$  von  $w$  mit  $\nu(\Theta') = U$ , in der auch  $B$  vorkommt, o.B.d.A.

$$\Theta' : S \xrightarrow{*} \bar{u}B\bar{v} \xrightarrow{*} \bar{u}\bar{u}\bar{v} = w.$$

Dann gilt auch:  $\Theta'' : S \xrightarrow{*} \bar{u}B\bar{v} \xrightarrow{*} \bar{u}uBv\bar{v} \xrightarrow{*} \bar{u}\bar{u}\bar{v}\bar{v} = w'$  mit  $\nu(\Theta'') = U$ , also ist  $w' \in L_U$ .

Weiter gilt:  $\Psi(w') = \Psi(ws_i)$ .

Also: Ist  $w \in L_U$ , dann gilt: Für jedes  $s_i$  gibt es ein  $w' \in L_U$  mit  $\Psi(w') = \Psi(ws_i)$ .

Durch wiederholte Anwendung dieser Aussage ergibt sich:

$$\text{Ist } ws_1^{n_1} \dots s_m^{n_m} \in F s_1^* \dots s_m^* \text{ (} w \in F \text{), dann gibt es ein } \tilde{w} \in L_U \text{ mit } \Psi(ws_1^{n_1} \dots s_m^{n_m}) = \Psi(\tilde{w}).$$

Also:  $\Psi(F s_1^* \dots s_m^*) \subseteq \Psi(L_U)$ . □

# Kapitel 18

## Literaturliste

### **Hopcraft, J.E. und Ullmann, J.D.:**

Introduction to Automata Theory, Languages und Computation  
Addison-Wesley 1979

### **Parchmann, R. und Duske, J.:**

Grammars, derivation modes, and properties of indexed and type-0-languages  
Theoretical Computer Science 49 (1987), pp. 23-42

# Index

## A

- Abbildung
  - semi-linear, 45
- Ableitung, 1
  - smodus, 32, 33
- Automat
  - deterministisch, 3
  - Keller-, 27
  - V-indiziert, 36
  - nichtdeterministisch, 4

## B

- Buchstabe, 1

## D

- DFA, 3

## E

- e-Überführung, 27
- e-Bedingung, 14

## G

- Generalized Sequential Machine, 41
- Grammatik, 1
  - Äquivalenz von, 14
  - eigentlich, 16
  - indiziert, 32
  - kontext-sensitiv, 40
  - kontextfrei, 12
    - linear, 12
  - linear, 41
    - rechtslinear, 3
  - rekursiv, 23
    - linksrekursiv, 23
    - rechtsrekursiv, 23
  - selbsteinbettend, 43
  - zyklenfrei, 16
- GSM, 41

## H

- Homomorphismus, 11

## K

- Kellerautomat, 27
  - V-indiziert, 36
- Konstante, 1

## M

- Monoid, 1

## N

- NFA, 4
- Nicht-Terminale, 1
- Normalform
  - Chomsky, 18
  - Greibach, 25

## O

- Ordnung

- linear, 25
- partiell, 25

## P

- PDA, 27
- Produktion
  - singuläre, 16
- Produktionensystem, 1

## R

- Rechtslinearität, 3

## S

- Satz, 1
  - form, 1
- Sprache, 1
  - bunchstabenäquivalent, 46
  - DFA, 3
  - Dyck, 18
  - elementar, 7
  - indiziert, 33
  - kontextfrei, 12
    - linear, 41
    - NFA, 4
- Startsymbol, 1
- Startvariable, 1
- Substitution, 11

## T

- Terminale, 1

## V

- V-i PDA, 36
- Variable, 1

## W

- Wort, 1

## Z

- Zeichen
  - nutzlos, 15
  - unerreichbar, 15