Entscheidbarkeiten bei kontextfreien Sprachen und Kuroda-Normalform für kontextsensitive Grammatiken

Prof. Dr. David Sabel

LFE Theoretische Informatik

Letzte Änderung der Folien: 20. Juni 2022

Ziele

Zunächst:

- Entscheidbare Probleme für Kontextfreie Grammatiken
- Resultate haben wir z.T. schon gesehen, aber Beweise fehlten

Danach:

- LBA-Probleme

Leerheitsproblem

Satz

Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis:

- Sei L als CFG gegeben
- Prüfe zunächst, ob $\varepsilon \in L$ (wenn ja, dann ist L nicht leer)
- Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-NF mit $L(G) = L \setminus \{\varepsilon\}$.
- Der folgende Algorithmus markiert alle $A \in V$ mit $\{w \in \Sigma^* \mid A \Rightarrow_G^* w\} \neq \emptyset$
- Prüfe, ob S markiert wird (wenn ja, dann ist L nicht-leer)

Algorithmus 9: Markierung der Variablen, die nichtleere Sprachen erzeugen

Eingabe: Grammatik $G = (V, \Sigma, P, S)$ in Chomsky-Normalform
Ausgabe: Menge $W \subseteq V$ aller Variablen, die nicht die leere Sprache erzeugen
Beginn

$W := \{A \in V \mid A \to a \in P, a \in \Sigma\}$;
wiederhole
  $W_{alt} := W$;
  $W := W_{alt} \cup \{A \mid A \to BC \in P, B \in W_{alt}, C \in W_{alt}\}$;
bis $W = W_{alt}$;
return W

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt False

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt False
- 2. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{A, C\}$

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt False
- 2. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{A, C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt False
- 2. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{A, C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt True

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt False
- 2. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{A, C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt True
- return $W = \{A, C\}$

Beispiel

Sei $G = (\{S, A, B, C\}, \{a, b\}, \{S \to AB, C \to CC \mid a \mid b, A \to CC \mid BB, B \to CB\})$
Ausführung von Algorithmus 9:

- $W := \{A \in V \mid A \to a \in P\} = \{C\}$
- 1. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt False
- 2. Durchlauf der wiederhole-Schleife:
  - $W_{alt} := W = \{A, C\}$
  - $W := W_{alt} \cup \{A \mid A \to BC \in P, B, C \in W_{alt}\} = \{A, C\} \cup \{A, C\} = \{A, C\}$
  - prüfe $W = W_{alt}$ ergibt True
- return $W = \{A, C\}$

Da $S \notin W$ folgt, dass G die leere Sprache erzeugt.

Endlichkeitsproblem

Satz

Das Endlichkeitsproblem für kontextfreie Sprachen ist entscheidbar.

Beweis: Sei $G = (V, \Sigma, P, S)$ eine CFG in Chomsky-NF. Sei n die Zahl aus dem Pumping-Lemma für CFLs (z.B. $n = 2^{|V|}$ siehe Beweis des Pumping-Lemma für CFLs).

Wir zeigen zunächst:

Es gilt $|L(G)| = \infty$ g.d.w. es ein Wort $z \in L(G)$ mit $n \leq |z| < 2n$ gibt.

„$\Leftarrow$":

- Sei $z \in L$ mit $|z| \geq n$.
- Pumping-Lemma zeigt: $uv^iwx^iy \in L$ für alle $i \in \mathbb{N}$.
- Also $|L(G)| = \infty$

Endlichkeitsproblem (2)

...

Wir zeigen zunächst:
Es gilt $|L(G)| = \infty$ g.d.w. es ein Wort $z \in L(G)$ mit $n \leq |z| < 2n$ gibt.

„$\Rightarrow$“:

- Beweis durch Widerspruch
- Annahme:
  Es gibt kein Wort $z \in L(G)$ für $n \leq |z| < 2n$, aber trotzdem gilt $|L(G)| = \infty$.
- Sei $z \in L(G)$ das kürzeste Wort mit $|z| \geq 2n$.
- Pumping-Lemma: Es gibt u, v, w, x, y gibt mit $z = uvwxy$, $|vx| > 0$ und $|vwx| \leq n$, sodass insbes. $uv^0wx^0y \in L$ gilt.
- Da $|uv^0wx^0y| = |uwy| < |uvwxy|$ und $|uwy| \geq n$ gilt, war z nicht minimal gewählt. Widerspruch!

Endlichkeitsproblem (3)

Entscheide Endlichkeitsproblem:

- Teste für alle Worte $w \in \Sigma^*$,
  der Länge $n \leq |w| < 2n$,
  ob $w \in L(G)$ gilt (mit CYK-Algorithmus).
- Wenn $w \in L(G)$, dann $|L(G)| = \infty$, sonst $|L(G)| < \infty$.

Weiteres Entscheidbarkeitsproblem

Das Problem, ob eine deterministisch kontextfreie Sprache äquivalent zu einer regulären Sprache ist, ist entscheidbar.

- Sei L_1 durch DPDA gegeben und L_2 durch einen DFA.
- Prüfe $\overline{L_1} \cap L_2 = \emptyset$ und $L_1 \cap \overline{L_2} = \emptyset$
- Beides ist entscheidbar, da DPDAs und DFAs abgeschlossen unter Komplementbildung, Schnittbildung zwischen DPDA und DFA durch DPDA konstruierbar ist und Leerheitsproblem für CFLs entscheidbar ist
- $\overline{L_i} \cap L_j = \emptyset$ impliziert $L_j \subseteq L_i$
- Daher ist $\bigwedge_{(i,j)\in\{(1,2),(2,1)\}} \overline{L_i} \cap L_j = \emptyset$ äquivalent zu $L_1 = L_2$.

Kontextsensitive Sprachen

Ziel: Beweis des Satz von Kuroda (nächste Vorlesung):
Kontextsensitive Sprachen werden genau von den LBAs erkannt.

Wiederholung: Kontextsensitive Grammatik (V, Σ, P, S) erfordert
wobei für alle $\ell \to r \in P$: $|\ell| \leq |r|$

Wie bei CFGs, gibt es auch Normalformen für kontextsensitive Sprachen:
die Kuroda-Normalform

Kuroda-Normalform für Typ 1-Sprachen

(benannt nach dem japanischen Linguisten Sige-Yuki Kuroda)

Definition

Eine Typ 1-Grammatik $G = (V, \Sigma, P, S)$ ist in Kuroda-Normalform, falls alle Produktionen in P einer der folgenden vier Formen entsprechen:

$$A \to a \qquad A \to B \qquad A \to BC \qquad AB \to CD$$

wobei $a \in \Sigma$ und $A, B, C, D \in V$.

Bemerkung: Die Kuroda-Normalform „erweitert“ kontextfreie Grammatiken um Regeln der Form $AB \to CD$.

Herstellen der Kuroda-Normalform

Satz

Sei L eine kontextsensitive Sprache mit $\varepsilon \notin L$.
Dann gibt es eine Grammatik in Kuroda-Normalform, die L erzeugt.

Beweis: Algorithmus 10 (nächste Folie) bewerkstelligt dies.

Algorithmus 10: Herstellung der Kuroda-Normalform

Eingabe: Eine Typ 1-Grammatik $G = (V, \Sigma, P, S)$ mit $\varepsilon \notin L(G)$
Ausgabe: Eine Typ 1-Grammatik in Kuroda-Normalform die $L(G)$ erzeugt

Entfernt alle $a \in \Sigma$ aus den Regeln bis auf neue $A \to a$-Regeln

Beginn

für alle $a \in \Sigma$ **tue**

/* Führe neue Variable A_a für a ein, und ersetze Vorkommen von a durch das Nichtterminal */

$G := (V \dot\cup \{A_a\}, \Sigma, \{\ell[A_a/a] \to r[A_a/a] \mid \ell \to r \in P\}, S);$

/* Nun sind alle Regeln von der Form $A \to a$ oder $A_1 \cdots A_m \to B_1 \cdots B_n$ mit $A_i, B_j \in V$ */

für alle $A \to B_1 \cdots B_n \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A \to B_1 \cdots B_n$ durch neue Regeln */ **tue**

Seien $C_1, \ldots, C_{n-2}$ neue Variablen;
$V := V \dot\cup \{C_1, \ldots, C_{n-2}\};$
$P := (P \setminus \{A \to B_1 \cdots B_n\})$
$\cup \{A \to B_1 C_1\} \cup \{C_i \to B_{i+1} C_{i+1} \mid i = 1, \ldots, n-3\} \cup \{C_{n-2} \to B_{n-1} B_n\};$

für alle $A_1 \cdots A_m \to B_1 \cdots B_n \in P$ *mit* $(m > 2$ *oder* $n > 2)$ *und* $n \geq m + 2$ /* Ersetze $A_1 \cdots A_m \to B_1 \cdots B_n$ durch neue Regeln */ **tue**

Seien $D_2, \ldots, D_{n-1}$ neue Variablen;
$V := V \dot\cup \{D_2, \ldots, D_{n-1}\};$
$P := (P \setminus \{A_1 \cdots A_m \to B_1 \cdots B_n\}) \cup \{A_1 A_2 \to B_1 D_2\} \cup \{D_i A_{i+1} \to B_i D_{i+1} \mid i = 2, \ldots, m-1\}$
$\cup \{D_i \to B_i D_{i+1} \mid i = m, \ldots, n-2\} \cup \{D_{n-1} \to B_{n-1} B_n\}$

für alle $A_1 \cdots A_n \to B_1 \cdots B_{n+1} \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A_1 \cdots A_n \to B_1 \cdots B_{n+1}$ durch neue Regeln */ **tue**

Seien $D_2, \ldots, D_n$ neue Variablen;
$V := V \dot\cup \{D_2, \ldots, D_{n-1}\};$
$P := (P \setminus \{A_1 \cdots A_n \to B_1 \cdots B_n\}) \cup \{A_1 A_2 \to B_1 D_2\} \cup \{D_i A_{i+1} \to B_i D_{i+1} \mid i = 2, \ldots, n-1\} \cup \{D_n \to B_n B_{n+1}\}$

für alle $A_1 \cdots A_n \to B_1 \cdots B_n \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A_1 \cdots A_n \to B_1 \cdots B_n$ durch neue Regeln */ **tue**

Seien $D_2, \ldots, D_{n-1}$ neue Variablen;
$V := V \dot\cup \{D_2, \ldots, D_{n-1}\};$
$P := (P \setminus \{A_1 \cdots A_n \to B_1 \cdots B_n\}) \cup \{A_1 A_2 \to B_1 D_2\} \cup \{D_i A_{i+1} \to B_i D_{i+1} \mid i = 2, \ldots, n-2\} \cup \{D_{n-1} A_n \to B_{n-1} B_n\}$

Gebe die so entstandene Grammatik aus;

Algorithmus 10: Herstellung der Kuroda-Normalform

Eingabe: Eine Typ 1-Grammatik $G = (V, \Sigma, P, S)$ mit $\varepsilon \notin L(G)$
Ausgabe: Eine Typ 1-Grammatik in Kuroda-Normalform die $L(G)$ erzeugt
Beginn

Entfernt alle $a \in \Sigma$ aus den Regeln bis auf neue $A \to a$-Regeln

für alle $a \in \Sigma$ **tue**

/* Führe neue Variable A_a für a ein, und ersetze Vorkommen von a durch das Nichtterminal */

$G := (V \dot\cup \{A_a\}, \Sigma, \{\ell[A_a/a] \to r[A_a/a] \mid \ell \to r \in P\}, S)$;

"Zerhacken" von rechten Seiten wie bei Chomsky-NF

/* Nun sind alle Regeln von der Form $A \to a$ oder $A_1 \cdots A_m \to B_1 \cdots$ */

für alle $A \to B_1 \cdots B_n \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A \to B_1 \cdots B_n$ durch neue Regeln */ **tue**

Seien $C_1, \ldots, C_{n-2}$ neue Variablen;
$V := V \dot\cup \{C_1, \ldots, C_{n-2}\}$;
$P := (P \setminus \{A \to B_1 \cdots B_n\})$
$\quad \cup \{A \to B_1 C_1\} \cup \{C_i \to B_{i+1} C_{i+1} \mid i = 1, \ldots, n-3\} \cup \{C_{n-2} \to B_{n-1} B_n\}$;

für alle $A_1 \cdots A_m \to B_1 \cdots B_n \in P$ *mit* ($m > 2$ *oder* $n > 2$) *und* $n \geq m + 2$ /* Ersetze $A_1 \cdots A_m \to B_1 \cdots B_n$ durch neue Regeln */ **tue**

Seien $D_2, \ldots, D_{n-1}$ neue Variablen;
$V := V \dot\cup \{D_2, \ldots, D_{n-1}\}$;
$P := (P \setminus \{A_1 \cdots A_m \to B_1 \cdots B_n\}) \cup \{A_1 A_2 \to B_1 D_2\} \cup \{D_i A_{i+1} \to B_i D_{i+1} \mid i = 2, \ldots, m-1\}$
$\quad \cup \{D_i \to B_i D_{i+1} \mid i = m, \ldots, n-2\} \cup \{D_{n-1} \to B_{n-1} B_n\}$

für alle $A_1 \cdots A_n \to B_1 \cdots B_{n+1} \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A_1 \cdots A_n \to B_1 \cdots B_{n+1}$ durch neue Regeln */ **tue**

Seien $D_2, \ldots, D_n$ neue Variablen;
$V := V \dot\cup \{D_2, \ldots, D_{n-1}\}$;
$P := (P \setminus \{A_1 \cdots A_n \to B_1 \cdots B_n\}) \cup \{A_1 A_2 \to B_1 D_2\} \cup \{D_i A_{i+1} \to B_i D_{i+1} \mid i = 2, \ldots, n-1\} \cup \{D_n \to B_n B_{n+1}\}$

für alle $A_1 \cdots A_n \to B_1 \cdots B_n \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A_1 \cdots A_n \to B_1 \cdots B_n$ durch neue Regeln */ **tue**

Seien $D_2, \ldots, D_{n-1}$ neue Variablen;
$V := V \dot\cup \{D_2, \ldots, D_{n-1}\}$;
$P := (P \setminus \{A_1 \cdots A_n \to B_1 \cdots B_n\}) \cup \{A_1 A_2 \to B_1 D_2\} \cup \{D_i A_{i+1} \to B_i D_{i+1} \mid i = 2, \ldots, n-2\} \cup \{D_{n-1} A_n \to B_{n-1} B_n\}$

Gebe die so entstandene Grammatik aus;

Algorithmus 10: Herstellung der Kuroda-Normalform

Eingabe: Eine Typ 1-Grammatik $G = (V, \Sigma, P, S)$ mit $\varepsilon \notin L(G)$
Ausgabe: Eine Typ 1-Grammatik in Kuroda-Normalform die $L(G)$ erzeugt
Beginn

Entfernt alle $a \in \Sigma$ aus den Regeln bis auf neue $A \to a$-Regeln

für alle $a \in \Sigma$ **tue**
  /* Führe neue Variable A_a für a ein, und ersetze Vorkommen von a durch das Nichtterminal */
  $G := (V \dot\cup \{A_a\}, \Sigma, \{\ell[A_a/a] \to r[A_a/a] \mid \ell \to r \in P\}, S)$;

/* Nun sind alle Regeln von der Form $A \to a$ oder $A_1 \cdots A_m \to B_1 \cdots$ */

"Zerhacken" von rechten Seiten wie bei Chomsky-NF

für alle $A \to B_1 \cdots B_n \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion $A \to B_1 \cdots B_n$ durch neue Regeln */ **tue**
  Seien $C_1, \ldots, C_{n-2}$ neue Variablen;
  $V := V \dot\cup \{C_1, \ldots, C_{n-2}\}$;
  $P := (P \setminus \{A \to B_1 \cdots B_n\})$
    $\cup \{A \to B_1C_1\} \cup \{C_i \to B_{i+1}C_{i+1} \mid i = 1, \ldots, n-3\} \cup \{C_{n-2} \to B_{n-1}B_n\}$;

für alle $A_1 \cdots A_m \to B_1 \cdots B_n \in P$ *mit* ($m > 2$ *oder* $n > 2$) *und* $n \geq m + 2$ /* Ersetze A … **tue**
  Seien $D_2, \ldots, D_{n-1}$ neue Variablen;
  $V := V \dot\cup \{D_2, \ldots, D_{n-1}\}$;
  $P := (P \setminus \{A_1 \cdots A_m \to B_1 \cdots B_n\}) \cup \{A_1A_2 \to B_1D_2\} \cup \{D_iA_{i+1} \to B_iD_{i+1} \mid i = 2$ …
    $\cup \{D_i \to B_iD_{i+1} \mid i = m, \ldots, n-2\} \cup \{D_{n-1} \to B_{n-1}B_n\}$

für alle $A_1 \cdots A_n \to B_1 \cdots B_{n+1} \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion A_1 … **tue**
  Seien $D_2, \ldots, D_n$ neue Variablen;
  $V := V \dot\cup \{D_2, \ldots, D_{n-1}\}$;
  $P := (P \setminus \{A_1 \cdots A_n \to B_1 \cdots B_n\}) \cup \{A_1A_2 \to B_1D_2\} \cup \{D_iA_{i+1} \to B_iD_{i+1} \mid i = 2$ …

für alle $A_1 \cdots A_n \to B_1 \cdots B_n \in P$ *mit* $n > 2$ /* Ersetze in P die Produktion A … **tue**
  Seien $D_2, \ldots, D_{n-1}$ neue Variablen;
  $V := V \dot\cup \{D_2, \ldots, D_{n-1}\}$;
  $P := (P \setminus \{A_1 \cdots A_n \to B_1 \cdots B_n\}) \cup \{A_1A_2 \to B_1D_2\} \cup \{D_iA_{i+1} \to B_iD_{i+1} \mid i = 2$ …

Gebe die so entstandene Grammatik aus;

Effekt:
Ableitung vorher:
$xA_1 \cdots A_m y$
$\Rightarrow xB_1 \cdots B_n y$
Ableitung nachher:
$xA_1A_2 \cdots A_m y$
$\Rightarrow xB_1D_2A_3 \cdots A_m y$
$\Rightarrow xB_1B_2D_3A_4 \cdots A_m y$
$\Rightarrow \ldots$
$\Rightarrow xB_1 \cdots B_{m-2}D_{m-1}A_m y$
$\Rightarrow xB_1 \cdots B_{m-1}D_m y$
$\Rightarrow xB_1 \cdots B_{m-1}B_mD_{m+1} y$
$\Rightarrow \ldots$
$\Rightarrow xB_1 \cdots B_{n-2}D_{n-1} y$
$\Rightarrow xB_1 \cdots B_{n-2}B_{n-1}B_n y$

Beispiel

Sei $G = (\{S, A, B\}, \{a, b\}, P, S)$ mit $P = \{S \to AS \mid BS \mid a,$

$$\begin{aligned}
& ABAA \to AAAB, \\
& ABAB \to AABB, \\
& BAA \to AAB, \\
& BAB \to ABB, \\
& BBA \to ABB, \\
& AA \to aa, \\
& BB \to bb\}
\end{aligned}$$

Beispiel (2)

Schritt 1: a, b durch neue Nichtterminale sharen ergibt:

$$
\begin{aligned}
V &= \{S, A, B, A_a, A_b\} \\
P &= \{S \to AS \mid BS \mid A_a, \\
&\qquad ABAA \to AAAB, \\
&\qquad ABAB \to AABB, \\
&\qquad BAA \to AAB, \\
&\qquad BAB \to ABB, \\
&\qquad BBA \to ABB, \\
&\qquad AA \to A_aA_a, \\
&\qquad BB \to A_bA_b. \\
&\qquad A_a \to a, \\
&\qquad A_b \to b\}
\end{aligned}
$$

Beispiel (3)

Regeln $A \to B_1, \ldots, B_m$ mit $m > 2$ gibt es nicht.

Regeln $A_1 \ldots A_m \to B_1, \ldots, B_n$ mit $m > 2$ oder $n > 2$, werden ersetzt:

- $ABAA \to AAAB$ wird ersetzt durch $AB \to AD_2$, $D_2A \to AD_3$, $D_3A \to AB$
- $ABAB \to AABB$ wird ersetzt durch $AB \to AD_4, D_4A \to AD_5, D_5B \to BB$
- $BAA \to AAB$ wird ersetzt durch $BA \to AD_6, D_6A \to AB$
- $BAB \to ABB$ wird ersetzt durch $BA \to AD_7, D_7B \to BB$
- $BBA \to ABB$ wird ersetzt durch $BB \to AD_8, D_8A \to BB$

Beispiel (4)

Grammatik in Kuroda-Normalform $G = (V, \{a, b\}, P, S)$ mit

$$
\begin{aligned}
V &= \{S, A, B, A_a, A_b, D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8\} \\
P &= \{S \to AS \mid BS \mid A_a, \\
&\qquad AA \to A_a A_a, BB \to A_b A_b. \\
&\qquad A_a \to a, A_b \to b, \\
&\qquad AB \to AD_2, D_2 A \to AD_3 \\
&\qquad D_3 A \to AB, AB \to AD_4, \\
&\qquad D_4 A \to AD_5, D_5 B \to BB, \\
&\qquad BA \to AD_6, D_6 A \to AB, \\
&\qquad BA \to AD_7, D_7 B \to BB, \\
&\qquad BB \to AD_8, D_8 A \to BB\}
\end{aligned}
$$