

## 8a

**Linear beschränkte Turingmaschinen (Fortsetzung)**

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für  
Theoretische Informatik und TheorembeweisenStand: 11. Juni 2024  
Basierend auf Folien von PD Dr. David Sabel

# Wiederholung: Kontextsensitive Grammatiken

---

- ▶ Die Linke Seite von Produktionen ist  $\ell \in (V \cup \Sigma)^+$ .
- ▶ Es gilt  $|\ell| \leq |r|$  für alle Produktionen  $\ell \rightarrow r$ .

## Definition

Eine **linear beschränkte Turingmaschine** (*linear bounded automaton*, **LBA**) ist ein 8-Tupel  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$ , wobei:

- ▶  $Z, \Sigma, \Gamma, \delta, z_0$  und  $E$  sind wie bei nichtdeterministischen Turingmaschinen
- ▶  $\langle, \rangle \in \Sigma$  sind **Start- bzw. Endmarker**
- ▶  $\delta$  überschreibt keinen der Marker
- ▶ bei  $\langle$  gibt  $\delta$  nie  $L$  aus und bei  $\rangle$  gibt  $\delta$  nie  $R$  aus.

# Der Satz von Kuroda

---

Unser Ziel ist es, den Satz von Kuroda zu beweisen:

## **Theorem (Satz von Kuroda)**

Die LBAs akzeptieren genau die kontextsensitiven Sprachen.

Unser Ziel ist es, den Satz von Kuroda zu beweisen:

## **Theorem (Satz von Kuroda)**

Die LBAs akzeptieren genau die kontextsensitiven Sprachen.

Der Beweis erfolgt in zwei Teilen:

- ▶ Jede kontextsensitive Sprache wird von einem LBA erkannt.
- ▶ Die akzeptierte Sprache von LBAs ist kontextsensitiv.

# Kuroda-Normalform für Typ 1-Sprachen

---

Wie bei kontextfreien Sprachen gibt es Normalformen auch für kontextsensitive Sprachen: [die Kuroda-Normalform](#).

Sie ist nach dem Linguisten Sige-Yuki Kuroda benannt.

# Kuroda-Normalform für Typ 1-Sprachen

Wie bei kontextfreien Sprachen gibt es Normalformen auch für kontextsensitive Sprachen: [die Kuroda-Normalform](#).

Sie ist nach dem Linguisten Sige-Yuki Kuroda benannt.

## 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:

$$A \rightarrow a \quad A \rightarrow B \quad A \rightarrow BC \quad AB \rightarrow CD$$

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

# Kuroda-Normalform für Typ 1-Sprachen

Wie bei kontextfreien Sprachen gibt es Normalformen auch für kontextsensitive Sprachen: [die Kuroda-Normalform](#).

Sie ist nach dem Linguisten Sige-Yuki Kuroda benannt.

## 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:

$$A \rightarrow a \quad A \rightarrow B \quad A \rightarrow BC \quad AB \rightarrow CD$$

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

Die Kuroda-Normalform erweitert kontextfreie Grammatiken um Regeln von der Form  $AB \rightarrow CD$ .



# 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.

# 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 (siehe Skript) bewerkstelligt dies.



# Kontextsensitive Sprachen werden von LBAs akzeptiert

---

## Satz

Für jede kontextsensitive Sprache  $L$  gibt es einen LBA  $M$  mit  $L(M) = L$ .

# Kontextsensitive Sprachen werden von LBAs akzeptiert

## Satz

Für jede kontextsensitive Sprache  $L$  gibt es einen LBA  $M$  mit  $L(M) = L$ .

**Beweis** Sei  $G = (V, \Sigma, P, S)$  mit  $L(G) = L \setminus \{\varepsilon\}$  eine kontextsensitive Grammatik in Kuroda-Normalform.

# Kontextsensitive Sprachen werden von LBAs akzeptiert

## Satz

Für jede kontextsensitive Sprache  $L$  gibt es einen LBA  $M$  mit  $L(M) = L$ .

**Beweis** Sei  $G = (V, \Sigma, P, S)$  mit  $L(G) = L \setminus \{\varepsilon\}$  eine kontextsensitive Grammatik in Kuroda-Normalform.

Grundgedanke:

- ▶ Konstruiere den LBA  $M$  mit Bandalphabet  $\Gamma := \Sigma \cup V \cup \{\langle, \rangle, x\}$ .
- ▶  $M$  versucht nichtdeterministisch für  $w \in \Sigma^*$  das Startsymbol  $S$  der Grammatik rückwärts herzuleiten, durch rückwärts Anwenden der Produktionen  $\ell \rightarrow r \in P$ , wobei Vorkommen von  $r$  durch  $\ell$  ersetzt werden.
- ▶ Akzeptiere, wenn  $\langle S \rangle$  (bzw.  $\langle \rangle$ , falls  $S \rightarrow \varepsilon \in P$ ) auf dem Band vorkommt.
- ▶ Nichtdeterminismus wird verwendet, um zu beschließen, welche Produktion wird rückwärts angewendet und für welches Vorkommen einer rechten Seite.

## Beispiel für einen Lauf des konstruierten LBA

---

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  
 $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

# Beispiel für einen Lauf des konstruierten LBA

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

Akzeptierender Lauf von  $M$ :

1. Am Anfang steht  $\langle ab \rangle$  auf dem Band.

## Beispiel für einen Lauf des konstruierten LBA

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

Akzeptierender Lauf von  $M$ :

1. Am Anfang steht  $\langle ab \rangle$  auf dem Band.
2.  $b$  wird durch  $B$  ersetzt. Auf dem Band steht nun  $\langle aB \rangle$ .



## Beispiel für einen Lauf des konstruierten LBA

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

Akzeptierender Lauf von  $M$ :

1. Am Anfang steht  $\langle ab \rangle$  auf dem Band.
2.  $b$  wird durch  $B$  ersetzt. Auf dem Band steht nun  $\langle aB \rangle$ .
3.  $a$  wird durch  $A$  ersetzt. Auf dem Band steht nun  $\langle AB \rangle$ .

## Beispiel für einen Lauf des konstruierten LBA

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

Akzeptierender Lauf von  $M$ :

1. Am Anfang steht  $\langle ab \rangle$  auf dem Band.
2.  $b$  wird durch  $B$  ersetzt. Auf dem Band steht nun  $\langle aB \rangle$ .
3.  $a$  wird durch  $A$  ersetzt. Auf dem Band steht nun  $\langle AB \rangle$ .
4.  $AB$  wird durch  $xS$  ersetzt. Auf dem Band steht nun  $\langle xS \rangle$ .

## Beispiel für einen Lauf des konstruierten LBA

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

Akzeptierender Lauf von  $M$ :

1. Am Anfang steht  $\langle ab \rangle$  auf dem Band.
2.  $b$  wird durch  $B$  ersetzt. Auf dem Band steht nun  $\langle aB \rangle$ .
3.  $a$  wird durch  $A$  ersetzt. Auf dem Band steht nun  $\langle AB \rangle$ .
4.  $AB$  wird durch  $xS$  ersetzt. Auf dem Band steht nun  $\langle xS \rangle$ .
5. Das  $x$  wird durch  $\langle$  ersetzt. Auf dem Band steht  $\langle \langle S \rangle$ .

## Beispiel für einen Lauf des konstruierten LBA

Grammatik  $G = (\{S, A, B\}, \{a, b\}, P, S)$  in Kuroda-Normalform mit  $P = \{S \rightarrow AB, S \rightarrow A, A \rightarrow a, B \rightarrow b\}$ .

Eingabewort:  $ab$

Akzeptierender Lauf von  $M$ :

1. Am Anfang steht  $\langle ab \rangle$  auf dem Band.
2.  $b$  wird durch  $B$  ersetzt. Auf dem Band steht nun  $\langle aB \rangle$ .
3.  $a$  wird durch  $A$  ersetzt. Auf dem Band steht nun  $\langle AB \rangle$ .
4.  $AB$  wird durch  $xS$  ersetzt. Auf dem Band steht nun  $\langle xS \rangle$ .
5. Das  $x$  wird durch  $\langle$  ersetzt. Auf dem Band steht  $\langle \langle S \rangle$ .
6.  $M$  akzeptiert.

# Kontextsensitive Sprachen werden von LBAs akzeptiert

---

**Beweis** (Fortsetzung) Suche nach einer rechter Seite  $r$ :

1. Beginne links an der Eingabe und laufe diese durch.
2. Speichere das Symbol links vom Schreib-Lesekopf im aktuellen Zustand.
3. Stelle mit dem aktuellen Symbol fest, ob es passende Produktion gibt.  
(Da rechte Seiten von Produktionen in Kuroda-Normalform aus maximal zwei Zeichen bestehen, genügt dies.)

# Kontextsensitive Sprachen werden von LBAs akzeptiert

---

**Beweis** (Fortsetzung) Ersetzung von  $r$  durch  $\ell$ :

1. Für  $A \rightarrow a$  und  $A \rightarrow B$  wird das aktuelle Symbol durch  $A$  ersetzt, anschließend wird der nächste Schritt gestartet.
2. Für  $AB \rightarrow CD$ , wird  $B$  geschrieben und der Kopf nach links wechseln, dann  $A$  geschrieben und der nächste Schritt gestartet.
3. Für  $A \rightarrow BC$  schreibe  $A$  und wechsele nach links, schreibe  $x$ , fahre ganz nach links und starte die Prozedur zum Verschieben der Zeichen nach rechts, solange bis  $x$  überschrieben ist.

# Kontextsensitive Sprachen werden von LBAs akzeptiert

---

**Beweis** (Fortsetzung) Verschieben nach rechts:

1. Speichere das linkeste Symbol im Zustand und schreibe  $\langle$ .
2. Laufe nach rechts. Dabei wird das aktuelle Symbol mit dem gespeicherten vertauscht.
3. Beende Vertauschen, nachdem  $x$  mit einem anderen Symbol vertauscht wird.  $\square$

# Bemerkungen und Typ 0-Grammatiken

---

- ▶ Der Grundgedanke der Konstruktion funktioniert auch für Typ 1-Grammatiken nicht in Kuroda-Normalform, ist aber komplizierter.



# Bemerkungen und Typ 0-Grammatiken

---

- ▶ Der Grundgedanke der Konstruktion funktioniert auch für Typ 1-Grammatiken nicht in Kuroda-Normalform, ist aber komplizierter.
- ▶ Der Grundgedanke der Konstruktion funktioniert auch für Typ 0-Grammatiken: Der Platz ist allerdings dann unbeschränkt, und eine NTM wird benötigt.

# Bemerkungen und Typ 0-Grammatiken

- ▶ Der Grundgedanke der Konstruktion funktioniert auch für Typ 1-Grammatiken nicht in Kuroda-Normalform, ist aber komplizierter.
- ▶ Der Grundgedanke der Konstruktion funktioniert auch für Typ 0-Grammatiken: Der Platz ist allerdings dann unbeschränkt, und eine NTM wird benötigt.

## Satz

Für jede Typ 0-Sprache  $L$  gibt es eine nichtdeterministischen Turingmaschine  $M$  mit  $L(M) = L$ .

**Beweis** Der Beweis des letzten Satzes kann angepasst werden, damit er eine NTM konstruiert, die eine beliebige Typ 0-Grammatik simuliert. □

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

---

## Satz

Sei  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  ein LBA. Dann ist  $L(M)$  kontextsensitiv.

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Satz

Sei  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  ein LBA. Dann ist  $L(M)$  kontextsensitiv.

**Beweis** Konstruiere eine Typ 1-Grammatik  $G$  mit  $L(G) = L(M)$ .

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Satz

Sei  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  ein LBA. Dann ist  $L(M)$  kontextsensitiv.

**Beweis** Konstruiere eine Typ 1-Grammatik  $G$  mit  $L(G) = L(M)$ .

Grundgedanke:

1. Erzeuge ein beliebiges Wort  $w$  und die Startkonfiguration von  $M$  für  $w$ .
2. Simuliere  $M$  zum Prüfen, ob  $w \in L(M)$ .
3. Wenn  $M$  akzeptiert, erzeuge  $w$  endgültig.

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Satz

Sei  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  ein LBA. Dann ist  $L(M)$  kontextsensitiv.

**Beweis** Konstruiere eine Typ 1-Grammatik  $G$  mit  $L(G) = L(M)$ .

Grundgedanke:

1. Erzeuge ein beliebiges Wort  $w$  und die Startkonfiguration von  $M$  für  $w$ .
2. Simuliere  $M$  zum Prüfen, ob  $w \in L(M)$ .
3. Wenn  $M$  akzeptiert, erzeuge  $w$  endgültig.

Die Variablen  $V$  von  $G$  sind  $S$ ,  $A$  und Tupeln von der Form  $\begin{bmatrix} u \\ v \end{bmatrix}$ , wobei  $u \in \Sigma$  und  $v \in (\Gamma \cup Z)^+$ .

Die oberen Komponenten einer Tupelfolge  $\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \cdots \begin{bmatrix} u_n \\ v_n \end{bmatrix}$  ergeben ein Wort  $w$ .  
Die unteren Komponenten ergeben eine LBA-Konfiguration.

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$z_0 \langle aa \rangle$



## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$z_0 \langle aa \rangle$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle \vdash \langle x x z_2 \rangle$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle xz_1 a \rangle \vdash \langle xxz_2 \rangle$$

Ableitung von  $aa$ :

$S$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit  
 $Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle xz_1 a \rangle \vdash \langle xxz_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \left[ \begin{smallmatrix} a \\ z_0 \langle a \end{smallmatrix} \right] A$$



## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit  
 $Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle xz_1 a \rangle \vdash \langle xxz_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix}$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit  
 $Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle \vdash \langle x x z_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle z_0 a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix}$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle \vdash \langle x x z_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle z_0 a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ z_1 a \rangle \end{bmatrix}$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle \vdash \langle x x z_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle z_0 a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ z_1 a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ x z_2 \rangle \end{bmatrix}$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit

$Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle \vdash \langle x x z_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle z_0 a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ z_1 a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ x z_2 \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} a$$

## Beispiel für eine Ableitung der konstruierten Grammatik

LBA  $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$  mit  
 $Z = \{z_0, z_1, z_2\}$ ,  $\Sigma = \{a, \langle, \rangle\}$ ,  $\Gamma = \Sigma \cup \{x\}$ ,  $E = \{z_2\}$  und

$$\begin{array}{lll} \delta(z_0, a) = (z_1, x, R) & \delta(z_0, \langle) = (z_0, \langle, R) & \delta(z_0, \rangle) = (z_0, \rangle, L) \\ \delta(z_1, a) = (z_2, x, R) & \delta(z_1, \langle) = (z_1, \langle, R) & \delta(z_1, \rangle) = (z_1, \rangle, L) \end{array}$$

Akzeptierender Lauf auf  $aa$ :

$$z_0 \langle aa \rangle \vdash \langle z_0 aa \rangle \vdash \langle x z_1 a \rangle \vdash \langle x x z_2 \rangle$$

Ableitung von  $aa$ :

$$S \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \Rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle z_0 a \rangle \end{bmatrix} \begin{bmatrix} a \\ a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ z_1 a \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} \begin{bmatrix} a \\ x z_2 \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ \langle x \rangle \end{bmatrix} a \Rightarrow aa$$

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

**Beweis** (Fortsetzung) Konstruiere  $G = (V, \Sigma - \{\langle, \rangle\}, P_0 \cup P_1 \cup P_2 \cup P_3, S)$ .

1. Erzeuge ein beliebiges Wort  $w \in \Sigma^*$  und die Startkonfiguration von  $M$  für  $w$ :

► Regeln zur Erzeugung von  $w \in \Sigma^*$  und Startkonfiguration zw:

$$P_0 := \{S \rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \mid a \in \Sigma\} \cup \{S \rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \mid a \in \Sigma\} \\ \cup \{A \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} A \mid a \in \Sigma\} \cup \{A \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \mid a \in \Sigma\}$$

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

**Beweis** (Fortsetzung) Konstruiere  $G = (V, \Sigma - \{\langle, \rangle\}, P_0 \cup P_1 \cup P_2 \cup P_3, S)$ .

1. Erzeuge ein beliebiges Wort  $w \in \Sigma^*$  und die Startkonfiguration von  $M$  für  $w$ :

► Regeln zur Erzeugung von  $w \in \Sigma^*$  und Startkonfiguration zw:

$$P_0 := \{S \rightarrow \begin{bmatrix} a \\ z_0 \langle a \end{bmatrix} A \mid a \in \Sigma\} \cup \{S \rightarrow \begin{bmatrix} a \\ z_0 \langle a \end{bmatrix} \mid a \in \Sigma\} \\ \cup \{A \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} A \mid a \in \Sigma\} \cup \{A \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \mid a \in \Sigma\}$$

► Für  $a_1 \cdots a_n \in \Sigma^*$  mit  $n \geq 1$  gilt

$$S \Rightarrow^* \begin{bmatrix} a_1 \\ z_0 \langle a_1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ a_n \end{bmatrix}$$



# Die akzeptierte Sprache von LBAs ist kontextsensitiv

**Beweis** (Fortsetzung) Konstruiere  $G = (V, \Sigma - \{\langle, \rangle\}, P_0 \cup P_1 \cup P_2 \cup P_3, S)$ .

1. Erzeuge ein beliebiges Wort  $w \in \Sigma^*$  und die Startkonfiguration von  $M$  für  $w$ :

► Regeln zur Erzeugung von  $w \in \Sigma^*$  und Startkonfiguration zw:

$$P_0 := \{S \rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} A \mid a \in \Sigma\} \cup \{S \rightarrow \begin{bmatrix} a \\ z_0 \langle a \rangle \end{bmatrix} \mid a \in \Sigma\} \\ \cup \{A \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} A \mid a \in \Sigma\} \cup \{A \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \mid a \in \Sigma\}$$

► Für  $a_1 \cdots a_n \in \Sigma^*$  mit  $n \geq 1$  gilt

$$S \Rightarrow^* \begin{bmatrix} a_1 \\ z_0 \langle a_1 \rangle \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ a_n \end{bmatrix}$$

►  $\varepsilon$  kann dadurch nicht erzeugt werden, daher:

$$P_1 := \{S \rightarrow \varepsilon \mid \varepsilon \in L(M)\}$$

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Beweis (Fortsetzung)

2. Simuliere  $M$  zum Prüfen, ob  $w \in L(M)$ .

- Regelmenge  $P_2$  simuliert  $M$  auf den unteren Komponenten:

$$\begin{aligned} P_2 := & \{ \begin{bmatrix} d \\ z a \end{bmatrix} \rightarrow \begin{bmatrix} d \\ z' b \end{bmatrix} \mid (z', b, N) \in \delta(z, a), d \in \Sigma \} \\ & \cup \{ \begin{bmatrix} d \\ c \end{bmatrix} \begin{bmatrix} e \\ z a \end{bmatrix} \rightarrow \begin{bmatrix} d \\ z' c \end{bmatrix} \begin{bmatrix} e \\ b \end{bmatrix} \mid (z', b, L) \in \delta(z, a), c \in \Gamma, d, e \in \Sigma \} \\ & \cup \{ \begin{bmatrix} d \\ z a \end{bmatrix} \begin{bmatrix} e \\ c \end{bmatrix} \rightarrow \begin{bmatrix} d \\ b \end{bmatrix} \begin{bmatrix} e \\ z' c \end{bmatrix} \mid (z', b, R) \in \delta(z, a), c \in \Gamma, d, e \in \Sigma \} \\ & \cup \dots \end{aligned}$$

- Weitere Regeln sind notwendig, um mit  $\langle$  und  $\rangle$  richtig umzugehen.

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Beweis (Fortsetzung)

3. Wenn  $M$  akzeptiert, erzeuge  $w$  endgültig.

- ▶ Nach Akzeptieren durch  $M$  erstellt Regelmenge  $P_3$  aus Tupelfolgen das Wort  $a_1 \cdots a_n$ .

$$P_3 := \left\{ \begin{bmatrix} b \\ z a \end{bmatrix} \rightarrow b \mid z \in E, a \in \Gamma, b \in \Sigma \right\} \\ \cup \left\{ \begin{bmatrix} b \\ a \end{bmatrix} \rightarrow b \mid a \in \Gamma, b \in \Sigma \right\} \\ \cup \dots$$

- ▶ Weitere Regeln sind notwendig um mit  $\langle$  und  $\rangle$  richtig umzugehen.

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Beweis (Fortsetzung)

3. Wenn  $M$  akzeptiert, erzeuge  $w$  endgültig.

- ▶ Nach Akzeptieren durch  $M$  erstellt Regelmenge  $P_3$  aus Tupelfolgen das Wort  $a_1 \cdots a_n$ .

$$P_3 := \left\{ \begin{bmatrix} b \\ z a \end{bmatrix} \rightarrow b \mid z \in E, a \in \Gamma, b \in \Sigma \right\} \\ \cup \left\{ \begin{bmatrix} b \\ a \end{bmatrix} \rightarrow b \mid a \in \Gamma, b \in \Sigma \right\} \\ \cup \dots$$

- ▶ Weitere Regeln sind notwendig um mit  $\langle$  und  $\rangle$  richtig umzugehen.
- ▶ Wenn  $z \in E$ , gilt

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \cdots \begin{bmatrix} a_m \\ b_m \end{bmatrix} \begin{bmatrix} a_{m+1} \\ z b_{m+1} \end{bmatrix} \begin{bmatrix} a_{m+2} \\ b_{m+2} \end{bmatrix} \cdots \begin{bmatrix} a_n \\ b_n \end{bmatrix} \Rightarrow^* a_1 \cdots a_n$$

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Beweis (Fortsetzung)

3. Wenn  $M$  akzeptiert, erzeuge  $w$  endgültig.

- ▶ Nach Akzeptieren durch  $M$  erstellt Regelmenge  $P_3$  aus Tupelfolgen das Wort  $a_1 \cdots a_n$ .

$$P_3 := \left\{ \begin{bmatrix} b \\ z a \end{bmatrix} \rightarrow b \mid z \in E, a \in \Gamma, b \in \Sigma \right\} \\ \cup \left\{ \begin{bmatrix} b \\ a \end{bmatrix} \rightarrow b \mid a \in \Gamma, b \in \Sigma \right\} \\ \cup \dots$$

- ▶ Weitere Regeln sind notwendig um mit  $\langle$  und  $\rangle$  richtig umzugehen.
- ▶ Wenn  $z \in E$ , gilt

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \cdots \begin{bmatrix} a_m \\ b_m \end{bmatrix} \begin{bmatrix} a_{m+1} \\ z b_{m+1} \end{bmatrix} \begin{bmatrix} a_{m+2} \\ b_{m+2} \end{bmatrix} \cdots \begin{bmatrix} a_n \\ b_n \end{bmatrix} \Rightarrow^* a_1 \cdots a_n$$

Dann gilt für alle  $w \in (\Sigma - \{\langle, \rangle\})^*$ :  $S \Rightarrow_G^* w$  g.d.w.  $w \in L(M)$ .

# Die akzeptierte Sprache von LBAs ist kontextsensitiv

## Beweis (Fortsetzung)

3. Wenn  $M$  akzeptiert, erzeuge  $w$  endgültig.

- ▶ Nach Akzeptieren durch  $M$  erstellt Regelmenge  $P_3$  aus Tupelfolgen das Wort  $a_1 \cdots a_n$ .

$$P_3 := \{ \begin{bmatrix} b \\ z a \end{bmatrix} \rightarrow b \mid z \in E, a \in \Gamma, b \in \Sigma \} \\ \cup \{ \begin{bmatrix} b \\ a \end{bmatrix} \rightarrow b \mid a \in \Gamma, b \in \Sigma \} \\ \cup \dots$$

- ▶ Weitere Regeln sind notwendig um mit  $\langle$  und  $\rangle$  richtig umzugehen.
- ▶ Wenn  $z \in E$ , gilt

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \cdots \begin{bmatrix} a_m \\ b_m \end{bmatrix} \begin{bmatrix} a_{m+1} \\ z b_{m+1} \end{bmatrix} \begin{bmatrix} a_{m+2} \\ b_{m+2} \end{bmatrix} \cdots \begin{bmatrix} a_n \\ b_n \end{bmatrix} \Rightarrow^* a_1 \cdots a_n$$

Dann gilt für alle  $w \in (\Sigma - \{\langle, \rangle\})^*$ :  $S \Rightarrow_G^* w$  g.d.w.  $w \in L(M)$ .

Außerdem gilt, dass  $G$  kontextsensitiv ist, da es keine verkürzenden Regeln gibt (außer  $S \rightarrow \varepsilon$ , falls vorhanden).



# Anpassung des Beweises für Typ 0-Sprachen

Die Konstruktion der Typ 1-Grammatik aus einem LBA kann für beliebige NTMs angepasst werden.

Grundgedanke:

- ▶ Zusätzliche Tupel  $\begin{bmatrix} \$ \\ v \end{bmatrix}$  für \$ ein neues Symbol.
- ▶ Konfiguration, die länger als das Eingabewort sind, werden so dargestellt:

$$\begin{bmatrix} a_1 \\ c_1 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ c_n \end{bmatrix} \begin{bmatrix} \$ \\ c_{n+1} \end{bmatrix} \cdots \begin{bmatrix} \$ \\ z_j c_m \end{bmatrix} \begin{bmatrix} \$ \\ c_r \end{bmatrix}$$

- ▶ Regelmenge  $P_3$  enthält Regeln  $\begin{bmatrix} \$ \\ c_j \end{bmatrix} \rightarrow \epsilon$ , die nicht vom Typ 1 sind.

# Anpassung des Beweises für Typ 0-Sprachen

Die Konstruktion der Typ 1-Grammatik aus einem LBA kann für beliebige NTMs angepasst werden.

Grundgedanke:

- ▶ Zusätzliche Tupel  $\begin{bmatrix} \$ \\ v \end{bmatrix}$  für  $\$$  ein neues Symbol.
- ▶ Konfiguration, die länger als das Eingabewort sind, werden so dargestellt:

$$\begin{bmatrix} a_1 \\ c_1 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ c_n \end{bmatrix} \begin{bmatrix} \$ \\ c_{n+1} \end{bmatrix} \cdots \begin{bmatrix} \$ \\ z_j c_m \end{bmatrix} \begin{bmatrix} \$ \\ c_r \end{bmatrix}$$

- ▶ Regelmenge  $P_3$  enthält Regeln  $\begin{bmatrix} \$ \\ c_j \end{bmatrix} \rightarrow \epsilon$ , die nicht vom Typ 1 sind.

Daher:

## Satz

Die nichtdeterministischen Turingmaschinen akzeptierten genau die Typ 0-Sprachen.



## 1. LBA-Problem

Erkennen deterministische LBAs dieselben Sprachen wie nichtdeterministische LBAs?

Das Problem ist bis heute ungeklärt.

## 1. LBA-Problem

Erkennen deterministische LBAs dieselben Sprachen wie nichtdeterministische LBAs?

Das Problem ist bis heute ungeklärt.

## 2. LBA-Problem

Sind die kontextsensitiven Sprachen abgeschlossen unter Komplementbildung?

Das Problem wurde 1964 formuliert von Kuroda, 1987 gelöst von sowohl Neil Immerman als auch Róbert Szelepcsényi.

## 1. LBA-Problem

Erkennen deterministische LBAs dieselben Sprachen wie nichtdeterministische LBAs?

Das Problem ist bis heute ungeklärt.

## 2. LBA-Problem

Sind die kontextsensitiven Sprachen abgeschlossen unter Komplementbildung?

Das Problem wurde 1964 formuliert von Kuroda, 1987 gelöst von sowohl Neil Immerman als auch Róbert Szelepcsényi.

Überraschenderweise **positiv**:

## Theorem (Satz von Immerman und Szelepcsényi)

Die kontextsensitiven Sprachen sind abgeschlossen unter Komplementbildung.

## 2. LBA-Problem

### Theorem (Satz von Immerman und Szelepcsényi)

Die kontextsensitiven Sprachen sind abgeschlossen unter Komplementbildung.

Naiver Beweis:

- ▶ Sei  $G = (V, \Sigma, P, S)$  eine Typ 1-Grammatik mit  $L(G) = L$ .
- ▶ Konstruiere den LBA  $M$  für  $\bar{L} = \Sigma^* \setminus L$ .
  1. Sei  $w \in \Sigma^*$ .
  2. Prüfe, ob  $S \Rightarrow_G^* w$  gilt (vgl. Beweis des Satzes von Kuroda).
  3. Wenn ja, dann verwirf.
  4. Wenn nein, dann akzeptiere.

## 2. LBA-Problem

### Theorem (Satz von Immerman und Szelepcsényi)

Die kontextsensitiven Sprachen sind abgeschlossen unter Komplementbildung.

Naiver Beweis:

- ▶ Sei  $G = (V, \Sigma, P, S)$  eine Typ 1-Grammatik mit  $L(G) = L$ .
- ▶ Konstruiere den LBA  $M$  für  $\bar{L} = \Sigma^* \setminus L$ .
  1. Sei  $w \in \Sigma^*$ .
  2. Prüfe, ob  $S \Rightarrow_G^* w$  gilt (vgl. Beweis des Satzes von Kuroda).
  3. Wenn ja, dann verwirf.
  4. Wenn nein, dann akzeptiere.

Problem: Die Berechnung von  $S \Rightarrow_G^* w$  mit einem LBA ist **nichtdeterministisch**.  
Der Test  $S \Rightarrow_G^* w$  kann **fehlschlagen**, auch wenn  $S \Rightarrow_G^* w$  gilt.

## 2. LBA-Problem

### Theorem (Satz von Immerman und Szelepcsényi)

Die kontextsensitiven Sprachen sind abgeschlossen unter Komplementbildung.

Naiver Beweis:

- ▶ Sei  $G = (V, \Sigma, P, S)$  eine Typ 1-Grammatik mit  $L(G) = L$ .
- ▶ Konstruiere den LBA  $M$  für  $\bar{L} = \Sigma^* \setminus L$ .
  1. Sei  $w \in \Sigma^*$ .
  2. Prüfe, ob  $S \Rightarrow_G^* w$  gilt (vgl. Beweis des Satzes von Kuroda).
  3. Wenn ja, dann verwirf.
  4. Wenn nein, dann akzeptiere.

Problem: Die Berechnung von  $S \Rightarrow_G^* w$  mit einem LBA ist **nichtdeterministisch**.  
Der Test  $S \Rightarrow_G^* w$  kann **fehlschlagen**, auch wenn  $S \Rightarrow_G^* w$  gilt.

Lösung: Siehe Skript.