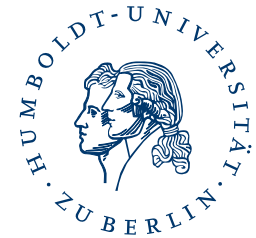
Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2017/18

Kontextsensitive Sprachen

Definition

Sei $G = (V, \Sigma, P, S)$ eine Grammatik.

1. G heißt vom Typ 3 oder regulär, falls für alle Regeln $u \to v$ gilt:
   $u \in V$ und $v \in \Sigma V \cup \Sigma \cup \{\varepsilon\}$.
   (d.h. alle Regeln haben die Form $A \to aB$, $A \to a$ oder $A \to \varepsilon$)
2. G heißt vom Typ 2 oder kontextfrei, falls für alle Regeln $u \to v$ gilt:
   $u \in V$. (d.h. alle Regeln haben die Form $A \to \alpha$)
3. G heißt vom Typ 1 oder kontextsensitiv, falls für alle Regeln $u \to v$ gilt:
   $|v| \geq |u|$. (mit Ausnahme der ε-Sonderregel, s. unten)
4. Jede Grammatik ist automatisch vom Typ 0.

Die ε-Sonderregel

In einer kontextsensitiven Grammatik ist auch die Regel $S \to \varepsilon$ zulässig.
Aber nur, wenn das Startsymbol S nur links vorkommt.

CFL ist echt in CSL enthalten

Bemerkung

- Wie wir gesehen haben, ist CFL in CSL enthalten.
- Zudem ist folgende Sprache nicht kontextfrei:

$$L = \{a^n b^n c^n \mid n \geq 1\}.$$

- L kann jedoch von einer kontextsensitiven Grammatik erzeugt werden.
- Daher ist CFL echt in CSL enthalten.

Eine kontextsensitive Grammatik für $\{a^n b^n c^n \mid n \geq 1\}$

Beispiel

- Betrachte die kontextsensitive Grammatik $G = (V, \Sigma, P, S)$ mit $V = \{S, B\}$, $\Sigma = \{a, b, c\}$ und den Regeln

$$P\colon S \to aSBc, abc\ (1,2) \quad cB \to Bc\ (3) \quad bB \to bb\ (4)$$

- In G lässt sich beispielsweise das Wort $w = aabbcc$ ableiten:

$$\underline{S} \underset{(1)}{\Rightarrow} a\underline{S}Bc \underset{(2)}{\Rightarrow} aab\underline{cB}c \underset{(3)}{\Rightarrow} aa\underline{bB}cc \underset{(4)}{\Rightarrow} aabbcc$$

- Allgemein gilt für alle $n \geq 1$:

$$\underline{S} \underset{(1)}{\Rightarrow}^{n-1} a^{n-1}\underline{S}(Bc)^{n-1} \underset{(2)}{\Rightarrow} a^{n-1}ab\underline{c(Bc)^{n-1}}$$
$$\underset{(3)}{\Rightarrow}^{\binom{n}{2}} a^n\underline{bB^{n-1}}c^n \underset{(4)}{\Rightarrow}^{n-1} a^n b^n c^n$$

- Also gilt $a^n b^n c^n \in L(G)$ für alle $n \geq 1$.

Eine kontextsensitive Grammatik für $\{a^n b^n c^n \mid n \geq 1\}$

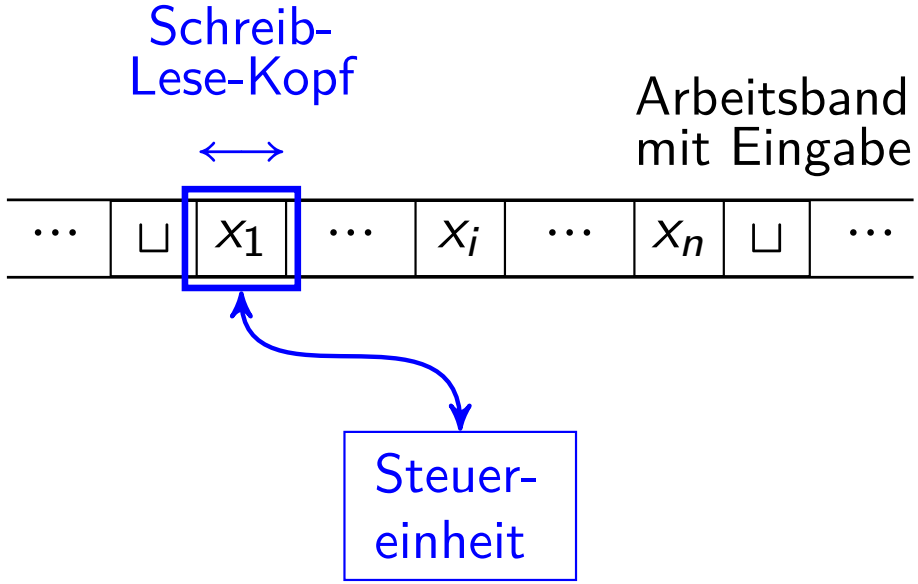
Beispiel (Schluss)

- Betrachte die kontextsensitive Grammatik $G = (V, \Sigma, P, S)$ mit $V = \{S, B\}$, $\Sigma = \{a, b, c\}$ und den Regeln

$$P\colon S \to aSBc, abc\ (1,2) \quad cB \to Bc\ (3) \quad bB \to bb\ (4)$$

- Umgekehrt folgt durch Induktion über die Ableitungslänge m, dass jede Satzform $\alpha \in (V \cup \Sigma)^*$ mit $S \Rightarrow^m \alpha$ die folgenden Bedingungen erfüllt:
  - $\#_a(\alpha) = \#_b(\alpha) + \#_B(\alpha) = \#_c(\alpha)$,
  - links von a und links von S kommen nur a's vor,
  - links von b kommen nur a's oder b's vor.
- Daraus ergibt sich, dass in G nur Wörter $w \in \Sigma^*$ der Form $w = a^n b^n c^n$ ableitbar sind, d.h. $L(G) = \{a^n b^n c^n \mid n \geq 1\} \in$ CSL. ◁

Die Turingmaschine



- Um ein geeignetes Maschinenmodell für die kontextsensitiven Sprachen zu finden, führen wir zunächst das Rechenmodell der nichtdeterministischen Turingmaschine (NTM) ein.
- Eine NTM erhält ihre Eingabe auf einem nach links und rechts unbegrenzten Band, das in Felder unterteilt ist. Daneben kann sie noch weitere Bänder benutzen, die zu Beginn der Rechnung leer sind.
- Während ihrer Rechnung kann sie den Schreib-Lese-Kopf auf dem Band in beide Richtungen bewegen und dabei die besuchten Bandfelder lesen sowie die gelesenen Zeichen gegebenenfalls überschreiben.

Das Rechenmodell der Turingmaschine

Definition

- Sei $k \geq 1$. Eine nichtdeterministische k-Band-Turingmaschine (k-NTM oder einfach NTM) wird durch ein 6-Tupel $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ beschrieben, wobei
  - Z eine endliche Menge von Zuständen,
  - Σ das Eingabealphabet (mit $\sqcup \notin \Sigma$; $\sqcup$ heißt Leerzeichen oder Blank),
  - Γ das Arbeitsalphabet (mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$),
  - $\delta\colon Z \times \Gamma^k \to \mathcal{P}(Z \times \Gamma^k \times \{L, R, N\}^k)$ die Überführungsfunktion,
  - q_0 der Startzustand und
  - $E \subseteq Z$ die Menge der Endzustände ist.
- Eine k-NTM M heißt deterministisch (kurz: M ist eine k-DTM oder einfach DTM), falls für alle $(q, a_1, \ldots a_k) \in Z \times \Gamma^k$ gilt:

$$\|\delta(q, a_1, \ldots a_k)\| \leq 1.$$

Das Rechenmodell der Turingmaschine

- Für $(q, b_1, \ldots, b_k, D_1, \ldots, D_k) \in \delta(p, a_1, \ldots a_k)$ schreiben wir auch

  $$(p, a_1, \ldots, a_k) \to (q, b_1, \ldots, b_k, D_1, \ldots, D_k).$$

- Eine solche Anweisung ist ausführbar, falls
  - p der aktuelle Zustand von M ist und
  - sich für $i = 1, \ldots, k$ der Kopf des i-ten Bandes auf einem mit a_i beschrifteten Feld befindet.
- Bei ihrer Ausführung
  - geht M vom Zustand p in den Zustand q über,
  - ersetzt M auf Band i das Symbol a_i durch b_i und
  - bewegt den Kopf auf Band i gemäß D_i (L: ein Feld nach links, R: ein Feld nach rechts, N: keine Bewegung).

Das Rechenmodell der Turingmaschine

Definition

- Eine Konfiguration ist ein $(3k+1)$-Tupel

$$K = (q, u_1, a_1, v_1, \ldots, u_k, a_k, v_k) \in Z \times (\Gamma^* \times \Gamma \times \Gamma^*)^k$$

und besagt, dass

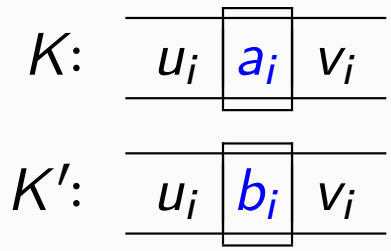
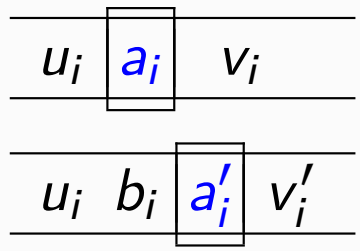
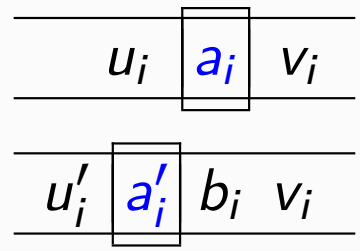
  - q der momentane Zustand ist und
  - das i-te Band mit $\ldots \sqcup u_i a_i v_i \sqcup \ldots$ beschriftet ist, wobei sich der Kopf auf dem Zeichen a_i befindet.

- Im Fall $k = 1$ schreiben wir für eine Konfiguration $K = (q, u, a, v)$ auch kurz $K = uqav$.

Das Rechenmodell der Turingmaschine

Definition

Seien $K = (p, u_1, a_1, v_1, \ldots, u_k, a_k, v_k)$ und $K' = (q, u'_1, a'_1, v'_1, \ldots, u'_k, a'_k, v'_k)$ Konfigurationen. K' heißt Folgekonfiguration von K (kurz $K \vdash K'$), falls eine Anweisung $(p, a_1, \ldots, a_k) \to (q, b_1, \ldots, b_k, D_1, \ldots, D_k)$ existiert, so dass für $i = 1, \ldots, k$ gilt:

im Fall $D_i = \mathrm{N}$:	$D_i = \mathrm{R}$:	$D_i = \mathrm{L}$:
		
$u'_i = u_i$, $a'_i = b_i$ und $v'_i = v_i$.	$u'_i = u_i b_i$ und $a'_i v'_i = \begin{cases} v_i, & v_i \neq \varepsilon, \\ \sqcup, & \text{sonst.} \end{cases}$	$u'_i a'_i = \begin{cases} u_i, & u_i \neq \varepsilon, \\ \sqcup, & \text{sonst} \end{cases}$ und $v'_i = b_i v_i$.

Das Rechenmodell der Turingmaschine

Definition

- Die Startkonfiguration von M bei Eingabe $x = x_1 \dots x_n \in \Sigma^*$ ist

$$K_x = \begin{cases} (q_0, \varepsilon, x_1, x_2 \dots x_n, \varepsilon, \sqcup, \varepsilon, \dots, \varepsilon, \sqcup, \varepsilon), & x \neq \varepsilon, \\ (q_0, \varepsilon, \sqcup, \varepsilon, \dots, \varepsilon, \sqcup, \varepsilon), & x = \varepsilon. \end{cases}$$

- Eine Rechnung von M bei Eingabe x ist eine (endliche oder unendliche) Folge von Konfigurationen $K_0, K_1, K_2 \dots$ mit $K_0 = K_x$ und $K_0 \vdash K_1 \vdash K_2 \cdots$.
- Die von M akzeptierte oder erkannte Sprache ist

$$L(M) = \{x \in \Sigma^* \mid \exists\, K \in E \times (\Gamma^* \times \Gamma \times \Gamma^*)^k : K_x \vdash^* K\}.$$

- Ein Wort x wird also genau dann von M akzeptiert (kurz: $M(x)$ akzeptiert), wenn es eine Rechnung von M bei Eingabe x gibt, bei der ein Endzustand erreicht wird.

Das Rechenmodell der Turingmaschine

Beispiel

Betrachte die 1-DTM $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ mit $Z = \{q_0, \ldots q_4\}$, $\Sigma = \{a, b\}$, $\Gamma = \Sigma \cup \{A, B, \sqcup\}$, $E = \{q_4\}$ und den Anweisungen

δ: $q_0 a \to q_1 AR$ (1) Beginn der Schleife: Falls ein a gelesen wird, ersetze es durch A und ...

$q_1 a \to q_1 aR$ (2) ... lies a's und B's bis ein b kommt (falls kein b

$q_1 B \to q_1 BR$ (3) kommt, halte ohne zu akzeptieren), ersetze

$q_1 b \to q_2 BL$ (4) das b durch ein B und ...

$q_2 a \to q_2 aL$ (5) ... bewege den Kopf wieder nach links bis

$q_2 B \to q_2 BL$ (6) auf das Feld hinter dem letzten A und

$q_2 A \to q_0 AR$ (7) gehe zum Beginn der Schleife

$q_0 B \to q_3 BR$ (8) Falls zu Beginn der Schleife ein B gelesen wird,

$q_3 B \to q_3 BR$ (9) teste, ob alle Eingabezeichen gelesen wurden

$q_3 \sqcup \to q_4 \sqcup N$ (10) (wenn ja, dann akzeptiere)

Das Rechenmodell der Turingmaschine

Beispiel (Fortsetzung)

Betrachte die 1-DTM $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ mit $Z = \{q_0, \ldots q_4\}$, $\Sigma = \{a, b\}$, $\Gamma = \Sigma \cup \{A, B, \sqcup\}$, $E = \{q_4\}$ und den Anweisungen

$$\begin{array}{llllll} \delta\colon & q_0 a \to q_1 AR & (1) & q_1 a \to q_1 aR & (2) & q_2 a \to q_2 aL \quad (5) \quad q_0 B \to q_3 BR \quad (8) \\ & & & q_1 B \to q_1 BR & (3) & q_2 B \to q_2 BL \quad (6) \quad q_3 B \to q_3 BR \quad (9) \\ & & & q_1 b \to q_2 BL & (4) & q_2 A \to q_0 AR \quad (7) \quad q_3 \sqcup \to q_4 \sqcup N \quad (10) \end{array}$$

- Dann akzeptiert M die Eingabe $aabb$ wie folgt:

$$\begin{aligned} q_0 aabb &\underset{(1)}{\vdash} A q_1 abb \underset{(2)}{\vdash} A a q_1 bb \underset{(4)}{\vdash} A q_2 aBb \underset{(5)}{\vdash} q_2 AaBb \\ &\underset{(7)}{\vdash} A q_0 aBb \underset{(1)}{\vdash} AA q_1 Bb \underset{(3)}{\vdash} AAB q_1 b \underset{(4)}{\vdash} AA q_2 BB \\ &\underset{(6)}{\vdash} A q_2 ABB \underset{(7)}{\vdash} AA q_0 BB \underset{(8)}{\vdash} AAB q_3 B \underset{(9)}{\vdash} AABB q_3 \sqcup \\ &\underset{10}{\vdash} AABB q_4 \sqcup \end{aligned}$$

- Ähnlich lässt sich für ein beliebiges $n \geq 1$ zeigen, dass $a^n b^n \in L(M)$ ist.

Das Rechenmodell der Turingmaschine

Beispiel (Schluss)

δ: $q_0a \to q_1AR$ (1) $\quad q_1a \to q_1aR$ (2) $\quad q_2a \to q_2aL$ (5) $\quad q_0B \to q_3BR$ (8)

$q_1B \to q_1BR$ (3) $\quad q_2B \to q_2BL$ (6) $\quad q_3B \to q_3BR$ (9)

$q_1b \to q_2BL$ (4) $\quad q_2A \to q_0AR$ (7) $\quad q_3\sqcup \to q_4\sqcup N$ (10)

- Andererseits führen die Eingaben aba, abb und aab auf die Rechnungen

$$q_0aba \underset{(1)}{\vdash} Aq_1ba \underset{(4)}{\vdash} q_2ABa \underset{(7)}{\vdash} Aq_0Ba \underset{(8)}{\vdash} ABq_3a \text{ und}$$

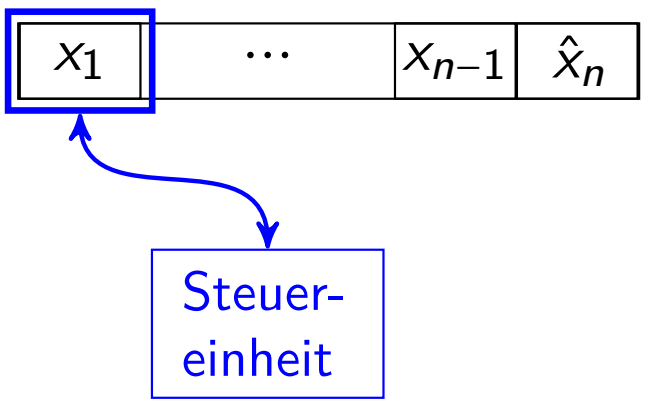
$$q_0abb \underset{(1)}{\vdash} Aq_1bb \underset{(4)}{\vdash} q_2ABb \underset{(7)}{\vdash} Aq_0Bb \underset{(8)}{\vdash} ABq_3b \text{ und}$$

$$q_0aab \underset{(1)}{\vdash} Aq_1ab \underset{(2)}{\vdash} Aaq_1b \underset{(4)}{\vdash} Aq_2aB \underset{(5)}{\vdash} q_2AaB \underset{(7)}{\vdash} Aq_0aB \underset{(1)}{\vdash} AAq_1B \underset{(8)}{\vdash} AABq_1\sqcup$$

- Da diese nicht fortsetzbar sind und M deterministisch ist, kann M nicht den Endzustand q_4 erreichen, d.h. $abb, aab \notin L(M)$.
- Tatsächlich lässt sich zeigen, dass $L(M) = \{a^n b^n \mid n \geq 1\}$ ist. $\triangleleft$

Ein Maschinenmodell für CSL

- In den Übungen werden wir auch für die Sprache $L' = \{a^n b^n c^n \mid n \geq 1\}$ eine 1-DTM M' konstruieren.
- Wie M besucht auch M' außer den Eingabefeldern nur das erste Blank hinter der Eingabe.
- Dies ist notwendig, damit M' das Ende der Eingabe erkennen kann.
- Falls wir jedoch das letzte Zeichen der Eingabe x markieren, muss der Eingabebereich im Fall $|x| \geq 1$ für diesen Zweck nicht mehr verlassen werden:



- NTMs und DTMs mit dieser Eigenschaft werden auch als LBAs bzw. DLBAs bezeichnet.

Linear beschränkte Automaten

Definition

- Für ein Alphabet Σ sei $\hat{\Sigma} = \Sigma \cup \{\hat{a} \mid a \in \Sigma\}$.
- Für $x = x_1 \dots x_n \in \Sigma^*$ sei $\hat{x} = x_1 \dots x_{n-1}\hat{x}_n$.
- Eine 1-NTM $M = (Z, \hat{\Sigma}, \Gamma, \delta, q_0, E)$ heißt LBA, falls gilt:

  $$\forall x \in \Sigma^+ : K_{\hat{x}} \vdash^* uqav \Rightarrow |uav| \leq |x|$$

- Die von einem LBA M akzeptierte oder erkannte Sprache ist

  $$L(M) = \{x \in \Sigma^* \mid M(\hat{x}) \text{ akzeptiert}\}$$

- Ein deterministischer LBA wird auch als DLBA bezeichnet.
- Die Klasse der deterministisch kontextsensitiven Sprachen ist DCSL $= \{L(M) \mid M \text{ ist ein DLBA}\}$.

Bemerkung

Jede k-NTM, die bei Eingaben der Länge n höchstens linear viele (also $cn + c$ für eine Konstante c) Bandfelder besucht, kann von einem LBA simuliert werden. LBA steht also für linear beschränkter Automat.