Einführung in die Theoretische Informatik

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

WS 2022/23

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, falls das Startsymbol S nicht auf der rechten Seite einer Regel vorkommt

CFL ist echt in CSL enthalten

- Wie wir gesehen haben, ist CFL in CSL enthalten
- Zudem ist die Sprache $L = \{a^n b^n c^n \mid n \geq 1\}$ nicht kontextfrei
- L kann jedoch von einer kontextsensitiven Grammatik erzeugt werden (siehe nächste Folie)
- 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 (Fortsetzung)

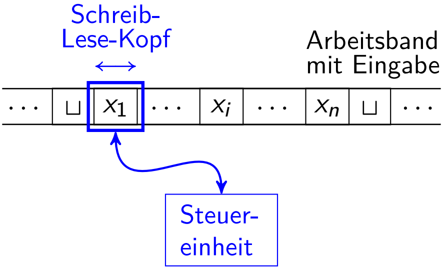
- Betrachte die kontextsensitive Grammatik $G = (V, \Sigma, P, S)$ mit $V = \{S, B\}$, $\Sigma = \{a, b, c\}$ und den Regeln

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

- 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. $L(G) = \{a^n b^n c^n \mid n \geq 1\} \in \mathrm{CSL}$ $\triangleleft$

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; zudem kann sie weitere Bänder benutzen, die zu Beginn der Rechnung komplett leer sind
- In jedem Rechenschritt kann sie die aktuell besuchten Bandfelder lesen, die gelesenen Zeichen überschreiben und den Schreib-Lese-Kopf auf jedem Band um maximal ein Feld nach links oder rechts bewegen

Das Rechenmodell der Turingmaschine

Definition. Sei $k \geq 1$.

- Eine nichtdeterministische k-Band-Turingmaschine (kurz k-NTM oder einfach NTM) wird durch ein 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 $\Sigma \cup \{\sqcup\} \subseteq \Gamma$)
  - δ: $Z \times \Gamma^k \to \mathcal{P}(Z \times \Gamma^k \times \{L, R, N\}^k)$ 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:

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

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

$$(p, a_1, \dots, a_k) \rightarrow (q, b_1, \dots, b_k, D_1, \dots, D_k)$$

- Eine solche Anweisung ist ausführbar, falls
  - p der aktuelle Zustand von M ist und
  - sich für $i = 1, \dots, k$ 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 auf Band $i = 1, \dots, k$ das Symbol a_i durch b_i und
  - bewegt den Kopf auf Band $i = 1, \dots, k$ gemäß $D_i \in \{L, R, N\}$ ($D_i = L$: um ein Feld nach links; $D_i = R$: um ein Feld nach rechts; $D_i = N$: keine Bewegung)

Das Rechenmodell der Turingmaschine

- 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 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 $k = 1$ notieren wir eine Konfiguration $K = (q, u, a, v)$ auch kurz in der Form $K = uqav$ (wobei wir annehmen, dass $Z \cap \Gamma = \varnothing$ ist)

Das Rechenmodell der Turingmaschine

- 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:

$D_i = \mathrm{N}$	$D_i = \mathrm{R}$	$D_i = \mathrm{L}$
K: $u_i \boxed{a_i} v_i$	K: $u_i \boxed{a_i} v_i$	K: $u_i \boxed{a_i} v_i$
K': $u_i \boxed{b_i} v_i$	K': $u_i\ b_i \boxed{a_i'} v_i'$	K': $u_i' \boxed{a_i'} b_i\ v_i$
$u_i' = u_i$ $a_i' = b_i$ $v_i' = v_i$	$u_i' = u_i b_i$ $a_i' v_i' = \begin{cases} v_i, & v_i \neq \varepsilon \\ \sqcup, & \text{sonst} \end{cases}$	$u_i' a_i' = \begin{cases} u_i, & u_i \neq \varepsilon \\ \sqcup, & \text{sonst} \end{cases}$ $v_i' = b_i v_i$

Das Rechenmodell der Turingmaschine

- Man beachte, dass sich die Länge der Bandinschrift $u_i a_i v_i$ beim Übergang von K zu K' genau dann um 1 erhöht, wenn in K' zum ersten Mal ein neues Feld auf dem i-ten Band besucht wird
- Andernfalls bleibt die Länge von $u_i a_i v_i$ unverändert
- Die Länge von $u_i a_i v_i$ entspricht also genau der Anzahl der bisher auf dem i-ten Band besuchten Felder (inkl. Eingabezeichen im Fall $i = 1$)
- 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) Konfigurationenfolge $K_0, K_1, K_2 \dots$ mit $K_0 = K_x$ und $K_0 \vdash K_1 \vdash K_2 \dots$
- Die von M akzeptierte oder erkannte Sprache ist

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

- M akzeptiert also genau dann eine Eingabe x (kurz: $M(x)$ akzeptiert), falls mindestens eine Rechnung von $M(x)$ einen Endzustand erreicht

Beispiel

Betrachte die 1-DTM $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ mit $Z = \{q_0, \dots q_4\}$, $\Sigma = \{a, b\}$, $\Gamma = \Sigma \cup \{A, B, \sqcup\}$, $E = \{q_4\}$ 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 das b
	$q_1 b \to q_2 BL$	(4)	durch ein B und ...
	$q_2 a \to q_2 aL$	(5)	... bewege den Kopf wieder nach links bis auf das
	$q_2 B \to q_2 BL$	(6)	Feld hinter dem letzten A und gehe zum Beginn
	$q_2 A \to q_0 AR$	(7)	der Schleife
	$q_0 B \to q_3 BR$	(8)	Falls zu Beginn der Schleife ein B gelesen wird, teste,
	$q_3 B \to q_3 BR$	(9)	ob alle Eingabezeichen gelesen wurden (wenn ja,
	$q_3 \sqcup \to q_4 \sqcup N$	(10)	dann akzeptiere, sonst halte ohne zu akzeptieren)

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

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

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

$$
\begin{array}{lllllllll}
q_0 aabb & \underset{(1)}{\vdash} & Aq_1 abb & \underset{(2)}{\vdash} & Aaq_1 bb & \underset{(4)}{\vdash} & Aq_2 aBb & \underset{(5)}{\vdash} & q_2 AaBb \\
 & \underset{(7)}{\vdash} & Aq_0 aBb & \underset{(1)}{\vdash} & AAq_1 Bb & \underset{(3)}{\vdash} & AABq_1 b & \underset{(4)}{\vdash} & AAq_2 BB \\
 & \underset{(6)}{\vdash} & Aq_2 ABB & \underset{(7)}{\vdash} & AAq_0 BB & \underset{(8)}{\vdash} & AABq_3 B & \underset{(9)}{\vdash} & AABBq_3 \sqcup \\
 & \underset{(10)}{\vdash} & AABBq_4 \sqcup & & & & & &
\end{array}
$$

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

Beispiel (Schluss)

$$\delta\colon \begin{array}{llllllll} q_0a \to q_1AR & (1) & q_1a \to q_1aR & (2) & q_2a \to q_2aL & (5) & q_0B \to q_3BR & (8) \\ & & q_1B \to q_1BR & (3) & q_2B \to q_2BL & (6) & q_3B \to q_3BR & (9) \\ & & q_1b \to q_2BL & (4) & q_2A \to q_0AR & (7) & q_3\sqcup \to q_4\sqcup N & (10) \end{array}$$

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

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

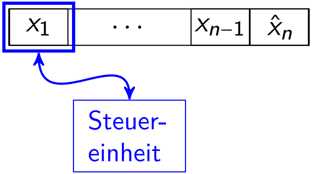
$$q_0abb \underset{(1)}{\vdash} Aq_1bb \underset{(4)}{\vdash} q_2ABb \underset{(7)}{\vdash} Aq_0Bb \underset{(8)}{\vdash} ABq_3b \text{ und}$$

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

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

Ein Maschinenmodell für CSL

- In den Übungen werden wir eine 1-DTM M' für die Sprache $L' = \{a^n b^n c^n \mid n \geq 1\}$ 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:



- 1-NTMs und 1-DTMs, die den Bereich der markierten Eingabe x im Fall $|x| \geq 1$ nicht verlassen, werden auch als LBAs bzw. DLBAs bezeichnet