Prof. Javier Esparza
Kevin Kappelmann, Simon Roßkopf, Lukas Stevens

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2023 – Übungsblatt 6

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Für den Rest des Semesters gilt: $\mathbb{N} := \mathbb{N} := \{0, 1, 2, ...\}$ und $\mathbb{N}_+ := \mathbb{N} \setminus \{0\}$.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü6.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- (inhärent) mehrdeutig
- Chomsky-Normalform
- Greibach-Normalform
- Pumping-Lemma für kontextfreie Sprachen

Übung und Nachbereitung

Übungsaufgabe Ü6.2. (*Sprache einer kontextfreien Grammatik*)

Sei $\Sigma := \{a, b\}$. Wir betrachten erneut die Grammatik G aus Übungsaufgabe 5.5 (b) für die Sprache $L := \{w \in \Sigma^* \mid w \neq w^R\}$:

$$S \to aSa \mid bSb \mid aTb \mid bTa$$
$$T \to aT \mid bT \mid \epsilon$$

Zeigen Sie $L(G) = L$ formal. Beweisen Sie dabei auch induktiv, welche Sprache von T erzeugt wird.

Lösungsskizze. Zunächst zeigen wir $L_G(T) = \Sigma^*$.

(a) Wir zeigen $L_G(T) \subseteq \Sigma^*$ per Induktion über die Erzeugung von Wörtern in $L_G(T)$. Der Fall $T \to \epsilon$ ist klar. Im Fall $T \to aT$ erhalten wir per I.H. $w \in \Sigma^*$ mit $T \to aT \to^* w$. Somit $aw \in \Sigma^*$. Der Fall $T \to bT$ folgt analog.

(b) Wir zeigen $\Sigma^* \subseteq L_G(T)$ per Induktion über die Wortlänge $w \in \Sigma^*$. Der Fall $w = \epsilon$ gilt wegen $T \to \varepsilon$. Im Fall $w \in \Sigma^{n+1}$ gibt es $x \in \Sigma$ und $v \in \Sigma^n$ mit $w = xv$. Da $|v| < |w|$ und $v \in \Sigma^*$ gibt es somit per I.H. eine Ableitung $T \to^* v$. Falls $x = a$ folgt $T \to aT \to^* av$. Falls $x = b$ folgt $T \to bT \to^* bv$.

Nun zeigen wir $L(G) = L$.

(a) Wir zeigen $L(G) \subseteq L$ per Induktion über die Erzeugung von Wörtern in $L_G(S)$. Im Fall $S \to aTb$ erhalten wir nach vorherigem $w \in \Sigma^*$ mit $S \to aTb \to^* awb$. Somit $awb \in \Sigma^*$. Außerdem gilt $awb \neq bw^Ra = (awb)^R$. Somit $awb \in L$. Der Fall $S \to bwa$ folgt analog.

Im Fall $S \to aSa$ erhalten wir per I.H. $w \in \Sigma^*$ mit $w \neq w^R$ und $S \to aSa \to^* awa$. Somit gilt $awa \neq aw^Ra = (awa)^R$. Der Fall $S \to bwb$ folgt analog.

(b) Wir zeigen $L \subseteq L(G)$ per Induktion über die Wortlänge n von $w = w_1 \cdots w_n \in L$. Da $w \in L$, gilt $w \neq w^R$.

In den Fällen $n \in \{0, 1\}$ folgt $w = w^R$. Mit der Annahme $w \neq w^R$ folgt ein Widerspruch und somit $w \in L(G)$ (ex falso quodlibet).

Im Fall $n \geq 2$ können wir das Wort w schreiben als $w = xvy$ mit $x, y \in \Sigma$ und $v \in \Sigma^{n-2}$. Per I.H. erhalten wir $S \to^* v$ für alle $v \in \Sigma^*$ für die $|v| < n$ und $v \neq v^R$ gilt. Wir betrachten zwei Fälle:

(1) Falls $x = y$ folgt wegen $xvx = xvy = w \neq w^R = yv^Rx = xv^Rx$, dass $v \neq v^R$. Per I.H. erhalten wir somit eine Ableitung $S \to^* v$. Falls nun $x = a$, folgt $S \to aSa \to^* ava$ und analoges gilt für $x = b$.

(2) Falls $x \neq y$ gibt es eine Ableitung $S \to xTy \to^* xvy$, da wir bereits gezeigt haben, dass $L_G(T) = \Sigma^*$.

Übungsaufgabe Ü6.3. (*Pumping-Lemma für Kontextfreie Sprachen*)

Zeigen Sie unter Verwendung des Pumping Lemmas für kontextfreie Sprachen, dass die folgende Sprache nicht kontextfrei ist:

$$L := \{a^n b^m c^l \mid n, m, l \in \mathbb{N} \wedge n > m \wedge n > l\}$$

Lösungsskizze.

- Angenommen, L wäre kontextfrei. Dann können wir das Pumping-Lemma für kontextfreie Sprachen anwenden.
- Sei $n \in \mathbb{N}_+$ eine Pumping-Lemma-Zahl für L.
- Es gilt $z := a^{n+1}b^n c^n \in L$ und $|z| \geq n$.
- Es gibt also für z eine Zerlegung $z = uvwxy$ mit den Eigenschaften

  (1) $vx \neq \varepsilon$

  (2) $|vwx| \leq n$

  (3) $\forall i \in \mathbb{N}.\ uv^iwx^iy \in L$

- Wir unterscheiden die folgenden Fälle:
  - $|vx|_a > 0$: Aus (2) und der Wahl von z folgt, dass $|vwx|_c = 0$. Folglich liegt das Wort uv^0wx^0y nicht mehr in L, da es höchstens so viele a's wie c's enthält. Dies ist ein Widerspruch zu (3).
  - $|vx|_a = 0$: In diesem Fall muss wegen (1) $|vx|_b > 0$ oder $|vx|_c > 0$ gelten. Folglich liegt das Wort uv^3wx^3y nicht mehr in L_1, da es entweder mehr b's oder mehr c's als a's enthält. Dies ist ein Widerspruch zu (3).
- Jeder Fall führt zu einem Widerspruch. Somit ist L nicht kontextfrei.

Übungsaufgabe Ü6.4. (*Chomsky-Normalform*)

Betrachten Sie die folgende Grammatik G:

$$S \to ASA \mid aB \qquad A \to B \mid S \qquad B \to b \mid \varepsilon$$

(a) Überführen Sie die Grammatik in Chomsky-Normalform.

(b) Erklären Sie in eigenen Worten, wie Sie nach Überführen einer Grammatik in CNF überprüfen können, dass Sie keine Fehler gemacht haben.[1]

Lösungsskizze.

(a) Wir überführen die Grammatik schrittweise in Chomsky-Normalform. Das Vorgehensweise für ε-Produktionen und Kettenproduktionen (Schritte 3 und 4) unterscheidet sich leicht von der Vorlesung führt aber zum selben Ergebnis. Die hier beschriebene Vorgehensweise ist allerdings leichter durchzuführen ist und hilft dabei, Fehler zu vermeiden.

(1) *Entfernen von Terminalen in langen Produktionen.* In jeder Regel $X \to \gamma$ mit $|\gamma| \geq 2$ Ersetzen jedes Vorkommen eines Terminals x durch X_x und Ergänzen der benötigten Produktionen $X_x \to x$:

$$G'\colon S \to ASA \mid X_aB \qquad A \to B \mid S \qquad B \to b \mid \varepsilon \qquad X_a \to a$$

Alle rechten Seiten sind jetzt von der Form $V^* \cup \Sigma$.

(2) *Entfernen langer Produktionen.* Überführen aller rechten Seiten, welche aus mindestens drei Variablen bestehen, in quadratische Monome über V. Die einfachste Variante ist dabei, aus XYZ einfach XX_{YZ} machen, wobei X_{YZ} einfach eine Hilfsvariable ist, die das Ergebnis von YZ mittels $X_{YZ} \to YZ$ zugewiesen bekommt. Damit:

$$G''\colon \begin{aligned} & S \to AX_{SA} \mid X_aB \\ & A \to B \mid S \\ & B \to b \mid \varepsilon \\ & X_a \to a \\ & X_{SA} \to SA \end{aligned}$$

(3) *Entfernen von ε-Produktionen.*

- Berechnen alle Variablen, die ε erzeugen können. Zuerst sind das alle Variablen X mit ε-Produktion ($X \to \varepsilon$). Dann kommen iterativ alle Variablen dazu, die eine Produktionen haben, deren rechte Seite nun ε werden kann. Das machen wir solange, bis wir keine neuen Variablen mehr finden. Formal:

$$\begin{aligned} E_0 &:= \{X \mid (X \to \varepsilon) \in P\} \\ E_{k+1} &:= E_k \cup \{X \mid (X \to \gamma) \in P \text{ und } \gamma \in E_k^*\} \quad \text{bis } E_{k+1} = E_k. \end{aligned}$$

  In unserem Fall:

$$E_0 = \{B\} \qquad E_1 = \{B, A\} = E_2$$

  Hier erkennt man, ob die Grammatik ε erzeugt: $\varepsilon \in L(G) \leftrightarrow S \in E_*$.

[1] Hier geht es nicht um einen formalen Beweis, dass die beiden Sprachen gleich sind, sondern um eine Strategie, wie Sie bei Aufgaben wie im vorherigen Aufgabenteil Fehler vermeiden.

- Erzeugen zusätzlicher Produktionen, welche alle möglichen Kombinationen von ε-Produktionen beachten:

$$\begin{array}{lcl}
S \to AX_{SA} & \rightsquigarrow & S \to AX_{SA} \mid X_{SA} \\
S \to X_a B & \rightsquigarrow & S \to X_a B \mid X_a \\
A \to B & \rightsquigarrow & A \to B \mid \varepsilon \\
A \to S & \rightsquigarrow & A \to S \\
B \to b & \rightsquigarrow & B \to b \\
B \to \varepsilon & \rightsquigarrow & B \to \varepsilon \\
X_a \to a & \rightsquigarrow & X_a \to a \\
X_{SA} \to SA & \rightsquigarrow & X_{SA} \to SA \mid S
\end{array}$$

- Entfernen aller ε-Produktionen:

$$G''' : \begin{array}{l}
S \to AX_{SA} \mid X_{SA} \mid X_a B \mid X_a \\
A \to B \mid S \\
B \to b \\
X_a \to a \\
X_{SA} \to SA \mid S
\end{array}$$

(4) *Entfernen von Kettenproduktionen.*

- Wir berechnen, welche Variablen mit Hilfe von Kettenproduktionen in welche anderen Variablen umgewandelt werden können. Wir beginnen mit den vorhanden Kettenproduktionen und fügen dann iterative die transitiven Kettenproduktionen hinzu. Formal:

$$\begin{aligned}
T_0 &:= \{(X \to Y) \in P \mid X, Y \in V\} \\
T_{k+1} &:= T_k \cup \{(X \to Y) \mid X, Y, Z \in V \wedge (X \to Z) \in T_k \wedge (Z \to Y) \in T_k\} \\
&\qquad \text{bis } T_{k+1} = T_k.
\end{aligned}$$

  In unserem Fall:

$$\begin{array}{lcl}
T_0 & = & \{(S \to X_{SA}), (S \to X_a), (A \to B) \\
 & & \quad , (A \to S), (X_{SA} \to S)\} \\
T_1 & = & T_0 \cup \{(S \to S), (A \to X_{SA}), (A \to X_a) \\
 & & \qquad , (X_{SA} \to X_{SA}), (X_{SA} \to X_a)\} = T_2
\end{array}$$

- Entfernen der Kettenproduktionen:

$$\begin{array}{l}
S \to AX_{SA} \mid X_a B \\
B \to b \\
X_a \to a \\
X_{SA} \to SA
\end{array}$$

- Wenn wir X mit Hilfe der entfernten Kettenproduktionen in Y umwandeln konnten, dann müssen wir X ermöglichen alle Produktionen von Y durchzuführen.

Formal:

$$G'''' : \quad \begin{array}{l} S \to AX_{SA} \mid SA \mid X_aB \mid a \\ A \to b \mid AX_{SA} \mid SA \mid X_aB \mid a \\ B \to b \\ X_a \to a \\ X_{SA} \to SA \mid AX_{SA} \mid SA \mid X_aB \mid a \end{array}$$

(b)
- Prüfen, dass alle Produktionen die Form $A \to BC$ oder $A \to a$ haben.
- Insbesondere darf ϵ nur vom Startsymbol (oder gar nicht, wie hier) produziert werden.
- Für einige kleine/leichte Worte überprüfen, dass Sie von beiden Grammatiken (nicht) erzeugt werden können.

Zusätzliche Übungsaufgabe Ü6.5. (*Fitnessstudio*)

Diese Aufgabe können Sie zusätzlich als Nachbereitung lösen. Es wird kein neuer Inhalt in dieser Aufgabe behandelt.

Wir betrachten Pfeil-Sprachen über dem Alphabet $\Sigma = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$. Wir interpretieren dabei ein Wort $w \in \Sigma^*$ als einen Pfad in einem 2D-Gitter.

(a) Geben Sie eine formale Definition für die folgenden beiden Sprachen an:

(1) Pfade, die in den Ursprung zurückkehren.

(2) Pfade, die „in großer Kurve umkehren" — *beliebig weit* nach rechts fahren, dann *noch weiter* entweder nach oben oder unten gehen und letztlich wieder umkehren und *noch weiter* nach links fahren. Diese Pfade enden dann links vom Ursprung.

(b) Zeigen Sie mit Hilfe des Pumping-Lemma für kontextfreie Sprachen, dass diese Sprachen nicht kontextfrei sind.

Lösungsskizze. Für diese Aufgabe gibt es eine Video-Lösung.

(a) (1)

$$L_a = \{w \in \Sigma^* \mid |w|_\uparrow = |w|_\downarrow \wedge |w|_\leftarrow = |w|_\rightarrow\}$$

(2)

$$L_b = \bigcup_{i<j<k} L(\rightarrow^i (\uparrow^j \mid \downarrow^j) \leftarrow^k)$$

(b) (1)
- Wir nehmen an, dass L_a kontextfrei ist und führen diese Annahme zum Widerspruch.
- Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L_a.
- Dann gilt $z := \rightarrow^n\uparrow^n\leftarrow^n\downarrow^n \in L_a$ und $|z| \geq n$.
- Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1)\ vx \neq \epsilon \qquad (2)\ |vwx| \leq n \qquad (3)\ \forall i \in \mathbb{N}.\ uv^iwx^iy \in L_a$$

- Aufgrund von (1) unterscheiden wir die folgenden Fälle:

- $|vx|_\rightarrow > 0$: Wegen (2) gilt $|vx|_\leftarrow = 0$. Allerdings gilt auch:

$$|uv^2wx^2y|_\rightarrow = n + |vx|_\rightarrow > n = |uv^2wx^2y|_\leftarrow$$

  Dies ist ein Widerspruch zu (3).
- $|vx|_\uparrow > 0$: Wegen (2) gilt $|vx|_\downarrow = 0$. Allerdings gilt auch:

$$|uv^2wx^2y|_\uparrow = n + |vx|_\uparrow > n = |uv^2wx^2y|_\downarrow$$

  Dies ist ein Widerspruch zu (3).
- $|vx|_\leftarrow > 0$: Wegen (2) gilt $|vx|_\rightarrow = 0$. Allerdings gilt auch:

$$|uv^2wx^2y|_\rightarrow = n < n + |vx|_\leftarrow = |uv^2wx^2y|_\leftarrow$$

  Dies ist ein Widerspruch zu (3).
- $|vx|_\downarrow > 0$: Wegen (2) gilt $|vx|_\uparrow = 0$. Allerdings gilt auch:

$$|uv^2wx^2y|_\uparrow = n < n + |vx|_\downarrow = |uv^2wx^2y|_\downarrow$$

  Dies ist ein Widerspruch zu (3).

- Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L_a nicht kontextfrei.

(2)
- Wir nehmen an, dass L_b kontextfrei ist und führen diese Annahme zum Widerspruch.
- Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L_b.
- Dann gilt $z :=\rightarrow^n\uparrow^{n+1}\leftarrow^{n+2}\in L_b$ und $|z| \geq n$.
- Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1)\ vx \neq \epsilon \qquad (2)\ |vwx| \leq n \qquad (3)\ \forall i \in \mathbb{N}.\ uv^iwx^iy \in L_b$$

- Zuerst informell: Da $|vwx| \leq n$, kann vwx nur von der Form $\rightarrow^*\uparrow^*$ oder $\uparrow^*\leftarrow^*$ sein. Wegen $|vx| > 0$ muss mindestens ein Pfeil gepumpt werden. Gilt $|vx|_\rightarrow > 0$, dann können wir die Anzahl der $\rightarrow$ über die Anzahl der $\leftarrow$ pumpen, enthält vx keinen $\rightarrow$ aber mindestens ein $\uparrow$, so kann man die Anzahl der $\rightarrow$ auf höchstens n reduzieren, indem man vx entfernt. Andernfalls besteht vx nur aus $\leftarrow$, dann aus mindestens einem, so dass man durch Entfernen von vx die Anzahl der $\leftarrow$ auf $n+1$ oder weniger reduziert wird, man also höchstens so viele $\leftarrow$ wie $\uparrow$ hat.
- Formal: Aufgrund von (1) unterscheiden wir die folgenden Fälle:
  - $|vx|_\rightarrow > 0$: Wegen (2) gilt $|vx|_\leftarrow = 0$. Allerdings gilt auch:

$$|uv^3wx^3y|_\rightarrow = n + 2|vx|_\rightarrow \geq n + 2 = |uv^3wx^3y|_\leftarrow$$

    Dies ist ein Widerspruch zu (3).

- $|vx|_\rightarrow = 0$ und $|vx|_\uparrow > 0$: Dann gilt:

  $$|uv^0wx^0y|_\uparrow = n + 1 - |vx|_\uparrow \leq n = |uv^0wx^0y|_\rightarrow$$

  Daher ist $uv^0wx^0y \notin L_b$, ein Widerspruch zu (3).
- $|vx|_\rightarrow = 0$ und $|vx|_\uparrow = 0$: Dann muss $|vx|_\leftarrow > 0$ gelten, und es folgt:

  $$|uv^0wx^0y|_\leftarrow = n + 2 - |vx|_\leftarrow < n + 1 = |uv^0wx^0y|_\uparrow$$

  Daher ist $uv^0wx^0y \notin L_b$, ein Widerspruch zu (3).

Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L_b nicht kontextfrei.