Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 6

Vorlesung Theoretische Grundlagen der Informatik im WS 21/22

Ausgabe: 14. Januar 2022
Abgabe: 28. Januar 2022 (digital im ILIAS)

Bitte bearbeiten Sie die Aufgaben **handschriftlich** und laden Sie eine eingescannte PDF-Version im Übungsmodul Ihrer ILIAS-Tutoriumsgruppe hoch! Beschriften Sie Ihren handschriftlichen Aufschrieb gut sichtbar mit Name und Matrikelnummer. Nicht handschriftliche oder unbeschriftete Abgaben werden nicht akzeptiert!

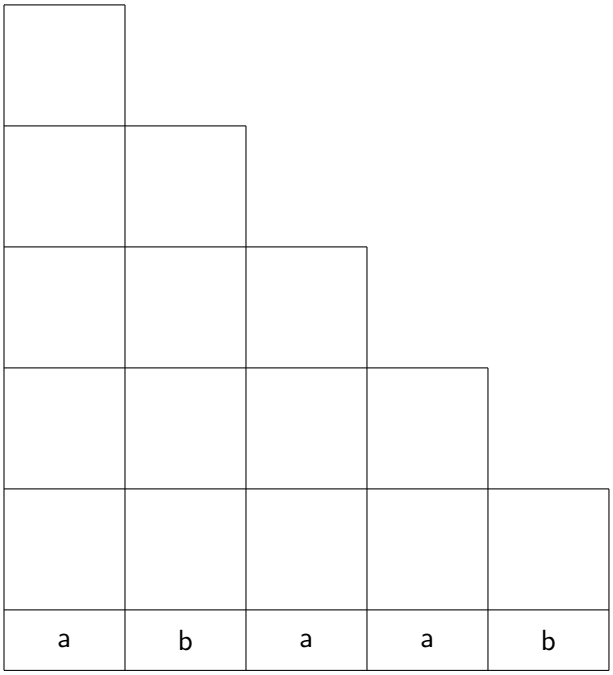
Aufgabe 1

(2 + 1 + 1 + 2 = 6 Punkte)

Gegeben sei die kontextfreie Grammatik $G = (\Sigma, V, S, R)$ in Chomsky-Normalform mit Terminalen $\Sigma = \{\mathsf{a}, \mathsf{b}\}$, Nichtterminalen $V = \{S, A, B, C, D, E\}$, Startsymbol S und Produktionen

$$\begin{aligned} R = \{ & S \to AB \mid SS \mid \mathsf{a}, \\ & A \to BS \mid CD \mid \mathsf{b}, \\ & B \to DD \mid \mathsf{b}, \\ & C \to DE \mid \mathsf{a} \mid \mathsf{b}, \\ & D \to \mathsf{a}, \\ & E \to SS\} \end{aligned}$$

(a) Überprüfen Sie mit dem CYK-Algorithmus, ob das Wort abaab in der Sprache $L(G)$ enthalten ist.



(b) Geben Sie einen Syntaxbaum für das Wort abaab an.

Betrachten Sie für eine Sprache L die Sprache $Pre(L) \subseteq L$, die ein Wort $w \in L$ genau dann enthält, wenn auch alle Präfixe von w in L enthalten sind, d.h.

$$Pre(L) = \left\{ w \;\middle|\; w[1 \dots i] \in L \text{ für alle } i \in \{1, \dots |w|\} \right\},$$

wobei $w[1 \dots i]$ das Präfix von w der Länge i ist.

(c) Sei $L = \{w \in \{\mathsf{a}, \mathsf{b}\}^* \mid w \text{ ist ein Palindrom}\}$. Geben Sie $Pre(L)$ an.

(d) Entwerfen Sie einen möglichst effizienten Algorithmus, der prüft, ob $w \in Pre(L(G))$ für ein gegebenes Wort $w \in \Sigma^*$ und eine kontextfreie Grammatik G in Chomsky-Normalform über einem endlichen Alphabet Σ. Welche Laufzeit hat Ihr Algorithmus?

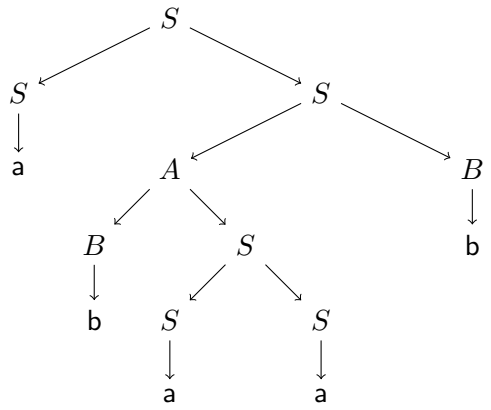
Lösung:

(a) Vergleiche folgende Tabelle:

S, E				
S, E	A, S			
$\emptyset$	S, A	S		
$\emptyset$	A	S, A, B, E	$\emptyset$	
S, C, D	A, B, C	S, C, D	S, C, D	A, B, C
a	b	a	a	b

Das Wort ist also in der Sprache enthalten.

(b) Vergleiche folgenden Syntaxbaum:



(c) Löscht man das letzte Zeichen eines Palindroms, ist das resultierende Wort genau dann ein Palindrom, wenn im ursprünglichen Wort alle Zeichen gleich sind. Es ist also $Pre(L) = \{a^n \mid n \in \mathbb{N}_0\} \cup \{b^n \mid n \in \mathbb{N}_0\}$.

(d) Sei $w \in \Sigma^*$ ein Wort, für das wir prüfen wollen, ob $w \in Pre(L(G))$ gilt. Dazu führen wir den CYK-Algorithmus auf w aus. Dieser berechnet für jedes Teilwort $w[i \dots j]$ von w, von welchen Nichtterminalen $w[i \dots j]$ abgeleitet werden kann. Kann jedes Präfix $w[1 \dots i]$ von w mit $i = 1, \dots, |w|$ von S abgeleitet werden, ist $w \in Pre(L(G))$. Sonst ist $w \notin Pre(L(G))$. Es muss also nur geprüft werden, ob jede Zelle der linkesten Spalte in der CYK-Tabelle das Startsymbol enthält. Damit benötigt der Algorithmus keinen zusätzlichen Zeit- oder Speicheraufwand, hat also auch die Laufzeit $\mathcal{O}\big(n^3|G|\big)$.

Aufgabe 2 (5 Punkte)

Sei eine Grammatik $G = (\Sigma, V, S, R)$ durch $V = \{S, A, B, C, E, F\}, \Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und folgende

Regelmenge R gegeben:

$$
\begin{aligned}
S &\to B \mid C\mathsf{a}C \\
A &\to \mathsf{a} \mid D \\
B &\to CC \mid E \mid \mathsf{a}C\mathsf{b} \\
C &\to \mathsf{c} \mid A \mid BC \mid \varepsilon \\
D &\to EC \\
E &\to \mathsf{b} \mid A
\end{aligned}
$$

Verwenden Sie das Verfahren aus der Vorlesung, um G in eine äquivalente Grammatik in erweiterter Chomsky-Normalform zu überführen. Es genügt dabei, wenn sie nach jedem Schritt bzw. jeder Phase das jeweilige Ergebnis angeben.

Lösung:
Schritt 1: Ersetze gemischte Produktionen mit Terminal- und Nichtterminalsymbolen auf der rechten Seite.

$$
\begin{aligned}
S &\to B \mid CY_{\mathsf{a}}C \\
A &\to \mathsf{a} \mid D \\
B &\to CC \mid E \mid Y_{\mathsf{a}}CY_{\mathsf{b}} \\
C &\to \mathsf{c} \mid A \mid BC \mid \varepsilon \\
D &\to EC \\
E &\to \mathsf{b} \mid A \\
Y_a &\to \mathsf{a} \\
Y_b &\to \mathsf{b}
\end{aligned}
$$

Schritt 2: Ersetze Produktionen mit Länge ≥ 3.

$$
\begin{aligned}
S &\to B \mid CZ_1 \\
A &\to \mathsf{a} \mid D \\
B &\to CC \mid E \mid Y_{\mathsf{a}}Z_2 \\
C &\to \mathsf{c} \mid A \mid BC \mid \varepsilon \\
D &\to EC \\
E &\to \mathsf{b} \mid A \\
Y_a &\to \mathsf{a} \\
Y_b &\to \mathsf{b} \\
Z_1 &\to Y_{\mathsf{a}}C \\
Z_2 &\to CY_{\mathsf{b}}
\end{aligned}
$$

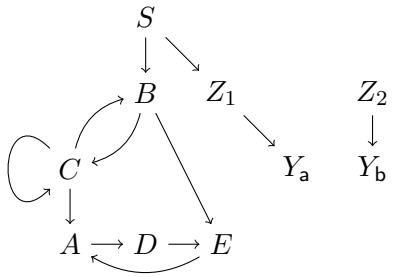
Schritt 3: Ersetze Regeln der Form $A \to \varepsilon$.

Die Menge der Variablen, die sich auf ε ableiten lassen, ist $V' = \{S, B, C\}$. Das folgt aus der Ableitungskette $S \to B \to CC \to \varepsilon$.

$$
\begin{aligned}
S &\to B \mid CZ_1 \mid Z_1 \\
A &\to \mathsf{a} \mid D \\
B &\to CC \mid C \mid E \mid Y_{\mathsf{a}}Z_2 \\
C &\to \mathsf{c} \mid A \mid B \mid C \mid BC \\
D &\to EC \mid E \\
E &\to \mathsf{b} \mid A \\
Y_a &\to \mathsf{a} \\
Y_b &\to \mathsf{b} \\
Z_1 &\to Y_{\mathsf{a}}C \mid Y_{\mathsf{a}} \\
Z_2 &\to CY_{\mathsf{b}} \mid Y_{\mathsf{b}}
\end{aligned}
$$

Schritt 4: Ersetze Kettenregeln der Form $A \to B$.

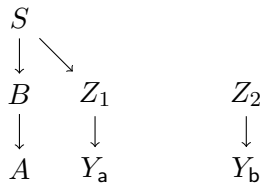
Abhängigkeitsgraph:



Es existieren drei zyklische Abhängigkeiten $A \to D \to E \to A$, $B \to C \to B$ und $C \to C$. Wir entfernen also D und E und ersetzen alle Vorkommen davon durch A. Außerdem entfernen wir C und ersetzen alle Vorkommen davon durch B. Im Anschluss entfernen wir alle Regeln der Form $X \to X$.

$$
\begin{aligned}
S &\to B \mid BZ_1 \mid Z_1 \\
A &\to \mathsf{a} \mid AB \mid \mathsf{b} \\
B &\to BB \mid A \mid Y_{\mathsf{a}}Z_2 \mid \mathsf{c} \\
Y_a &\to \mathsf{a} \\
Y_b &\to \mathsf{b} \\
Z_1 &\to Y_{\mathsf{a}}B \mid Y_{\mathsf{a}} \\
Z_2 &\to BY_{\mathsf{b}} \mid Y_{\mathsf{b}}
\end{aligned}
$$

Es entsteht folgender neuer Abhängigkeitsgraph:



Eine mögliche topologische Sortierung ist $S, B, A, Z_1, Z_2, Y_\mathsf{a}, Y_\mathsf{b}$.

$$
\begin{aligned}
S &\to BB \mid \mathsf{a} \mid AB \mid \mathsf{b} \mid Y_\mathsf{a}Z_2 \mid \mathsf{c} \mid BZ_1 \mid Y_\mathsf{a}B \\
A &\to \mathsf{a} \mid AB \mid \mathsf{b} \\
B &\to BB \mid \mathsf{a} \mid AB \mid \mathsf{b} \mid Y_\mathsf{a}Z_2 \mid \mathsf{c} \\
Y_a &\to \mathsf{a} \\
Y_b &\to \mathsf{b} \\
Z_1 &\to Y_\mathsf{a}B \mid \mathsf{a} \\
Z_2 &\to BY_\mathsf{b} \mid \mathsf{b}
\end{aligned}
$$

Schritt 5: Stelle die Ableitung $S \overset{*}{\to} \varepsilon$ wieder her.

Das Endergebnis ist eine Grammatik $G' = (\Sigma, V', S', R')$ mit $V' = \{S', S, A, B, Y_\mathsf{a}, Y_\mathsf{b}, Z_1, Z_2\}, \Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und Regelmenge R':

$$
\begin{aligned}
S' &\to S \mid \varepsilon \\
S &\to BB \mid \mathsf{a} \mid AB \mid \mathsf{b} \mid Y_\mathsf{a}Z_2 \mid \mathsf{c} \mid BZ_1 \mid Y_\mathsf{a}B \\
A &\to \mathsf{a} \mid AB \mid \mathsf{b} \\
B &\to BB \mid \mathsf{a} \mid AB \mid \mathsf{b} \mid Y_\mathsf{a}Z_2 \mid \mathsf{c} \\
Y_a &\to \mathsf{a} \\
Y_b &\to \mathsf{b} \\
Z_1 &\to Y_\mathsf{a}B \mid \mathsf{a} \\
Z_2 &\to BY_\mathsf{b} \mid \mathsf{b}
\end{aligned}
$$

Aufgabe 3

(2 + 2 = 4 Punkte)

Sei $G = (\Sigma, S, V, R)$ eine kontextfreie Grammatik in Chomsky-Normalform.

(a) Sei $w \in L(G)$. Wie viele Ableitungsschritte sind nötig, um w aus S abzuleiten? Beweisen Sie.

(b) Sei $L(G)$ nun eine endliche Sprache. Wie lang kann ein Wort $w \in L(G)$ höchstens sein? Beweisen Sie eine möglichst scharfe obere Schranke in Abhängigkeit von G.

Lösung:

(a) Jedes Wort $w \in L(G)$ benötigt genau $2|w| - 1$ Ableitungsschritte.

Da die Grammatik in Chomsky-Normalform ist, gibt es für jeden Ableitungsschritt nur zwei Möglichkeiten:

(1) Ein Nichtterminalsymbol wird durch zwei Nichtterminalsymbole ersetzt ($A \to BC$).

(2) Ein Nichtterminalsymbol wird durch ein Terminalsymbol ersetzt ($A \to \mathsf{a}$).

Um die Terminalsymbole in w abzuleiten, hat man genau $|w|$ Ableitungsschritte von Typ (2). Jeder dieser Ableitungsschritte entfernt dabei genau ein Nichtterminalsymbol. Ableitungsschritte von Typ (1) erhöhen die Anzahl der Nichtterminalsymbole um genau 1. Insgesamt benötigt man also $|w|-1$ Ableitungsschritte, um $|w|$ Nichtterminale abzuleiten (da das Startsymbol bereits ein Nichtterminalsymbol ist) und $|w|$ Ableitungsschritte, um $|w|$ Terminale abzuleiten.

(b) Das längste Wort ist höchstens $2^{|V|-1}$ Zeichen lang. Wir zeigen die Behauptung durch vollständige Induktion über die Anzahl der Nichtterminalsymbole $|V|$ in G.

Für $|V| = 1$ ist S das einzige Nichtterminal und es gibt nur Ableitungen der Form $S \to a$ mit $a \in \Sigma$, da $L(G)$ endlich ist. Das längste Wort ist also $1 = 2^0 = 2^{|V|-1}$ lang.

Für $|V| > 1$ nehmen wir an, dass es keine nutzlosen Variablen gibt. Sei der erste Ableitungsschritt von der Form $S \to AB$ mit $A, B \in V$. Da $L(G)$ endlich ist, ist S nicht von A oder B aus durch Ableitungen erreichbar (d.h. A und B können auf nichts abgeleitet werden, was wieder das Startsymbol enthält). Wir können also die Grammatik G' betrachten, die A als Startsymbol hat und in der S nutzlos ist, also eine Variable weniger hat. Nach Induktionsvoraussetzung ist das längste Wort, das sich aus A in G' ableiten lässt, höchstens $2^{|V|-2}$ Zeichen lang. Analog gilt das gleiche für das längste Wort, das sich aus B ableiten lässt. Damit ist das längste Wort, das sich aus S ableiten lässt, höchstens $2^{|V|-2} + 2^{|V|-2} = 2^{|V|-1}$ Zeichen lang.

Aufgabe 4

(2 + 3 = 5 Punkte)

Geben Sie für die folgenden Sprachen jeweils eine Grammatik mit maximalem Chomsky-Typ an. Erläutern Sie die Idee der von Ihnen angegebenen Regeln.

(a) $L_1 = \{w \mid w \in \{\mathsf{a}, \mathsf{b}\}^*,\ w \neq w^R\}$
Dabei bezeichnet w^R das Spiegelwort von w.

(b) $L_2 = \{w{:}w \mid w \in \{\mathsf{a}, \mathsf{b}\}^*\} \subseteq \{\mathsf{a}, \mathsf{b}, :\}^*$

Lösung:

(a) L_1 ist kontextfrei. $G_1 = (\Sigma, V, S, R)$ mit $\Sigma = \{\mathsf{a}, \mathsf{b}\}$, $V = \{S, A\}$ und

$$\begin{array}{rll} R = \{S & \to \mathsf{a}S\mathsf{a} \mid \mathsf{b}S\mathsf{b} \mid \mathsf{a}A\mathsf{b} \mid \mathsf{b}A\mathsf{a} & \text{mindestens eine Stelle, an der sich } w \text{ und } w^R \text{ unterscheiden} \\ A & \to \mathsf{a}A \mid \mathsf{b}A \mid \varepsilon\} & \text{Rest beliebig} \end{array}$$

(b) L_2 ist kontextsensitiv. Die Idee ist, erst das Wort w hinter dem Doppelpunkt zu erzeugen, wobei aber jedes Zeichen in w doppelt erzeugt wird (einmal als A bzw. B und einmal als A' bzw. B'). Danach werden die A' bzw. die B' nach links bis vor den Doppelpunkt geschoben. Wichtig ist, dass diese sich nicht „überholen“ können, die Reihenfolge also gleich bleibt.

$G_2 = (\Sigma, V, S, R)$ mit $\Sigma = \{\mathsf{a}, \mathsf{b}, :\}$, $V = \{S, S', A, A', B, B', X\}$ und

$R = \{S$	$\to XS' \mid X$	Trennzeichen erzeugen
S'	$\to S'A'A \mid S'B'B \mid A'A \mid B'B$	Wort w erzeugen, jedes Zeichen doppelt
XA'	$\to \mathsf{a}X$	bis zum Trennzeichen geschoben,
XB'	$\to \mathsf{b}X$	dann Terminal vor Trennzeichen
AA'	$\to A'A$	kopiertes Zeichen nach links schieben
AB'	$\to B'A$	
BB'	$\to B'B$	
BA'	$\to A'B$	
A	$\to \mathsf{a}$	Terminale ableiten
B	$\to \mathsf{b}$	
X	$\to :\}$	

Aufgabe 5 (2 + 2 = 4 Punkte)

Ein Roboter R bewegt sich auf dem unendlichen zweidimensionalen Gitter $\mathbb{Z} \times \mathbb{Z}$. Dieser kann vom Ursprung $(0, 0)$ aus nach oben (O), unten (U), links (L) oder rechts (R) gesteuert werden. Wir können also einen Pfad des Roboters als Wort über dem Alphabet $\Sigma = \{\mathsf{O}, \mathsf{U}, \mathsf{L}, \mathsf{R}\}$ auffassen.

Geben Sie jeweils für die folgenden Sprachen eine kontextfreie Grammatik (mit kurzer Erklärung) an oder zeigen Sie, dass die Sprache nicht kontextfrei ist.

(a) $L_1 = \{w \in \Sigma^* \mid \text{Pfad } w \text{ endet im Ursprung}\}$

(b) $L_2 = \{w \in \Sigma^* \mid \text{Pfad } w \text{ befindet sich zu keinem Zeitpunkt unterhalb der } x\text{-Achse}\}$

Lösung:

(a) L_1 ist nicht kontextfrei. Wir geben zunächst eine formale Definition der Sprache an:

$$L_1 = \left\{ w \in \Sigma^* \;\middle|\; |w|_{\mathsf{O}} = |w|_{\mathsf{U}} \text{ und } |w|_{\mathsf{L}} = |w|_{\mathsf{R}} \right\}$$

Nehme an, dass L_1 kontextfrei ist. Dann existiert nach dem Pumping-Lemma ein $n \in \mathbb{N}$, sodass für jedes Wort z mit $|z| \geq n$ eine Zerlegung $z = uvwxy$ mit $|vx| \geq 1$ und $|vwx| \leq n$, so dass $uv^iwx^iy \in L_1$ für alle $i \geq 0$. Wähle $z = \mathsf{O}^n\mathsf{L}^n\mathsf{U}^n\mathsf{R}^n$. Wegen $|vwx| \leq n$ kann vx höchstens 2 unterschiedliche Symbole enthalten. Dann kann vx weder gleichzeitig O und U noch gleichzeitig L und R enthalten. Das Wort uv^2wx^2y enthält also nicht gleich viele O und U oder nicht gleich viele L und R, liegt also nicht in L_1, was mit dem Pumping-Lemma ein Widerspruch zur Annahme, dass L_1 kontextfrei sei, ergibt.

(b) Die Sprache ist kontextfrei. Jedes Präfix des Pfades muss mehr Os als Us enthalten. Für jedes U, das im Pfad enthalten ist, muss also bereits schon vorher ein O vorkommen.
$G = (\Sigma, V, S, R)$ mit $V = \{S\}$ und $R = \{S \to SS \mid \mathsf{O}S\mathsf{U} \mid \mathsf{O} \mid \mathsf{L} \mid \mathsf{R} \mid \varepsilon\}$

Aufgabe 6 (2 + 1 + 2 + 2 = 7 Punkte)

Seien Σ und Γ zwei Alphabete. Eine Abbildung $h\colon \Sigma^* \to \Gamma^*$ ist ein Homomorphismus, wenn für alle Worte $u, v \in \Sigma^*$ gilt: $h(uv) = h(u)h(v)$. Beachten Sie, dass ein Homomorphismus vollständig durch die Bilder $h(a)$ für alle $a \in \Sigma$ festgelegt ist und dass $h(\varepsilon) = \varepsilon$ gilt.

(a) Zeigen Sie, dass die Klasse der kontextfreien Sprachen unter Homomorphismen abgeschlossen ist, d.h. für jede kontextfreie Sprache L und jeden Homomorphismus h gilt: $h(L) = \{h(w) \mid w \in L\}$ ist auch kontextfrei.

Beachten Sie, dass die Alphabete Σ und Γ nicht disjunkt sein müssen.

(b) Gilt auch die Umkehrung? Also ist L kontextfrei, wenn $h(L)$ kontextfrei und h ein Homomorphismus ist? Begründen Sie.

Im Folgenden sollen Sie zeigen, dass die Klasse der kontextsensitive Sprachen nicht unter Homomorphismen abgeschlossen ist. Betrachten Sie dazu eine beliebige Grammatik $G = (\Sigma, S, V, R)$ und die Sprache

$$L_G = \{w_1 \# w_2 \# \dots \# w_k \mid w_1 = S,\ w_k \in \Sigma^* \text{ und } w_i \to_G w_{i+1} \text{ für } i = 1, \dots, k-1\}$$

Die Sprache L_G beschreibt genau die Menge aller in G erlaubten Folgen von Wortableitungen.

(c) Zeigen Sie, dass L_G eine kontextsensitive Sprache ist, indem Sie eine linear beschränkte TM angeben, die L_G entscheidet.

Nun erweitern wir das Alphabet um neue Symbole $\widetilde{\Sigma} = \{\widetilde{a} \mid a \in \Sigma\}$ und definieren für ein Wort $w = a_1 \dots a_n \in \Sigma^*$ das Wort $\widetilde{w} = \widetilde{a_1} \dots \widetilde{a_n} \in \widetilde{\Sigma}^*$. Damit modifizieren wir die Sprache L_G zu:

$$\widetilde{L_G} = \{w_1 \# w_2 \# \dots \# \widetilde{w_k} \mid w_1 \# w_2 \# \dots \# w_k \in L_G\}$$

Offensichtlich ist $\widetilde{L_G}$ wie auch L_G kontextsensitiv.

(d) Folgern Sie, dass es eine kontextsensitive Sprache L' und einen Homomorphismus h gibt, sodass $h(L')$ nicht kontextsensitiv ist.

Lösung:

(a) Sei L eine kontextfreie Sprache über dem Alphabet Σ und $G = (\Sigma, S, V, R)$ eine Typ-2-Grammatik, die L erzeugt. Sei $h\colon \Sigma^* \to \Gamma^*$ außerdem ein Homomorphismus. Wir zeigen, dass $h(L)$ eine kontextfreie Sprache ist, indem wir eine Typ-2-Grammatik G' für $h(L)$ angeben. Dafür führen wir zusätzlich zu V für jedes Terminal $a \in \Sigma$ eine neue Variable V_a ein und ersetzen jedes Terminal $a \in \Sigma$ in den Produktionsregeln durch die neue Variable V_a. Außerdem erweitern wir die Produktionsregeln von G, indem wir $|\Sigma|$ Produktionen hinzufügen, die jede neue Variable V_a auf $h(a) \in \Gamma$ ableiten. Ein Wort $h(a_1 \dots a_n) \in h(L)$ mit $a_i \in \Sigma$ wird also abgeleitet, indem erst $V_{a_1} \dots V_{a_n}$ vom Startsymbol S abgeleitet wird und jede Variable V_a durch $h(a)$ ersetzt wird.

(b) Die Umkehrung gilt nicht. Die triviale Abbildung h mit $h(x) = \varepsilon$ für alle $x \in \Sigma^*$ ist ein Homomorphismus. Es gilt also $h(L) = \{\varepsilon\}$ (und damit ist $h(L)$ kontextfrei) für jede beliebige Sprache $L \subseteq \Sigma^*$, insbesondere auch für eine nicht-kontextfreie Sprache L.

(c) Für eine gegebene Grammatik G ist L_G kontextsensitiv, da wir eine Turingmaschine M mit linear beschränktem Band angeben können, die L_G entscheidet. Für eine Eingabe $w = w_1\#w_2\#\ldots\#w_k$ muss die Turingmaschine M für alle $i = 1, \ldots, k-1$ prüfen, ob w_{i+1} sich mit einer Produktionsregel in R aus w_i ableiten lässt. Dazu kann sie in jedem Schritt alle möglichen Produktionsregeln auf w_i anwenden und prüfen, ob das Wort nach dem Ableitungsschritt w_{i+1} entspricht. Gibt es keine solche Produktionsregel, ist $w \notin L_G$. Zuletzt muss M noch prüfen, dass $w_1 = S$ und $w_k \in \Sigma^*$ gilt.

(d) Die Idee ist nun, jede Ableitungsfolge $w_1\#w_2\#\ldots\#w_k$ mit einem geeigneten Homomorphismus h abzubilden, sodass nur noch das abgeleitete Wort $w_k \in L(G)$ übrigbleibt. Da G beliebig ist, können wir insbesondere G so wählen, dass $L(G)$ nicht kontextsensitiv ist. Damit hätten wir einen Homomorphismus h gefunden, der eine kontextsensitive Sprache L_G auf eine nicht-kontextsensitive Sprache $L(G)$ abbildet.

Allerdings müssen wir noch zwischen dem abgeleiteten Wort w_k und den anderen w_i unterscheiden. Das können wir erreichen, indem wir $\widetilde{L_G}$ verwenden statt L_G. Für $\widetilde{L_G}$ ist es nun einfach, einen Homomorphismus $h : (\Sigma \cup \widetilde{\Sigma} \cup \{\#\})^* \to \Sigma^*$ anzugeben, der das Gewünschte tut. Für $a \in \Sigma \cup \{\#\}$ definieren wir $h(a) = \varepsilon$ und $h(\widetilde{a}) = a$ für $\widetilde{a} \in \widetilde{\Sigma}$. Dann gilt: $h(w_1\#w_2\#\ldots\#\widetilde{w_k}) = w_k$ und $h(\widetilde{L_G}) = L(G)$.