

Einführung in die theoretische Informatik
Sommersemester 2019 – Übungsblatt Lösungsskizze 6

AUFGABE 6.1. (*Wichtige Begriffe*)

Stufe A

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

- Chomsky-Normalform
- mehrdeutig
- inhärent mehrdeutig
- CYK-Algorithmus
- Pumping-Lemma für kontextfreie Sprachen

AUFGABE 6.2. (*Chomsky-Normalform üben*)

Stufe B

Rufen Sie die Website <http://grammar.epfl.ch/> auf und üben Sie das Überführen von Grammatiken in Chomsky-Normalform.

Beachten Sie dabei dass die Website Feedback gibt, ob Ihre Lösung richtig oder falsch ist, aber nicht überprüft ob Sie das Verfahren aus der Vorlesung verwendet haben.

AUFGABE 6.3. (*Mehrdeutigkeit*)

Stufe B

Betrachten Sie die folgende Grammatik G:

$$\begin{aligned} S &\rightarrow N-P V-P \\ N-P &\rightarrow C-N \mid C-N P-P \\ V-P &\rightarrow C-V \mid C-V P-P \\ P-P &\rightarrow P C-N \\ C-N &\rightarrow A N \\ C-V &\rightarrow V \mid V N-P \\ A &\rightarrow a \mid the \\ N &\rightarrow girl \mid boy \mid flower \\ V &\rightarrow touches \mid likes \mid sees \\ P &\rightarrow with \end{aligned}$$

- (a) Zeigen Sie, dass die Grammatik mehrdeutig ist, indem Sie zwei verschiedene Ableitungen für das folgende Wort angeben:

a girl touches a boy with a flower

- (b) Zeigen Sie, dass die Sprache $L(G)$ endlich ist.
(c) Zeigen Sie, dass es keine endliche Sprache gibt, die inhärent mehrdeutig ist.
(d) Für Programmiersprachen sind mehrdeutige Grammatiken unerwünscht, da jedes Programm nur eine einzige, eindeutige Bedeutung haben soll. Diskutieren Sie anhand den verschiedenen Bedeutungsmöglichkeiten des in Aufgabenteil (a) angegebenen Satzes, warum man natürliche Sprache nur mit mehrdeutigen Grammatiken sinnvoll repräsentieren kann.

Lösungsskizze

- (a) (i) $S \rightarrow N-P V-P \rightarrow C-N V-P \rightarrow A N V-P \rightarrow^* a \text{ girl } V-P \rightarrow a \text{ girl } C-V P-P \rightarrow a \text{ girl } C-V P C-N \rightarrow a \text{ girl } C-V \text{ with } C-N \rightarrow a \text{ girl } C-V \text{ with } A N \rightarrow^* a \text{ girl } C-V \text{ with } a \text{ flower} \rightarrow a \text{ girl } V N-P \text{ with } a \text{ flower} \rightarrow a \text{ girl touches } N-P \text{ with } a \text{ flower} \rightarrow a \text{ girl touches } C-N \text{ with } a \text{ flower} \rightarrow a \text{ girl touches } A N \text{ with } a \text{ flower} \rightarrow^* a \text{ girl touches } a \text{ boy with } a \text{ flower}$
(ii) $S \rightarrow N-P V-P \rightarrow C-N V-P \rightarrow A N V-P \rightarrow^* a \text{ girl } V-P \rightarrow a \text{ girl } C-V \rightarrow a \text{ girl } V N-P \rightarrow a \text{ girl touches } N-P \rightarrow a \text{ girl touches } C-N P-P \rightarrow a \text{ girl touches } A N P-P \rightarrow^* a \text{ girl touches } a \text{ boy } P-P \rightarrow a \text{ girl touches } a \text{ boy } P C-N \rightarrow a \text{ girl touches } a \text{ boy } P A N \rightarrow^* a \text{ girl touches } a \text{ boy with } a \text{ flower}$
- (b) Der Graph (V, E) , wobei die Knotenmenge V gerade die Menge aller Nicht-Terminale ist und $E := \{(V, V') \mid \exists V \rightarrow wV'w'\}$, ist azyklisch.
- (c) Sei L eine endliche Sprache, d.h. $|L| = n$ für ein $n \in \mathbb{N}$. Die Grammatik $G = (\{S\}, \Sigma, \{S \rightarrow w \mid w \in L\}, S)$ erkennt dann genau L . Jedes Wort aus L hat somit eine eindeutige Ableitung und die Grammatik ist somit nicht inhärent mehrdeutig.
- (d) Der Satz auf Aufgabenteil (a) hat zwei Bedeutungen: In der ersten Bedeutung berührt ein Mädchen einen Jungen mit einer Blume. In der anderen berührt ein Mädchen einen Jungen, der eine Blume hält. Obwohl beide Bedeutungen durch zwei syntaktisch absolut identische Sätze dargestellt werden, haben sie unterschiedliche Semantik. Möchte man mit einer Grammatik diese beiden unterschiedlichen Bedeutungen repräsentieren, so

darf man diesen beiden Bedeutungen nicht die gleiche Ableitung zuweisen. Damit ist die Eindeutigkeit, um natürliche Sprachen darzustellen, nicht erwünschenswert. Mehrdeutigkeit ist an dieser Stelle notwendig, um die Mehrdeutigkeit der Sprache darzustellen.

Stufe C

AUFGABE 6.4. (*Chomsky-Normalform*)

Die CFG G bestehe aus folgenden Produktionen über dem Alphabet $\Sigma = \{a, b\}$:

$$\begin{aligned} S &\rightarrow ASA \mid aB \\ A &\rightarrow B \mid S \mid CB \\ B &\rightarrow b \mid \varepsilon \\ C &\rightarrow aC \\ D &\rightarrow aSCb \mid a \end{aligned}$$

- Beschreiben Sie in eigenen Worten, wann ein Nichtterminal *nützlich* in einer Grammatik ist.
- Reduzieren Sie die Grammatik G auf die nützlichen Nichtterminale.
- Überführen Sie die reduzierte Grammatik dann in Chomsky-Normalform.
- Erklären Sie in eigenen Worten, wie Sie nach Überführen einer Grammatik in CNF überprüfen können, dass Sie keine Fehler gemacht haben.¹

Lösungsskizze

- Ein Nichtterminal ist nützlich, wenn es eine Ableitung vom Startsymbol zu einem Wort (also einer Folge von Terminalen) gibt, in der das Nichtterminal verwendet wird. Nützliche Symbole sind stets erzeugend und erreichbar. Erzeugend bedeutet, dass von dem Nichtterminal ausgehend ein Wort produziert werden kann; erreichbar bedeutet, dass vom Startsymbol ausgehend das Nichtterminal produziert werden kann. Der Begriff ist relevant, da ähnlich wie bei Automaten angenommen wird, dass alle Zustände erreichbar sind, man bei Grammatiken annehmen möchte, dass sie nur nützliche Nichtterminale enthalten. Deswegen benötigt man eine klare Definition und eine Möglichkeit, eine Grammatik, die die Anforderung nicht erfüllt, in eine mit nur nützlichen Nichtterminalen zu überführen.

(b) *Erzeugende Variablen:*

- Intuition: C kann kein Wort produzieren, da eine Ableitung, die ein C enthält, niemals "terminiert", also nie nur Terminale enthält.
- Formal bauen wir iterativ die Menge aller Nichtterminale, die eine Möglichkeit haben, ein Wort aus nur Terminalen zu erzeugen.
- $P_0 := \{X \mid \exists(X, \gamma) \in P. \gamma \in \Sigma^*\}$
- $P_{k+1} := P_k \cup \{X \in V \mid \exists(X, \gamma) \in P. \forall Y \in V. (|\gamma|_Y > 0 \rightarrow Y \in P_k)\}$ bis $P_{k+1} = P_k$.
- Führt auf:

$$P_0 = \{B, D\} \quad P_1 = \{B, D, A, S\} = P_2$$

Damit kann C samt $C \rightarrow aC$, $A \rightarrow CB$ und $D \rightarrow aSCb$ entfernt werden.

Erreichbare Variablen:

- Intuition: D kann von S aus nicht erreicht werden.
- Formal bauen wir die Menge aller Nichtterminale, die von S aus erzeugt werden können.
- $R_0 := \{S\}$
- $R_{k+1} := R_k \cup \{Y \in V \mid \exists(X, \gamma) \in P. X \in R_k \wedge |\gamma|_Y > 0\}$ bis $R_{k+1} = R_k$.
- Führt auf:

$$R_0 = \{S\} \quad R_1 = \{S, A, B\} = R_2$$

Damit kann D und $D \rightarrow a$ entfernt werden.

Ab jetzt steht G für die reduzierte (bereinigte) Grammatik:

$$S \rightarrow ASA \mid aB \quad A \rightarrow B \mid S \quad B \rightarrow b \mid \varepsilon$$

(c) *Überführung in CNF:*

- In jeder Regel (X, γ) mit $|\gamma| \geq 2$ Ersetzen jedes Vorkommen eines Terminals x durch X_x und Ergänzen der benötigten Produktionen $X_x \rightarrow x$:

$$G': S \rightarrow ASA \mid X_a B \quad A \rightarrow B \mid S \quad B \rightarrow b \mid \varepsilon \quad X_a \rightarrow a$$

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

- 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 $X_{YZ} \rightarrow YZ$ zugewiesen bekommt. Damit:

$$G'': S \rightarrow AX_{SA} \mid X_a B \quad A \rightarrow B \mid S \quad B \rightarrow b \mid \varepsilon \quad X_a \rightarrow a \quad X_{SA} \rightarrow SA$$

¹Hier 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.

Erkennen von ε -Regeln:

- $E_0 := \{X \in V \mid (X, \varepsilon) \in P\}$
- $E_{k+1} := E_k \cup \{X \in V \mid \exists (X, \gamma) \in P. \gamma \in E_k^*\}$ bis $E_{k+1} = E_k$.

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

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

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

- Entfernen aller ε -Produktionen:

$$G''': S \rightarrow AX_{SA} \mid X_{SA} \mid X_a B \mid X_a \quad A \rightarrow B \mid S \quad B \rightarrow b \quad X_a \rightarrow a \quad X_{SA} \rightarrow SA \mid S$$

Zusammenziehen von Kettenproduktionen:

- $T_0 := \{(X, Y) \in P \cap V \times V\}$
- $T_{k+1} := T_k \cup \{(X, Y) \in V \times V \mid \exists Z \in V. (X, Z) \in T_k \wedge (Z, Y) \in T_k\}$ bis $T_{k+1} = T_k =: T_*$ (einfach transitiver Abschluss über durch Kettenproduktionen gegebener Kantenrelation auf V).

$$\begin{array}{l} T_0 = \{(S, X_{SA}), (S, X_a), (A, B), (A, S), (X_{SA}, S)\} \\ T_1 = T_0 \cup \{(S, S), (A, X_{SA}), (A, X_a), (X_{SA}, X_{SA}), (X_{SA}, X_a)\} = T_2 \end{array}$$

- Dann (1) Entfernen aller Kettenproduktion und (2) anschließend, falls $(X, Y) \in T_*$, füge (X, γ) zu P hinzu für jede Regel $(Y, \gamma) \in P$.

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

- (d)
- Prüfen, dass alle Produktionen die Form $A \rightarrow BC$ oder $A \rightarrow a$ 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.

AUFGABE 6.5. (CYK-Algorithmus)

Wir betrachten die Grammatik $G = (\{S, T, U, A, B, C\}, \{a, b, c\}, P, S)$ in CNF mit den folgenden Produktionen P :

$$\begin{aligned} S &\rightarrow TS \mid CT \mid a & A &\rightarrow a \\ T &\rightarrow AU \mid TT \mid c & B &\rightarrow b \\ U &\rightarrow SB \mid AB & C &\rightarrow c \end{aligned}$$

Bestimmen Sie mit dem CYK-Algorithmus, ob $ccaab \in L(G)$ und $aabcc \in L(G)$. Geben Sie dabei auch die berechneten Tabellen an.

Lösungsskizze

Nach dem CYK-Algorithmus ergeben sich folgende Berechnungstabellen:

15 S, T				
14		25 S, T		
13 S	24	35 T		
12 S, T	23 S	34	45 U	
11 C, T	22 C, T	33 S, A	44 S, A	55 B
c	c	a	a	b

15 S, T				
14 T		25		
13 T	24	35		
12	23 U	34	45 S, T	
11 S, A	22 S, A	33 B	44 C, T	55 C, T
a	a	b	c	c

Also ist $ccaab \in L(G)$ und $aabcc \in L(G)$.

AUFGABE 6.6. (Pumping-Lemma für Kontextfreie Sprachen)

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:
 - (i) Pfade, die in den Ursprung zurückkehren.
 - (ii) 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

- (a) (i)

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

- (ii)

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

- (b) (i)
 - 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 .
 - Sei zusätzlich $z = \rightarrow^n \uparrow^n \leftarrow^n \downarrow^n$, d.h., $z \in L_a$ 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:

$$(1) \ vx \neq \varepsilon \quad (2) \ |vwx| \leq n \quad (3) \ \forall i \in \mathbb{N}_0. \ uv^iwx^iy \in L_a$$

- 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.
- (ii) • 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 ist $z = \rightarrow^n \uparrow^{n+1} \leftarrow^{n+2}$, d.h., $z \in L_b$ 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:

$$(1) \quad vx \neq \varepsilon \quad (2) \quad |vwx| \leq n \quad (3) \quad \forall i \in \mathbb{N}_0. \quad uv^iwx^iy \in L_b$$

- 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 $vw \in L(\uparrow^*)$, dann können wir die Anzahl der \uparrow über die Anzahl der \leftarrow pumpen, enthält vw mindestens ein \rightarrow und damit kein \leftarrow , so kann man die Anzahl der \rightarrow beliebig groß, insbesondere größer als die Anzahl der \leftarrow machen. Besteht vw 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.

AUFGABE 6.7.

Stufe D

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$. L ist *präfix-abgeschlossen*, wenn für alle $w \in L$ und $v \in \Sigma^*$ mit $v \preceq w$ (also v ist ein Präfix von w) gilt: $v \in L$.

Beweisen Sie, dass jede unendliche, präfix-abgeschlossene, kontextfreie Sprache eine unendliche, reguläre Teilmenge enthält.

Lösungsskizze

Sei L eine unendliche, präfix-abgeschlossene, kontextfreie Sprache. Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L . Da L unendlich ist, gibt es ein $z \in L$ mit $|z| \geq n$. Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1) \ vx \neq \varepsilon \quad (2) \ |vwx| \leq n \quad (3) \ \forall i \in \mathbb{N}_0. \ uv^iwx^iy \in L$$

Wir unterscheiden nun zwei Fälle:

- Angenommen sei $v \neq \varepsilon$. Dann definieren wir nun die reguläre Sprache R_1 :

$$R_1 = L(uv^*) = \{uv^i \mid i \in \mathbb{N}_0\}$$

R_1 ist unendlich und regulär. Wir haben dann $R_1 \subseteq L$, wegen der Präfix-Abgeschlossenheit und (3).

- Angenommen sei $v = \varepsilon$. Aus (1) folgt dann $x \neq \varepsilon$. Wir definieren nun die reguläre Sprache R_2 :

$$R_2 = L(uwx^*) = \{uwx^i \mid i \in \mathbb{N}_0\}$$

R_2 ist unendlich und regulär. Wir haben dann $R_2 \subseteq L$, wegen der Präfix-Abgeschlossenheit und (3).

AUFGABE 6.8. (Abschlusseigenschaften kontextfreier Sprachen)

Stufe D

Beweisen Sie die folgenden beiden Aussagen:

- (a) Der Präfixabschluss $L_{pre} := \{u \mid \exists v. uv \in L\}$ einer kontextfreien Sprache $L \subseteq \Sigma^*$ über einem Alphabet Σ ist wieder kontextfrei. Geben Sie hierzu ein Übersetzung an, die die Grammatik G für L in eine Grammatik G' umwandelt, sodass $L_{pre} = L(G')$ übersetzt.
- (b) Kontextfreie Sprachen sind unter Schnitt mit regulären Sprachen abgeschlossen.

Lösungsskizze

- (a) Sei $G = (V, \Sigma, P, S)$ CFG in CNF mit $L = L(G)$ und $L' := \{u \in \Sigma^* \mid \exists v \in \Sigma^*. uv \in L(G)\}$.

Idee:

Es gilt: $u \in L'$ gdw. es gibt einen Ableitungsbaum zu uv bzgl. G für ein $v \in \Sigma^*$.

Betrachte in diesem Ableitungsbaum zu uv den eindeutigen Pfad vom letzten Zeichen von u zur Wurzel. Wir erweitern G so zu G' , dass G' diesen Pfad raten kann und die Teilbäume, die Zeichen von v erzeugen, weglassen kann.

Konstruiere $G' = (V \uplus V' \uplus \{S''\}, \Sigma, P \uplus P', S'')$ hierzu wie folgt:

- $V' = \{X' \mid X \in V\}$ – die Markierung rät den Pfad von der Wurzel zum letzten Zeichen von u und steht immer am rechten NT einer Ableitung bzgl. G'
- $P' = \{X' \rightarrow YZ', X' \rightarrow Y' \mid X \rightarrow YZ\} \cup \{X' \rightarrow a \mid X \rightarrow a \in P\} \cup \{S'' \rightarrow S' \mid \varepsilon\}$ erlaubt G' zu raten, ob das letzte Zeichen von u durch Y oder Z erzeugt wird. Falls es durch Y erzeugt wird, wird der durch Z erzeugte Suffix fallengelassen (Fall $X' \rightarrow Y'$).

Behauptung:

$$L(G') = L' := \{u \in \Sigma^* \mid \exists v \in \Sigma^*. uv \in L(G)\}.$$

Beweis:

Sei $u \in L'$. Betrachte entsprechend der obigen Beschreibung einen Ableitungsbaum zu $uv \in L(G)$ bzgl. G . Markiere die Knoten entlang des Pfades vom letzten Zeichen von u bis zur Wurzel. Schneide die Teilbäume zu v ab. Nach Konstruktion erhält man einen Ableitungsbaum zu u bzgl. G' .

Sei $u \in L(G')$. Betrachte Ableitungsbaum zu u bzgl. G' . Für jeden inneren Knoten, der (1) kein Terminal erzeugt, (2) einem markierten NT X' entspricht und (3) nur ein Kind hat, das nach Definition von P' damit auch einem markierten NT Y' entsprechen muss, gibt es eine Regel $X \rightarrow YZ \in P$ bzgl. der originalen Grammatik G . OBdA. ist G auf die nützlichen Symbole reduziert, womit es mindestens einen Ableitungsbaum zu Z gibt. Füge diesen als rechten Teilbaum bei dem inneren Knoten an. Ersetze dann X' durch X im Baum, um einen Ableitungsbaum bzgl. G zu erhalten, der ein Wort der Form uv erzeugt.

- (b) Sei L kontextfrei und L' regulär. Ohne Einschränkung gibt es $G = (V, \Sigma, P, S)$ CFG in CNF und $A = (Q, \Sigma, \delta, q_I, F)$ DFA mit $L = L(G)$ und $L' = L(A)$.

Idee: Wir erweitern G so, dass eine akzeptierende Berechnung von A geraten wird.

Definiere $G' = (V', \Sigma, P', S')$ wie folgt:

- $V' = Q \times V \times Q \cup \{S'\}$
- $P' = P'_{X \rightarrow YZ} \uplus P'_{X \rightarrow a} \uplus P'_{S'}$ mit

$$P'_{X \rightarrow YZ} := \{(q, X, q') \rightarrow (q, Y, q'')(q'', Z, q') \mid X \rightarrow YZ \in P\}$$

$$P'_{X \rightarrow a} := \{(q, X, q') \rightarrow a \mid X \rightarrow a \in P \wedge \delta(q, a) = q'\}$$

$$P'_{S'} := \{S' \rightarrow (q_I, S, q_F) \mid q_F \in F\} \cup \{S' \rightarrow \varepsilon \mid \varepsilon \in L(G) \cap L(A)\}$$

Behauptung: $L(G') = L(G) \cap L(A)$

Sei $w = a_1 a_2 \dots a_l \in L(G')$. Dann gibt es eine Ableitung der Form

$$S' \rightarrow (q_I, S, q_F) \rightarrow^* (q_0, X_1, q_1)(q_1, X_2, q_2) \dots (q_{l-1}, X_l, q_l)$$

mit $(q_{i-1}, X_i, q_i) \rightarrow a_i$.

Nach Def. von $P_{X \rightarrow YZ}$ muss $q_0 = q_I$ und $q_l = q_F$ gelten.

Nach Def. von $P_{X \rightarrow a}$ muss $q_i = \delta(q_{i-1}, a)$ gelten.

Damit existiert eine akzeptierende Berechnung von A zu w , also $w \in L(A)$.

Nach Definition erhält man durch Vergessen der q, q' , also durch Übergang von (q, X, q') zu X aus der obigen Ableitung auch eine Ableitung in G , womit auch $w \in L(G)$ folgt.

Insgesamt $w \in L(A) \cap L(G)$.

Sei $w \in L(A) \cap L(G)$. Dann existiert eine akzeptierende Berechnung $q_i = \delta(q_{i-1}, a_i)$ von A zu w und ein Ableitungsbaum zu w bzgl. G . Letzteren erweitert man induktiv zu einem Ableitungsbaum von G' : Den inneren Knoten, an dem das Blatt zu a_i hängt, schreibt man von X auf (q_{i-1}, X, q_i) um. Bottom-up/induktiv schreibt man dann X nach (q_i, X, q_k) um, falls seine beiden Kinder (da CNF) gerade $(q_i, Y, q_j), (q_j, Z, q_k)$ sind. Alternativ: aus X wird (q_i, X, q_k) , falls X das Teilwort $a_{i+1} \dots a_k$ erzeugt. Da G' die Zwischenzustände q'' frei mittels $P_{X \rightarrow YZ}$ raten kann, handelt es sich tatsächlich um einen Ableitungsbaum von w bzgl. P' , also $w \in L(G')$.

Alternativer Beweis über PDA:

Sei $P = (Q_P, \Sigma, \Gamma, q_{0_P}, Z_0, \delta_P, F_P)$ ein PDA der L akzeptiert und $M = (Q_M, \Sigma, \delta_M, q_{0_M}, F_M)$ ein DFA der L' akzeptiert.

Dann konstruieren wir einen PDA $P' = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ der $L \cap L'$ akzeptiert wie folgt:

- $Q = Q_P \times Q_M$
- $q_0 = (q_{0_P}, q_{0_M})$
- $F = F_P \times F_M$
- $\delta((p, q), a, Z) = \{((p', q'), \beta) \mid (p', \beta) \in \delta_P(p, a, Z) \wedge q' = \delta_M(q, a)\}$

Beweisskizze: Es ist zu zeigen dass ein $w \in \Sigma^*$ genau dann von P' (mit Endzustand) akzeptiert wird wenn es in $L \cap L'$ liegt. Wir zeigen die folgende verallgemeinerte Aussage mit Induktion über w

$$(q_0, Z_0) \xrightarrow{w}_{P'} ((p, q), \sigma) \iff (q_{0_P}, Z_0) \xrightarrow{w}_P (p, \sigma) \wedge q_{0_M} \xrightarrow{w}_M q$$

Induktionsbasis: $w = \varepsilon$

Folgt aus der Definition von q_0 und δ .

Induktionsschritt: Sei $w' = wa$ für ein beliebiges $a \in \Sigma$. Mit der IH gilt $(q_0, Z_0) \xrightarrow{w}_{P'} ((p, q), \sigma) \iff (q_{0_P}, Z_0) \xrightarrow{w}_P (p, \sigma) \wedge q_{0_M} \xrightarrow{w}_M q$. Sei o.B.d.A $\sigma = Z\sigma'$ für ein $Z \in \Gamma$. Mit der Definition von δ gilt $((p, q), Z\sigma') \xrightarrow{a}_{P'} ((p', q'), \alpha\sigma') \iff (p, Z\sigma') \xrightarrow{a}_P (p', \alpha\sigma') \wedge q_{0_M} \xrightarrow{a}_M q'$. Mit der Induktionshypothese folgt die Behauptung.