René Maseli	**Theoretische Informatik 1** **Große Übung 5**	TU Braunschweig
Prof. Dr. Roland Meyer		Wintersemester 2024/25

Übungsaufgabe 9.7:
Sei Σ ein Alphabet. Wir definieren $\text{rev} : \Sigma^* \to \Sigma^*$ mit $\text{rev}(w_1 w_2 \dots w_{n-1} w_n) = w_n w_{n-1} \dots w_2 w_1$ und $\text{rev}(L) := \{\, \text{rev}(w) \mid w \in L \,\}$. Zeigen Sie, dass sowohl die Klasse der regulären, als auch die der kontextfreien Sprachen unter rev abgeschlossen sind.

Zwei Fliegen, eine Klappe: Für eine formale Grammatik $G = \langle \Sigma, N, S, P \rangle$ definiere die formale Grammatik $\text{rev}(G) := \langle \Sigma, N, S, P' \rangle$ mit $P' := \{\, \text{rev}(\alpha) \to \text{rev}(\beta) \mid \alpha \to \beta \in P \,\}$.

Lemma
Es gilt $\mathcal{L}(\text{rev}(G)) = \text{rev}(\mathcal{L}(G))$.

Beweis
Sei $w \in \mathcal{L}(G)$ und $S = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_m = w$ eine Ableitung von G. Dann ist $S = \text{rev}(\alpha_0) \Rightarrow \text{rev}(\alpha_1) \Rightarrow \cdots \Rightarrow \text{rev}(\alpha_m) = \text{rev}(w)$ eine Ableitung von $\text{rev}(G)$. Es folgt also $\text{rev}(w) \in \mathcal{L}(\text{rev}(G))$ und, da dies für alle $w \in \mathcal{L}(G)$ gilt, folgt $\text{rev}(\mathcal{L}(G)) \subseteq \mathcal{L}(\text{rev}(G))$.
Da für alle $w \in \Sigma^*$ auch $\text{rev}(\text{rev}(w)) = w$ gilt, kann man analog auch $\mathcal{L}(\text{rev}(G)) \subseteq \text{rev}(\mathcal{L}(G))$ zeigen. $\square$

Korollar
Falls L kontextsensitiv ist, ist $\text{rev}(G)$ kontextsensitiv.
Falls G linkslinear ist, ist $\text{rev}(G)$ rechtslinear.
Falls G rechtslinear ist, ist $\text{rev}(G)$ linkslinear.

Beweis
Sei L kontextfrei. Dann gibt es eine kontextfreie Grammatik G mit $\mathcal{L}(G) = L$. Wie erwähnt, ist $\text{rev}(G)$ eine kontextfreie Grammatik, also ist $\mathcal{L}(\text{rev}(G)) = \text{rev}(L)$ kontextfrei.
Sei weiterhin L regulär. Dann gibt es eine linkslineare Grammatik G mit $\mathcal{L}(G) = L$. $\text{rev}(G)$ ist dann rechtslinear und damit ist $\mathcal{L}(\text{rev}(G)) = \text{rev}(L)$ wieder regulär. $\square$

Übungsaufgabe 9.8:
Es sei die folgende kontextfreie Grammatik G gegeben über $\Sigma = \{c, d\}$.

$$S \to A \mid ABB \qquad\qquad A \to Scc \mid \varepsilon \qquad\qquad B \to dB \mid c$$

Finden Sie eine äquivalente Grammatik G' in Chomsky-Normalform.
Finden Sie anschließend eine äquivalente Grammatik G'' in Greibach-Normalform.

Zur Erinnerung: Eine Grammatik ist in Chomsky-Normalform (CNF), falls alle Produktionen die Form $X \to YZ$ oder $X \to s$ haben. Die Greibach-Normalform (GNF) benötigt alle Produktionen in der Form $X \to sX_1 \dots X_n$ (mit $X, Y, Z, X_1, \dots X_n \in N$ und $s \in \Sigma$).

Chomsky-Normalform

Für die CNF eliminiert man zuerst die ε-Produktionen (sowohl A als auch S können zu ε übersetzt werden). Die resultierende Grammatik G_1 erfüllt $\mathcal{L}(G_1) = \mathcal{L}(G) \setminus \{\varepsilon\}$.

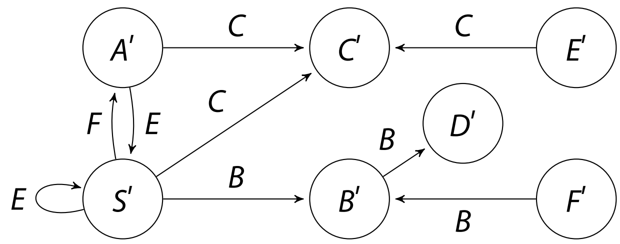
$$G_1 : \quad S \to A \mid ABB \mid BB \qquad A \to Scc \mid cc \qquad B \to dB \mid c$$

Nur $S \to BB$ und $B \to c$ haben schon CNF, der Rest muss angepasst werden, notfalls mit neuen Nichtterminalen. Dadurch erhalten wir eine Grammatik G' in CNF mit $\mathcal{L}(G') = \mathcal{L}(G_1)$.

$$G' : \quad S \to SE \mid CC \mid AF \mid BB \quad A \to SE \mid CC \quad B \to DB \mid c \quad C \to c \quad D \to d \quad E \to CC \quad F \to BB$$

Greibach-Normalform, Schritt 1

Für die GNF simuliert man die starke Linksableitung: für jedes $X \in N$ und $s \in \Sigma$ ist die Sprache $L_{X,s} = \{ \alpha \in N^* \mid X \Rightarrow^*_{SL} s.\alpha \}$ regulär. Um nah an dem zu bleiben, das wir kennen, kann man aus der Grammatik die Transitionen von NFAs ablesen. Lediglich die Start- und Endzustände variieren je nach $X \in N$ und $s \in \Sigma$. Beachtet, dass das Alphabet hier $N = \{S, A, B, C, D, E, F\}$ ist.



z.B. für $X = S$ und $s = c$ ist $q_0 = S'$ und $Q_F = \{B', C'\}$.

In dieser Richtung beschreibt der Automat allerdings das Reverse der gesuchten Sprache.

Die Nebenprodukt-Sprachen lassen sich nun mit Arden's Lemma errechnen und tabellarisch auflisten.

rev($L_{X,s}$)	X	S	A	B	C	D	E	F
s	$Q_F \setminus q_0$	S'	A'	B'	C'	D'	E'	F'
c	$\{B', C'\}$	$(E{+}FE)^*(B{+}C{+}FC)$	$C + E(E{+}FE)^*(B{+}C{+}FC)$	ε	ε	$\emptyset$	C	B
d	$\{D'\}$	$(E{+}FE)^*BB$	$E(E{+}FE)^*BB$	B	$\emptyset$	ε	$\emptyset$	BB

Diese Sprachen muss man nur noch reversen, was mit regulären Ausdrücken ganz leicht geht.

$L_{X,s}$	X	S	A	B	C	D	E	F
s	$Q_F \setminus q_0$	S'	A'	B'	C'	D'	E'	F'
c	$\{B', C'\}$	$(B{+}C{+}CF)(E{+}EF)^*$	$C + (B{+}C{+}CF)(E{+}EF)^*E$	ε	ε	$\emptyset$	C	B
d	$\{D'\}$	$BB(E{+}EF)^*$	$BB(E{+}EF)^*E$	B	$\emptyset$	ε	$\emptyset$	BB

Alternative

Die links-linearen Grammatiken liefern für jedes Terminal direkt ein Gleichungssystem, das man mit dem *reversen* Ardens Lemma lösen kann, um reguläre Ausdrücke für alle $L_{X,s}$ zu bekommen.

Seien $L, U, V \subseteq \Sigma^*$ Sprachen mit $\varepsilon \notin U$. Nach Ardens Lemma gilt $L = UL \cup V$ gdw. $L = U^*V$. Das Reverse von Ardens Lemma lautet $L = LU \cup V$ gdw. $L = VU^*$ und gilt genauso.

$L_{X,s}$	S	A	B	C	D	E	F
c	$L_{S,c}E + L_{C,c}C + L_{A,c}F + L_{B,c}B$ $= (C{+}CF{+}B)(E{+}EF)^*$	$L_{S,c}E + L_{C,c}C$ $= C + (C{+}CF{+}B)(E{+}EF)^*E$	$L_{D,c}B + \varepsilon$ $= \varepsilon$	ε	$\varnothing$	$L_{C,c}C$ $= C$	$L_{B,c}B$ $= B$
d	$L_{S,d}E + L_{C,d}C + L_{A,d}F + L_{B,d}B$ $= BB(E{+}EF)^*$	$L_{S,d}E + L_{C,d}C$ $= BB(E{+}EF)^*E$	$L_{D,d}B + \varnothing$ $= B$	$\varnothing$	ε	$L_{C,d}C$ $= \varnothing$	$L_{B,d}B$ $= BB$

Greibach-Normalform, Schritt 2

Für alle Ausdrücke mit Kleene-Stern sollte man **rechtslineare** Grammatiken erzeugen. Viele der Sprachen teilen sich reguläre Teilausdrücke, was einem die Arbeit erleichtert. Hier ist es nur ein Ausdruck $(E + EF)^*$. Die Grammatik kann wieder ε-Produktionen enthalten, aber die werden wir später eliminieren.

$$H \to \varepsilon \mid EH \mid EFH$$

Jetzt kann man den Teilausdruck mit H ersetzen. Jetzt produzieren wir die GNF-Produktionen für die originalen Nichtterminale. Die Idee: man produziert vorne ein Terminal und hinten die Sprache der zugehörigen Nebenprodukte. Dazu helfen Umformungen der regulären Ausdrücke, wie $C + (B + C + CF)HE = C + BHE + CHE + CFHE$.

$S \to cBH \mid cCH \mid cCFH \mid dBBH$

$A \to cC \mid cBHE \mid cCHE \mid cCFHE \mid dBBHE$

$B \to c \mid dB \qquad C \to c \qquad D \to d \qquad E \to cC \qquad F \to cB \mid dBB \qquad H \to \underline{\varepsilon} \mid \underline{E}H \mid \underline{E}FH$

Der letzte Schritt besteht nur noch aus Einsetzen und ε-Elimination:

$S \to cB \mid cBH \mid cC \mid cCH \mid cCF \mid cCFH \mid dBB \mid dBBH$

$A \to cC \mid cBE \mid cBHE \mid cCE \mid cCHE \mid cCFE \mid cCFHE \mid dBBE \mid dBBHE$

$B \to c \mid dB \qquad C \to c \qquad D \to d \qquad E \to cC \qquad F \to cB \mid dBB \qquad H \to cC \mid cCH \mid cCF \mid cCFH$

Alternative Für alle Sprachen generiert man eigene rechts-lineare Grammatiken.

$G_{X,s}$	S	A	B	C	D	E	F
c	$T_{S,c} \to CX \mid CFX \mid BX$ $X \to \varepsilon \mid EX \mid EFX$	$T_{A,c} \to CY \mid CFY \mid BY$ $Y \to E \mid EY \mid EFY$	$T_{B,c} \to \varepsilon$	$T_{C,c} \to \varepsilon$		$T_{E,c} \to C$	$T_{F,c} \to B$
d	$T_{S,d} \to BBX$ $X \to \varepsilon \mid EX \mid EFX$	$T_{A,d} \to BBX$ $X \to \varepsilon \mid EX \mid EFX$	$T_{B,d} \to B$		$T_{D,d} \to \varepsilon$		$T_{F,d} \to BB$

Nun vereinigen wir alle Grammatiken und erzwingen das Terminal-Symbol in jede Produktion. Ab hier behandeln wir N wieder als Nichtterminale. Dabei beschränken wir uns auf die nützlichen Nichtterminale, also starten mit S und vermeiden $T_{D,c}$, $T_{C,d}$ und $T_{E,d}$.

Iteration 0: $S \to cT_{S,c} \mid dT_{S,d}$
Iteration 1: $T_{S,c} \to cT_{C,c}X \mid cT_{C,c}FX \mid cT_{B,c}X \mid dT_{B,d}X$ $\quad T_{S,d} \to cT_{B,c}BX \mid dT_{B,d}BX$
Iteration 2: $T_{C,c} \to \varepsilon$ $\quad X \to \varepsilon \mid cT_{E,c}X \mid cT_{E,c}FX$ $\quad F \to cT_{F,c} \mid dT_{F,d}$ $\quad T_{B,c} \to \varepsilon$ $\quad T_{B,d} \to cT_{B,c} \mid dT_{B,d}$ $\quad B \to cT_{B,c} \mid dT_{B,d}$
Iteration 3: $T_{E,c} \to cT_{C,c}$ $\quad T_{F,c} \to cT_{B,c} \mid dT_{B,d}$ $\quad T_{F,d} \to cT_{B,c}B \mid dT_{B,d}B$

Zuletzt müssen nochmal die ε-Produktionen eliminiert werden. Betroffen sind $T_{C,c}$, X und $T_{B,c}$. Diese Grammatik G''' ist in GNF und erfüllt $\mathcal{L}(G''') = \mathcal{L}(G'') \setminus \{\varepsilon\} = \mathcal{L}(G) \setminus \{\varepsilon\}$.

$$S \to cT_{S,c} \mid dT_{S,d} \quad T_{S,c} \to c \mid cX \mid cF \mid cFX \mid dT_{B,d} \mid dT_{B,d}X \quad T_{S,d} \to cB \mid cBX \mid dT_{B,d}B \mid dT_{B,d}BX$$
$$X \to cT_{E,c} \mid cT_{E,c}X \mid cT_{E,c}F \mid cT_{E,c}FX \qquad F \to cT_{F,c} \mid dT_{F,d} \qquad T_{B,d} \to c \mid dT_{B,d}$$
$$B \to c \mid dT_{B,d} \qquad T_{E,c} \to c \qquad T_{F,c} \to c \mid dT_{B,d} \qquad T_{F,d} \to cB \mid dT_{B,d}B$$

Übungsaufgabe 9.9:
Betrachten Sie die Sprache $L = \{\, ww \mid w \in \{a, b\}^* \,\}$. Zeigen Sie, dass L nicht kontextfrei ist.

Zur Erinnerung: Das Pumping-Lemma nach Bar-Hillel, Perles & Shamir gibt jeder kontextfreien Sprache L eine Pumping-Konstante p_L, sodass jedes längere Wort $z \in L$ innerhalb der Sprache mindestens eine Zerlegung $z = uvwxy$ hat, mit einem kurzen Infix $|vwx| \leq p_L$, echten zu pumpenden Elementen $|vw| \geq 1$ und ohne Möglichkeit, durch Pumpen die Sprache zu verlassen: $\forall i \in \mathcal{N} : uv^iwx^iy \in L$.

Beweis
Nehme an, L sei kontextfrei. Dann gibt es eine Pumping-Konstante $p_L \in \mathbb{N}$.
Betrachte das Wort $z = a^{p_L}b^{p_L}a^{p_L}b^{p_L}$. Es gilt $z \in L$.
Sei $z = uvwxy$ eine Zerlegung mit (1) $|vwx| \leq p_L$ und (2) $vx \neq \varepsilon$. (Nach dem Pumping-Lemma gibt es mindestens eine pumpbare Zerlegung.)
Wichtige Beobachtung: Entweder befindet sich das Infix vwx komplett in einer der beiden Hälften von z, oder es enthält einen Suffix der vorderen Hälfte und einen Präfix der hintern Hälfte.
Im ersten Fall betrachte $z' = uv^iwx^iy$ mit $i = 0$ ($i \neq 1$ kann hier sogar beliebig gewählt werden). Wegen (2) wird die betroffene Hälfte verändert und stimmt nicht länger mit der anderen Hälfte überein. Es folgt $z' \notin L$. (Detailliertere Rechnung für Infix in vorderer Hälfte: Es gilt $|vx|_a > 0$ oder $|vx|_b > 0$ und daher $z' = a^{p_L+(i-1)\cdot|vx|_a}b^{p_L+(i-1)\cdot|vx|_b}a^{p_L}b^{p_L} \notin L$.
Anderenfalls beachte, dass der Infix wegen (1) weder ein vorderes a, noch ein hinteres b enthalten kann. Betrachte wieder $i = 0$ (oder ein beliebig anderes $i \neq 1$) und $z' = uv^0wx^0y$. Wegen (2) fehlt mindestens ein b der vorderen Hälfte oder ein a der hinteren Hälfte.
Da es keine weiteren Fälle mehr gibt, ist $z \in L$ nicht pumpbar.
Da dies aber dem Pumping-Lemma widerspricht, ist die Annahme falsch, also L nicht kontextfrei. □

Bemerkung: Tatsächlich sind diese z jeweils kürzeste Worte, die sich für diesen Beweis eignen.