René Maseli
Thomas Haas
Prof. Dr. Roland Meyer

Theoretische Informatik 1
Große Übung 5

TU Braunschweig
Wintersemester 2022/23

Definition
Sei Σ ein Alphabet und $w = w_1 w_2 \dots w_{n-1} w_n$ ein Wort. Wir definieren $\text{reverse}(w) = w_n w_{n-1} \dots w_2 w_1$.
Sei $L \subseteq \Sigma^*$ eine Sprache. Wir definieren $\text{reverse}(L) = \{ \text{reverse}(w) \mid w \in L \}$.

Theorem
Die Klasse der regulären Sprachen und die Klasse der kontextfreien Sprachen sind jeweils unter reverse abgeschlossen.

Beweis
Sei Σ ein Alphabet. Induktion über die „Tiefe" der regulären Ausdrücke:
Induktionsanfang: Sei L eine reguläre Sprache mit Tiefe 0. Die Ausdrücke mit Tiefe 0 sind $\varnothing$, $\{\varepsilon\}$, und für jedes $s \in \Sigma : \{s\}$. Es gilt $\text{reverse}(L) = L$ für alle diese Fälle.
Induktionsvoraussetzung: Es sei das Reverse jeder regulären Sprache mit Tiefe $\leq n$ wieder regulär.
Induktionsschluss: Sei L eine reguläre Sprache mit Tiefe $n + 1$. Dann gibt es reguläre Sprachen U und V mit Tiefe $\leq n$, sodass entweder $L = U \cup V$, $L = U.V$ oder $L = U^*$ gilt. Dann gilt jeweils $\text{reverse}(L) = \text{reverse}(U) \cup \text{reverse}(V)$, $\mathrm{L} = \text{reverse}(V).\text{reverse}(U)$ oder $\mathrm{L} = \mathrm{U}^*$. Da dies nun für alle Tiefen gilt, ist nun das Reverse jeder regulären Sprache wieder regulär. □

Bemerkung
Die Tiefe einer regulären Sprache meint hier die kleinste Tiefe eines regulären Ausdrucks zu dieser Sprache. Definiert ist $\text{depth}(x) = 0$ für atomaren Ausdrücke $\{\varnothing, \varepsilon\} \cup \Sigma$, sowie $\text{depth}(U + V) = \text{depth}(U.V) = 1 + \max(\text{depth}(U), \text{depth}(V))$ und $\text{depth}(U^*) = 1 + \text{depth}(U)$ für zusammengesetzte Ausdrücke.

Beweis
Sei $G = \langle N, \Sigma, P, S \rangle$ eine kontextfreie Grammatik. Die Grammatik $G_{\text{reverse}} = \langle N, \Sigma, P_{\text{reverse}}, S \rangle$ mit $P_{\text{reverse}} = \{ X, \text{reverse}(\alpha) \mid \langle X, \alpha \rangle \in P \}$ produziert genau $\mathcal{L}(G_{\text{reverse}}) = \text{reverse}(\mathcal{L}(G))$: Sei $w \in \mathcal{L}(G)$ und $S \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_m \Rightarrow w$ eine Ableitung von G. Dann ist $S \Rightarrow \text{reverse}(\alpha_1) \Rightarrow \cdots \Rightarrow \text{reverse}(\alpha_m) \Rightarrow \text{reverse}(w)$ eine Ableitung von G_{reverse}. Es folgt also $\text{reverse}(w) \in \mathcal{L}(G_{\text{reverse}})$ und da dies für alle $w \in \mathcal{L}(G)$ gilt, folgt $\text{reverse}(\mathcal{L}(G)) \subseteq \mathcal{L}(G_{\text{reverse}})$.
Da für alle $w \in \Sigma^*$ auch $\text{reverse}(\text{reverse}(w)) = w$ gilt, kann man analog auch $\mathcal{L}(G_{\text{reverse}}) \subseteq \text{reverse}(\mathcal{L}(G))$ zeigen. □

Chomsky-Normalform

Gefragt ist, ob die folgendekontextfreie Grammatik G das Wort *eedcedc* produzieren kann.

$$S \to ABC \mid CAB \qquad A \to dC \mid eBC \qquad B \to ed \mid BBC \qquad \to c$$

Dafür benötigen wir eine Chomsky-Normalform (CNF) von G. Mit den Regeln aus der Vorlesung erhalten wir so eine Grammatik G', die $\mathcal{L}(G') = \mathcal{L}(G) \setminus \{\varepsilon\}$ erfüllt (in diesem Fall sogar mit $\mathcal{L}(G) \setminus \{\varepsilon\} = \mathcal{L}(G)$).

$$S \to AF \mid CH \quad A \to DC \mid EF \quad B \to ED \mid BB \quad C \to c \quad D \to d \quad E \to e \quad F \to BC \quad H \to AB$$

E			$\underline{A}$		H	$\underline{S}$
	E	$\underline{B}$	$\underline{F}$			
		D	A		H	S
			C			
				E	$\underline{B}$	$\underline{F}$
					D	A
						C

Es gilt also $S \Rightarrow$ *eedcedc*. Die Ableitungen kann man mithilfe der Tabelle zurückverfolgen.

Greibach-Normalform

Gesucht ist eine Greibach-Normalform (GNF) zu der folgenden kontextfreien Grammatik G. Dabei sollen alle Produktionen die Form $X \to sX_1 \ldots X_n$ mit $X, X_1, \ldots X_n \in N$ und $s \in \Sigma$ haben.

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

Um die Transformation aus der Vorlesung anwenden zu können, benötigt man zuerst eine CNF. Dazu eliminiert man zuerst die ε-Produktionen (sowohl A als auch S können zu ε übersetzt werden). Die resultierende Grammatik G' erfüllt $\mathcal{L}(G') = \mathcal{L}(G) \setminus \{\varepsilon\}$.

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

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

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

Jetzt 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. Die links-linearen Grammatiken liefern ein Gleichungssystem, das man mit dem reverse von Ardens Lemma lösen kann, um reguläre Ausdrücke für alle $L_{X,s}$ zu bekommen. Beachtet, dass das Alphabet hier $N = \{S, A, B, C, D, E, F\}$ ist.

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

Bemerkung
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. $L = VU^*$.

Für diese Sprachen braucht man nun 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: $T_{S,c} \to cT_{C,c}X \mid cT_{C,c}FX \mid cT_{B,c}X \mid dT_{B,d}X \qquad T_{S,d} \to cT_{B,c}BX \mid dT_{B,d}BX$
Iteration 2: $T_{C,c} \to \varepsilon \qquad X \to \varepsilon \mid cT_{E,c}X \mid cT_{E,c}FX \qquad F \to cT_{F,c} \mid dT_{F,d} \qquad T_{B,c} \to \varepsilon$ $T_{B,d} \to cT_{B,c} \mid dT_{B,d} \qquad B \to cT_{B,c} \mid dT_{B,d}$
Iteration 3: $T_{E,c} \to cT_{C,c} \qquad T_{F,c} \to cT_{B,c} \mid dT_{B,d} \qquad T_{F,d} \to cT_{B,c}B \mid dT_{B,d}B$

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 $\mathcal{L}(G''') = \mathcal{L}(G'') \setminus \{\varepsilon\} = \mathcal{L}(G) \setminus \{\varepsilon\}$.

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