

Theoretische Informatik - Übungsblatt 13
Übungsgruppe 7 - Mi, 16-18 Uhr
Elena Kvassova

Jan Hendrik Dithmar (Matrikelnummer 2031259)
Dirk Heine (Matrikelnummer 2030941)

Aufgabe 1

- (a) Sei ein Wort $w = w_1w_2w_3w_4w_5 \dots$ gegeben. Nach der Regel $h(uv) = h(u)h(v)$ lässt sich die Abbildung des Wortes w in seine Zeichen zerlegen:

$$\begin{aligned} h(w) &= h(w_1w_2w_3w_4w_5 \dots) \\ &= h(w_1)h(w_2w_3w_4w_5 \dots) \\ &= h(w_1)h(w_2)h(w_3w_4w_5 \dots) \\ &= h(w_1)h(w_2)h(w_3)h(w_4w_5 \dots) \\ &= h(w_1)h(w_2)h(w_3)h(w_4)h(w_5) \dots \end{aligned}$$

Der Homomorphismus $h : \Sigma^* \rightarrow \Delta^*$ bildet somit jedes Zeichen aus Σ auf genau ein Zeichen aus Δ oder auf ε ab. Die Zeichen des Wortes können vollkommen isoliert betrachtet werden.

Eine kontextfreie Grammatik definiert sich durch Regeln, bei denen auf der linken Seite lediglich ein einzelnes Non-Terminal stehen darf, auf der rechten Seite jedoch eine beliebige Kombination von Terminalen, Non-Terminalen und auch ε . Um eine kontextfreie Grammatik für $h(L)$ zu bauen, brauchen wir also lediglich in den Regeln der bereits existierenden kontextfreien Grammatik für L die rechte Seite zu verändern. Dort werden alle Vorkommen von Terminalen a durch den entsprechenden Wert $h(a)$ ersetzt. Da ausschließlich die rechte Seite der Regeln verändert wird und durch die Änderung die Gültigkeit der Regeln nicht beeinflusst wird, ist die Grammatik dann immer noch kontextfrei. $\square$

- (b) Zeigen zunächst, dass die L Konkatenation einer kontextfreien Sprache L' und einer regulären Sprache L'' ebenfalls kontextfrei ist. Dazu konstruieren wir einen Kellerautomaten, der L akzeptiert. Dazu wird auf der Eingabe zunächst der Kellerautomat von L' ausgeführt. Während der Ausführung „raten“ wir nicht-deterministisch, an welcher Stelle das von L' akzeptierte Teilwort endet und das von L'' akzeptierte Teilwort beginnt. Dazu springen wir jeweils mit einer ε -Kante in den Startzustand des endlichen Automaten von L'' . Dieser Automat hat keinen Keller und kann somit problemlos in den Automaten L integriert werden. Es ist also gezeigt, dass die Konkatenation einer kontextfreien und einer regulären Sprache ebenfalls wieder kontextfrei ist. $\square$

Nun untersuchen wir die Aussage, dass L_1/L_2 kontextfrei ist. Beweis durch Widerspruch. Angenommen die Sprache L_1/L_2 sei *nicht* kontextfrei. Dann konkatenieren wir eine *nicht* kontextfreie Sprache L_1/L_2 mit einer regulären Sprache L_2 . Nach Voraussetzung müsste die Konkatenation kontextfrei sein, dies stellt jedoch einen Widerspruch zur bereits oben bewiesenen Aussage dar. $\square$

Aufgabe 2

Sei der Kellerautomat $K = (Q, \Sigma, \Gamma, q_0, Z_0, \delta)$, der mit leerem Keller akzeptiert, gegeben. Seien $q_i \in Q$ die Zustände des Automaten. Wir konstruieren den Kellerautomaten $K' = (Q', \Sigma, \Gamma', q'_0, Z'_0, \delta')$ wie folgt:

$$\begin{aligned} Q' &= \{q'_0\} \\ \Gamma' &= Q \cup \Gamma \end{aligned}$$

Zur Initialisierung des Stacks wird Z'_0 verwendet, wobei an unterster Stelle auf dem Stack alle Initialisierungswerte $a \in Z_0$ in der selben Reihenfolge stehen. An oberster

Stelle auf dem Stack steht der jeweils aktuelle Zustand des (simulierten) Automaten K , bei der Initialisierung also q_0 .

Die Überföhrungsfunktion δ' bestimmt anhand des obersten Symbols auf dem Stack immer den aktuellen Zustand von K und verwendet δ , um K entsprechend auszuföhren. Die Überföhrungsfunktion δ' definieren wir wie folgt:

$$q'_0, a, q_i :: S \longrightarrow q'_0, q''_i :: S' \quad \text{mit } (q''_i, S') = \delta(q_i, a, S)$$

Der Kellerautomat K' akzeptiert, wenn das Eingabewort komplett gelesen ist und der Stack bis auf einen Zustand q_i leer ist.

Aufgabe 3

- (a) Die Transformation von einer kurzen Greibach-Normalform in die Chomsky-Normalform überföhrt werden, in den folgende Ersetzungsregeln angewandt werden:

$$\begin{array}{lll} A \rightarrow a & \implies & A \rightarrow a \\ A \rightarrow aB & \implies & A \rightarrow A'B \\ & & A' \rightarrow a \\ A \rightarrow aBC & \implies & A \rightarrow A'D \\ & & A' \rightarrow a \\ & & D \rightarrow BC \end{array}$$

- (b) Zum Angeben von oberen Schranken für $|V'|$ und $|P'|$ reicht es den schlimmstmöglichen Fall zu betrachten. Der schlimmstmögliche Fall ist offensichtlich $A \rightarrow aBC$, da dann potentiell zwei neue Regeln und zwei neue Non-Terminale eingeföhrt werden müssen.

Im schlimmsten müssen wir $|P|$ Ersetzungen vornehmen, da wir $|P|$ Regeln haben, die potentiell geändert werden müssen. Für jede Regel aus G können im schlimmsten Fall zwei neue Regeln und zwei neue Non-Terminale notwendig sein. Die würde bedeutet, dass schlimmstenfalls $2 \cdot |P|$ neue Regeln und $2 \cdot |P|$ neue Non-Terminale eingeföhrt werden müssen.

Die neuen Regeln der Form $A' \rightarrow a$ sind in ihrer Anzahl jedoch durch die Anzahl der Terminale beschränkt, da jedes für diese Ergänzungen eingeföhrt Non-Terminal auf genau ein Terminal abbildet. Dadurch lässt sich die Anzahl der Ergänzungen dieser Art zusätzlich auf die Anzahl der Terminale beschränken.

Insgesamt gelten also folgende obere Schranken:

$$\begin{array}{ll} |V'| & \leq |P| + \min(|P|, |T|) \\ |P'| & \leq |P| + \min(|P|, |T|) \end{array}$$