Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Christian Ivicevic

Sommersemester 2015
Studentisches Tutorium
Übungsblatt 6
1. Juni 2015

Theoretische Informatik

– Strukturelle Induktion, Kellerautomaten 1 (NPDA) –

Grammatiken und Korrektheit

In vielen Situationen ist es nötig, eine Grammatik herzuleiten, die Wörter erzeugt, die eine bestimmte Struktur haben. Um zu beweisen, dass eine Grammatik korrekt ist (also $L = L(G)$ gilt) nutzen wir das Konzept von *struktureller Induktion* in zwei Richtungen ($L(G) \subset L$ und $L \subset L(G)$).

Beispiel 1. Wir betrachten das Alphabet $\Sigma = \{a, b\}$ und wollen eine Grammatik G herleiten, für die gilt, dass alle Wörter in $L(G)$ doppelt so viele a's wie b's besitzen, i.Z.

$$L(G) = \{w \in \Sigma^* \mid \#_a(w) = 2 \cdot \#_b(w)\}.$$

Sei nun durch $S \to aSaSb \mid aSbSa \mid bSaSa \mid SS \mid \varepsilon$ eine Grammatik impliziert von der wir vermuten, dass sie korrekt ist.

Hinrichtung ($L(G) \subset L$) Unsere Induktionshypothese lautet

$$w \in L(G) \implies w \in L \implies \#_a(w) = 2 \cdot \#_b(w). \tag{*}$$

Hierzu betrachen wir jede Produktion in der Grammatik G und werden beweisen, dass Wörter, die erzeugt werden, die gegebene Eigenschaft erfüllen.

- Fall (1): Sei $w = au_1au_2b$ mit $u_1, u_2 \in L(G)$, dann gilt

$$\begin{aligned}\#_a(w) &= 1 + \#_a(u_1) + 1 + \#_a(u_2)\\ &\overset{(*)}{=} 1 + 2 \cdot \#_b(u_1) + 1 + 2 \cdot \#_b(u_2)\\ &= 2 \cdot (1 + \#_b(u_1) + \#_b(u_2)) = 2 \cdot \#_b(w).\end{aligned}$$

- Fall (2) + (3): Analog zu Fall (1).
- Fall (4): Sei $w = u_1u_2$ mit $u_1, u_2 \in L(G)$, dann gilt

$$\begin{aligned}\#_a(w) &= \#_a(u_1) + \#_a(u_2)\\ &\overset{(*)}{=} 2 \cdot \#_b(u_1) + 2 \cdot \#_b(u_2)\\ &= 2 \cdot (\#_b(u_1) + \#_b(u_2)) = 2 \cdot \#_b(w).\end{aligned}$$

- Fall (5): Sei $w = \varepsilon$, dann gilt $\#_a(w) = 0 = 2 \cdot 0 = 2 \cdot \#_b(w)$.

Rückrichtung ($L \subset L(G)$) Der umgekehrte Beweis ist meist sehr komplex, weshalb wir nur einen Teil beispielhaft beweisen, aber die Rückrichtung damit **nicht** vollständig wäre! In diesem Teil wollen wir beweisen, dass Wörter $(ab)^n a^n$ für $n \geq 0$, die offensichtlich die Eigenschaft erfüllen, auch von unserer Grammatik ableitbar sind. Hierzu nutzen wir reguläre Induktion und erhalten für $n = 0$

$$(ab)^0 a^0 = \varepsilon \in L(G), \text{ durch } S \to \varepsilon.$$

Somit ist unsere Indukstionshypothese

$$S \to_G^* (ab)^n a^n \text{ gilt für ein beliebiges, aber festes, } n. \tag{**}$$

Durch strategisches Ausprobieren erhalten wir bei $n \to n+1$

$$S \to_G aSbSa \to_G abSa \xrightarrow{(**)}{}^*_G ab(ab)^n a^n a = (ab)^{n+1} a^{n+1}.$$

Somit ist jedes Wort der Form $(ab)^n a^n$ von unserer Grammatik G ableitbar und ein Teil der Rückrichtung wäre an dieser Stelle bewiesen.

Beispiel 2. Sei $\Sigma = \{0, 1\}$. Zu einem Wort $w \in \Sigma^*$ definieren wir dessen Spiegelung w^R und Negation $\overline{w}$ wie folgt

$$w^R = \begin{cases} \varepsilon, & \text{falls } w = \varepsilon, \\ u^R a, & \text{falls } w = au, a \in \Sigma, u \in \Sigma^*. \end{cases}$$

$$\overline{w} = \begin{cases} \varepsilon, & \text{falls } w = \varepsilon, \\ \hat{a}\overline{u}, & \text{falls } w = au, a \in \Sigma, u \in \Sigma^*. \end{cases}$$

Dabei setzen wir $\hat{0} = 1$ und $\hat{1} = 0$. Außerdem gilt (ohne expliziten Beweis) $(ua)^R = au^R$ und $\overline{ua} = \overline{u}\hat{a}$ für alle $a \in \Sigma, u \in \Sigma^*$. In diesem Beispiel betrachten wir die Sprache $L = \{w \in \Sigma^* \mid w^R = \overline{w}\}$ und die Grammatik $G : S \to \varepsilon \mid 0S1 \mid 1S0$ und werden beweisen, dass L genau die von G beschriebene Sprache ist, also $L = L(G)$ gilt.

1. Zunächst zeigen wir die Hinrichtung, also das $L(G) \subset L$ gilt und damit natürlich auch $w \in L(G) \implies w \in L$.

   - Bei der ersten Produktion $S \to \varepsilon$ erhalten wir $w = \varepsilon \in L(G)$ und es gilt

     $$\varepsilon^R = \varepsilon = \overline{\varepsilon}$$

     mithin auch $w = \varepsilon \in L$.
   - Für die zweite Produktion $S \to 0S1$ wählen wir ein Teilwort $u \in L(G)$ mit $w = 0u1 \in L(G)$ und nach Induktionshypothese gilt auch $u \in L$. Daraus folgt

     $$(0u1)^R = (u1)^R 0 = 1u^R 0 \overset{u \in L}{=} 1\overline{u}0 = \hat{0}\overline{u}\hat{1} = \overline{0u}\hat{1} = \overline{0u1},$$

     mithin auch $w = 0u1 \in L$.
   - Die letzte Produktion ist analog zu beweisen.

Da wir jede Produktion abgedeckt haben ist die Behauptung $L(G) \subset L$ bewiesen.

2. Die Rückrichtung, also $L \subset L(G)$ und somit auch $w \in L \implies w \in L(G)$ beweisen wir mit vollständiger Induktion über die Wortlänge $|w|$ eines Wortes w.

   - Sei $|w| = 0$ und somit $w = \varepsilon \in L$, da die Eigenschaft offensichtlich erfüllt ist. Mit der Produktion $S \to \varepsilon$ ist somit auch $w \in L(G)$ bewiesen.
   - Für $|w| = 1$ erhalten wir $w = w^R$, aber $w \neq \overline{w}$ und somit $w \notin L$. Damit gilt auch $w \in L \implies w \in L(G)$.
   - Betrachten wir nun $|w| \geq 2$ so hat dieses Wort die Form $w = aub$ für das gilt $a, b \in \Sigma, u \in \Sigma^*$. Durch geschickte Anwendung der Rechenregeln erhalten wir die Gleichung

     $$bu^Ra = (ub^R)a = (aub)^R \overset{w \in L}{=} \overline{aub} = \hat{a}\overline{ub} = \hat{a}\overline{u}\hat{b}$$

     und können nun einige Eigenschaften ablesen. Es muss nun $b = \hat{a}$ und $a = \hat{b}$ gelten, sowie $u^R = \overline{u}$. Da zudem $|u| < |w|$ gilt, folgt nach der Induktionshypothese, dass $u \in L(G)$ ebenso gilt. Daraus folgt, dass das Wort w die Form $0u1$ oder $1u0$ mit $u \in L(G)$ haben muss. Dies ist gerade mit $S \to 0S1$ bzw. $S \to 1S0$ gegeben.

   In diesem Beweis haben wir nun unabhängig von der Wortlänge gezeigt, dass jedes Wort in L auch in $L(G)$ vorkommt. (Im Gegensatz zum vorherigen Beispiel, in dem wir nur eine Teilmenge betrachtet haben.)

Kellerautomaten (PDA)

Bisher haben wir gesehen, dass eine reguläre Sprache von einem DFA akzeptiert werden kann, dieser aber nicht mächtig genug ist um eine kontextfreie Sprache zu akzeptieren, da wir Zusatzinformationen speichern müssten. Hierzu führen wir einen neuen Automaten ein, der zusätzlich mit einem Keller (auf neuhochdeutsch auch als Stack bekannt) arbeitet und auf diesem zusätzliche Informationen abspeichern kann.

Definition (Kellerautomat)**.** *Ein Kellerautomat K (engl. (D/N)PDA) ist ein 7-Tupel $K = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, der zusätzlich auf einem Keller operiert, der anfangs mit dem Kellerstartsymbol Z_0 befüllt ist, welches Teil des Kelleralphabets Γ ist. Die Zustandsübergangsrelation ist über $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$ definiert. Ein $\delta(q, s, z) \ni (q', z')$ beschreibt einen Zustandsübergang von q nach q', falls $s \in \Sigma$ gelesen wurde und zugleich $z \in \Gamma$ das oberste Element auf dem Keller ist. Das Symbol z wird entfernt und das Wort $z' = z'_1 \dots z'_n \in \Gamma^*$ neu auf den Keller gelegt, wobei z'_1 das oberste Element auf dem Keller sein wird. Falls $|\delta(q, s, z)| \leq 1$ für alle $q \in Q, s \in \Sigma \cup \{\varepsilon\}, z \in \Gamma$ gilt, so bezeichnen wir K als* deterministisch*, andernfalls als* nichtdeterministisch*. Weiterhin nennen wir ein Tupel $(q, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$* Konfiguration *und beschreiben damit, dass sich ein Automat aktuell im Zustand q befindet, noch das Wort w lesen muss und sich α noch auf dem Keller befindet. Ein Schritt in der Konfiguration $(q, w_1w_2 \dots w_m, \alpha_1\alpha_2 \dots \alpha_n) \to_K (q', w_2 \dots w_m, \beta_1 \dots \beta_k\alpha_2 \dots \alpha_n)$ impliziert $\delta(q, w_1, \alpha_1) \ni (q', \beta_1 \dots \beta_k)$. K akzeptiert nun* *ein Wort $w \in \Sigma^*$ <u>durch akzeptierenden Zustand</u>, i.Z. $w \in L(K)$, falls $(q_0, w, Z_0) \to_K^* (q, \varepsilon, \alpha)$ mit $q \in F, \alpha \in \Gamma^*$ gilt. Alternativ wird ein Wort $w \in \Sigma^*$ <u>durch leeren Keller</u> akzeptiert, i.Z. $w \in L_\varepsilon(K)$, falls $(q_0, w, Z_0) \to_K^* (q, \varepsilon, \varepsilon)$ gilt.*

Beispiel 3. Sei $K = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ mit $Q = \{q\}, \Sigma = \{a, b, \#\}, \Gamma = \{Z_0, X, Y, Z\}$ und $F = \emptyset$ gegeben mit

$$\begin{aligned}
\delta(q, \varepsilon, Z_0) &= \{(q, XZ)\}, \\
\delta(q, \#, X) &= \{(q, \varepsilon)\}, \\
\delta(q, a, Z) &= \{(q, \varepsilon)\}, \\
\delta(q, a, X) &= \{(q, XY)\}, \\
\delta(q, b, Y) &= \{(q, \varepsilon)\}.
\end{aligned}$$

Wir können einen Kellerautomaten graphisch darstellen, indem wir für Zustandsübergänge die einzelnen Kanten von q nach q' mit $s, A/B$ beschriften, sofern $\delta(q, s, A) \ni (q', B)$ gilt. Angewandt auf unser Beispiel erhalten wir folgenden Kellerautomaten:

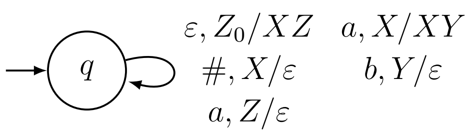

Abbildung 1: Graphische Darstellung eines Kellerautomaten

Sofern der gegebene Automat das Wort $a\#ba$ liest durchläuft er die Zustandsfolge

$$\begin{aligned}
(q, a\#ba, Z_0) &\to_K (q, a\#ba, XZ) \to_K (q, \#ba, XYZ) \\
&\to_K (q, ba, YZ) \to_K (q, a, Z) \to_K (q, \varepsilon, \varepsilon)
\end{aligned}$$

und akzeptiert somit mit leerem Keller, d.h. $a\#ba \in L_\varepsilon(K)$.

Beispiel 4. Wir betrachten die Sprache $\{d^{2k}e^k \mid k \in \mathbb{N}_0\}$ und können mit dem Kelleralphabet $\Gamma = \{Z_0, \lambda\}$ einen PDA konstruieren, der mit leerem Keller akzeptiert:

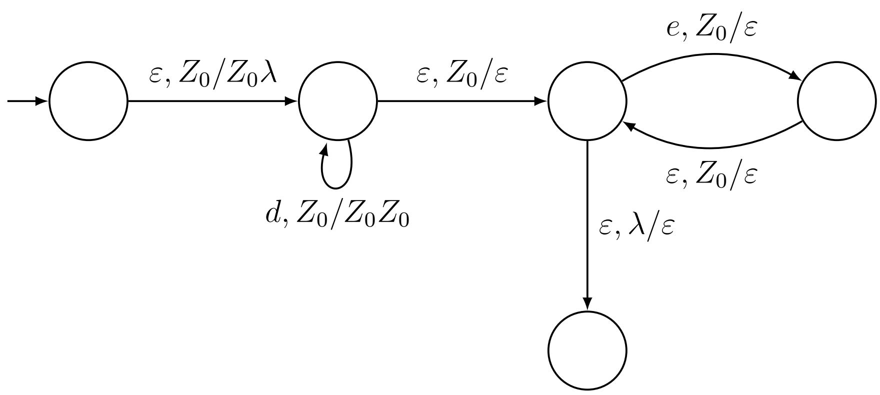

Abbildung 2: PDA für $\{d^{2k}e^k \mid k \in \mathbb{N}_0\}$