Myhill-Nerode-Äquivalent und Pumping-Argumente

Niklas Rieken

May 31, 2017

In diesem Dokument geht es um die Myhill-Nerode-Äquivalenz und ihre Anwendungen für reguläre Sprachen. Sie liefern vorallem ein notwendiges und hinreichendes Bedingung dafür ob eine Sprache regulär ist oder nicht. Der in der Vorlesung (zu Unrecht, meiner Meinung nach) am meisten beworbene Weg um zu zeigen, dass eine Sprache nicht regulär ist, ist die Anwendung des Pumping-Lemma, was für viele FoSAP-Hörer das Schreckgespenst der Vorlesung ist, obwohl das dahinter stehende Argument recht einfach ist. Wir zeigen hier einen ähnliches Argument: Es ist nicht genau das Pumping-Lemma. Stattdessen beweisen wir etwas direkter, dass für eine nicht-reguläre Sprache kein endlicher Automat existieren kann.

Myhill-Nerode-Äquivalenz

Bisher sind reguläre Sprachen durch Formalismen wie reguläre Ausdrücke und endliche Automaten charakterisiert worden. Nun soll eine Eigenschaft gefunden werden, die reguläre Sprachen auf einer stark mathematisch-strukturellen Ebene charakterisiert, d.h. mit Aussagen, die sich auf die Struktur

$$(\Sigma^*, \cdot, \varepsilon)$$

beziehen. Diese Struktur bezeichnet man auch als *Wortmonoid*.

Grundlegende Überlegungen

Bevor wir zur eigentlichen Definition der Myhill-Nerode-Äquivalenz kommen, wiederholen wir ein paar Definitionen:

Definition 1. Eine **Äquivalenzrelation** über einer Menge A ist eine zweistelltige Relation $\sim \subseteq A \times A$, die

(i) reflexiv (für alle x gilt $x \sim x$),

(ii) symmetrisch (wenn $x \sim y$ dann auch $y \sim x$) und

(iii) transitiv (wenn $x \sim y$ und $y \sim z$ dann auch $x \sim z$)

ist.

Da die Myhill-Nerode-Relation eigentlich sogar etwas mehr ist, hier noch eine weitere Definition:

Definition 2. Sei A eine Menge und $\circ$ eine zweistellige Funktion auf A. Eine Relation $\sim \subseteq A \times A$ heißt **rechtsseitige Kongruenz** (bezüglich $\circ$), wenn

(i) $\sim$ eine Äquivalenzrelation ist und

(ii) $\circ$ diese Relation respektiert (d.h. für alle $x \sim y$ und alle $z \in A$ gilt $x \circ z \sim y \circ z$).

Die Myhill-Nerode-Äquivalenz ist definiert durch:

Definition 3. Sei Σ ein Alphabet und $L \subseteq \Sigma^*$. Seien $u, v \in \Sigma^*$. u und v heißen **Myhill-Nerode-äquivalent** ($u \equiv_L v$) genau dann, wenn für alle $w \in \Sigma^*$ gilt, dass $uw \in L$ g.d.w. $vw \in L$.

Diese Relation ist streng genommen nicht nur eine Äquivalenz sondern auch eine rechtsseitige (!) Kongruenz, deswegen sind auch die Begriffe *Myhill-Nerode-Rechtskongruenz* und – nicht ganz korrekterweise – *Myhill-Nerode-Kongruenz* gebräuchlich.
Eine einfacher Folgerung aus der Definition ist

Lemma 4. *Sei $L \subseteq \Sigma^*$ eine Sprache. Für alle $u, v, w \in \Sigma^*$ mit $u \equiv_L v$ gilt*

$$uw \equiv_L vw.$$

Zuletzt der Begriff der Äquivalenzklasse:

Definition 5. Sei A eine Menge und $\equiv \subseteq A \times A$ eine Äquivalenzrelation. Eine Menge $A' \subseteq A$ heißt **Äquivalenzklasse**, wenn

(i) Wenn $a \in A'$ und $b \equiv a$, dann ist auch $b \in A'$,

(ii) Für alle $a, b \in A'$ gilt $a \equiv b$.

Man benennt Äquivalenzklassen auch durch einzelne Repräsentanten, z.B. mit $[u]_L$ oder $u/_L$ für die Äquivalenzklasse in der alle Elemente enthalten sind, die zu u äquivalent sind. Die Vereinigung aller Äquivalenzklassen zu einer Äquivalenz über einer Menge A bildet stets eine Partition von A.

Finden der Äquivalenzklassen

Lemma 4 kann dafür verwendet werden um Äquivalenzklassen (der Myhill-Nerode-Äquivalenz $\equiv_L$) zu finden. Wir gehen dabei wie folgt vor:
Wir starten mit dem leeren Wort ε und der zugehörigen Äquivalenzklasse $[\varepsilon]_L$. Danach wiederholen wir folgende Schritte bis keine neue Äquivalenzklassen mehr gefunden werden:
Für jedes Symbol $a \in \Sigma$ und jede bisher gefundene Äquivalenzklasse $[u]_L$ betrachte die Äquivalenzklasse $[ua]_L$. Ist $[ua]_L = [v]_L$ für eine bereits gefundene Äquivalenzklasse, passiert nichts. Ist dies nicht der Fall, fügen wir $[ua]_L$ als neue Äquivalenzklasse hinzu.
Beachte, dass dieses Verfahren nur für reguläre Sprachen terminiert. Nichtreguläre Sprachen haben unendlich viele Äquivalenzklassen (s. Satz 6).
Dazu machen wir ein Beispiel. Sei

$$L = \{w \in \{a, b\}^* \mid w \text{ beginnt mit } b \text{ und nach jedem } a \text{ folgt sofort ein } b\}.$$

- Jedes Wort aus $\{a, b\}^*$ hat eine Äquivalenzklasse, also auch ε. Wir bezeichnen diese mit $[\varepsilon]_L$.
- Betrachte nun $[\varepsilon a]_L = [a]_L$. Es ist $[a]_L \neq [\varepsilon]_L$ (also $a \not\equiv_L \varepsilon$), da b die Wörter trennt, d.h. $a \cdot b \notin L$, aber $\varepsilon \cdot b \in L$. d.h. $[a]_L$ ist eine neue Äquivalenzklasse.
- Betrachte nun $[\varepsilon b]_L = [b]_L$. Es ist $[b]_L \neq [\varepsilon]_L$, da ε die Wörter trennt, d.h. $b \cdot \varepsilon \in L$, aber $\varepsilon \cdot \varepsilon \notin L$. Außerdem $[b]_L \neq [a]_L$, weil ε die Wörter trennt, d.h. $b \cdot \varepsilon \in L$, aber $a \cdot \varepsilon \notin L$.
- Betrachte nun $[aa]_L$. Es ist $[aa]_L = [a]_L$, da es kein trennendes Wort gibt (egal, was man anhängt, man kann nicht mehr in der Sprache landen in beiden Fällen, da das Wort nicht mit b beginnt).
- Betrachte nun $[ab]_L$. Es ist $[ab]_L = [a]_L$, da es kein trennendes Wort gibt (egal, was man anhängt, man kann nicht mehr in der Sprache landen in beiden Fällen, da das Wort nicht mit b beginnt).
- Betrachte nun $[ba]_L$. Es ist $[ba]_L = [\varepsilon]_L$, da es kein trennendes Wort gibt.
- Betrachte nun $[bb]_L$. Es ist $[bb]_L = [b]_L$, da es kein trennendes Wort gibt.
- Damit sind alle Äquivalenzklassen gefunden: $[\varepsilon]_L, [a]_L, [b]_L$. Dieser Punkt ist für viele Studenten oft erstmal schwierig zu sehen. Es folgt aber direkt aus der rechtsseitigen Kongruenz bzw. Lemma 4. Egal welches Wort wir betrachten, wir können es einer dieser Äquivalenklassen zuordnen. Betrachte zum Beispiel $[bab]_L$. Wegen $ba \equiv_L \varepsilon$ ist $ba \cdot b \equiv_L \varepsilon \cdot b = b$ und somit $[bab]_L = [b]_L$.

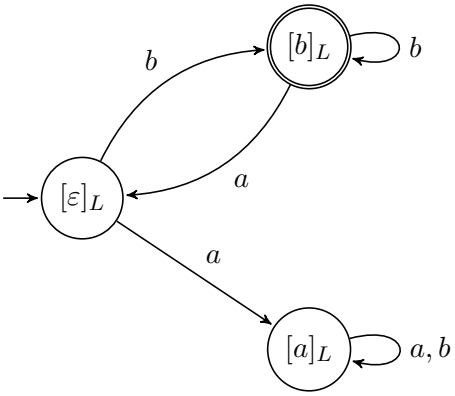

Figure 1: Minimaler DFA für die Sprache L.

Wofür ist das nützlich?

Die Myhill-Nerode-Äquivalenz ist inhaltlich eines der schönsten Sachen in FoSAP, weil sie 1. eine notwendige und hinreichende Bedingung liefert, dass eine Sprache regulär ist und weil sie 2. auch einen direkten Weg zu einem minimalen deterministischen endlichen Automaten liefert.

Theorem 6 (Satz von Nerode)**.** *Eine Sprache ist genau dann regulär, wenn die Anzahl der Myhill-Nerode-Äquivalenzklassen endlich ist.*

Man schreibt auch $index(\equiv_L) < \infty$. Die Sprache L im Beispiel hat $index(\equiv_L) = 3 < \infty$ und ist somit regulär. Betrachte nun ein Standardbeispiel für eine nicht-reguläre Sprache

$$K = \{a^n b^n : n \in \mathbb{N}\}.$$

K ist nicht regulär. Betrachte die Wörter $u_i = a^i$ und $u_j = a^j$ mit $i \neq j$. Es gilt offensichtlich $u_i \not\equiv_K u_j$, denn das Wort b^i trennt die Wörter, weil $u_i b^i = a^i b^i \in K$, aber $u_j b^i = a^j b^i \notin K$. Da es unendlich viele $i, j \in \mathbb{N}$ mit $i \neq j$ gibt, gibt es auch unendlich viele Äquivalenzklassen. Also ist K nach dem Satz von Nerode nicht regulär.

Theorem 7. *Der minimale DFA zu einer reguläre Sprache L ist isomorph zum Myhill-Nerode-DFA:*

$$\mathcal{A}_L = (\Sigma^*/_L, \Sigma, \delta, [\varepsilon]_L, \{[u]_L : u \in L\}),$$

mit $\delta([u]_L, a) = [ua]_L$.

Für die Sprache L erhalten wir also den DFA in Abbildung 1.

Pumping-Argumente

Dies ist ein Thema mit der viele Studenten in der Regel Probleme haben, da das Pumping-Lemma etwas sperrig ist und nicht sehr natürlich wirkt. Zur Erinnerung aber trotzdem nochmal:
Sei L eine reguläre Sprache. Dann existiert ein $n \in \mathbb{N}_+$, sodass für alle $w \in L$ mit $|w| \geq n$ eine Zerlegung $w = xyz$ existiert für die gilt:

(i) $|xy| \leq n$,

(ii) $y \neq \varepsilon$,

(iii) $xy^iz \in L$ für alle $i \in \mathbb{N}$.

Wir betrachten jetzt die Sprache

$$L = \{uav \,|\, u, v \in \{a, b\}^* \text{ mit } |u| = |v|\}$$

Der herkömmliche Weg, der auch in der Vorlesung vorgestellt wurde ist nun genau dieses Lemma benutzen und die Annahme, dass L regulär ist zu einem Widerspruch zu führen. Das ist aber genau das womit viele Schwierigkeiten haben, deswegen versuche ich das hier etwas anschaulicher:
Angenommen L ist regulär, dann gibt es einen endlichen Automaten $\mathcal{A} = (Q, \Sigma, \Delta, q_0, F)$, der L erkennt. Angenommen $\mathcal{A}$ habe n Zustände. Wir betrachten das Wort $w = b^nab^n$. Das Wort ist in der Sprache und hat die Länge $2n + 1 > n$. Nach Lesen des Präfix b^n muss also spätestens eine Zustandswiederholung im akzeptierenden Lauf von $\mathcal{A}$ auf w aufgetreten sein (*pigeonhole principle, Schubfachprinzip*), d.h. vor dem Lesen vom a. Wir nehmen an, der Zustand, der sich wiederholt trat nach Lesen des i-ten und j-ten (oBdA $j > i$) b auf (s. Abbildung 2). Wir betrachten also den Lauf

$$r = (q_0, b, q_1, \ldots, b, q_i, b, \ldots, q_j, b, q_{j+1}, \ldots, b, q_n, a, q_{n+1}, b, \ldots, b, q_{2n+1}),$$

mit $q_{2n+1} \in F$ wobei $q_i = q_j$. Sei $j - i = k > 0$. Der Lauf

$$r = (q_0, b, \ldots, q_i, b, q_{j+1}, \ldots, b, q_{2n+1})$$

ist also ebenfalls möglich und auch akzeptierend mit dem Wort $w' = b^{n-k}ab^n$, aber $w' \notin L$, also erkennt $\mathcal{A}$ nicht die Sprache mit n Zuständen. Da n beliebig war gibt es also keinen endlichen Automaten, der L akzeptiert. Somit folgt, dass L nicht regulär ist.
Man sieht vielleicht, dass in dieser Lösung die Wörter und Zahlen aus dem Pumping Lemma wieder auftauchen, was natürlich kein Zufall ist (n, Das Präfix b^j entspricht xy, das Infix b^k mit $k > 0$ entspricht y und das Weglassen von b^k in w' entspricht der Betrachtung von xz in *iii)*).

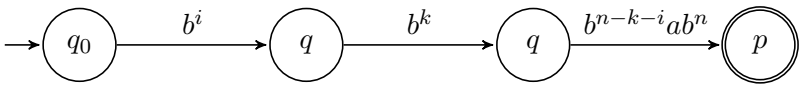

Figure 2: Zustandswiederholung in q. Es gibt also einen Lauf von q_0 nach q und einen Lauf von q nach p. Den Zwischenlauf von q nach q kann man also weglassen (oder auch beliebig oft wiederholen).