Contributions to the structure theory of ω-languages

Albert Zeyer

10. Februar 2011

Inhaltsverzeichnis

1 Introduction

...

2 Automat

Ein **Automat** $\mathcal{A}$ auf dem Alphabet Σ ist gegeben durch eine Menge Q von Zuständen und einer Teilmenge $E \subset Q \times A \times Q$ von Transitionen. Außerdem ist in der Regel eine Teilmenge $I \subset Q$ von Startzuständen und eine Teilmenge $F \subset Q$ von Endzuständen gegeben.

Wir schreiben dafür: $\mathcal{A} = (Q, \Sigma, E, I, F)$.

Der Automat ist endlich genau dann, wenn Q und Σ endlich sind.

Der Automat ist deterministisch, wenn E eine Menge von Funktionen $Q \times A \to \mathcal{Q}$ und wenn $|I| = 1$ sind.

2.1 Pfad

Zwei Transitionen $(p, a, q), (p', a', q') \in E$ sind aufeinanderfolgend, wenn $q = p'$.

Ein Pfad in dem Automat $\mathcal{A}$ ist eine Folge von aufeinanderfolgenden Transitionen, geschrieben als: $q_0 \to^{a_0} q_1 \to^{a_1} q_2 \ldots$

2.2 Akzeptanz von endlichen Wörtern

Ein Automat $\mathcal{A} = (Q, \Sigma, E, I, F)$ **akzeptiert** ein endliches Wort $w = (a_0, a_1, ..., a_n) \in \Sigma^*$ genau dann, wenn es einen Pfad $q_0 \to^{a_0} q_1 \to^{a_1} q_2 \cdots \to^{a_n} q_{n+1}$ gibt mit $q_0 \in I$ und $q_{n+1} \in F$.

Die Sprache $L^*(\mathcal{A})$ ist definiert als die Menge aller Wörter, die von $\mathcal{A}$ akzeptiert werden.

3 $*$-Sprachklassen

Die $*$-Sprachklasse ist die Menge aller Sprachen von Wörtern $w \in \Sigma^*$, also die Menge von Sprachen von endlichen Wörtern.

3.1 reguläre Sprachen

Eine Sprache ist genau dann regulär, wenn sie von einem endlichen Automat erkannt wird.

3.2 piece-wise testable

3.3 k-locally testable

3.4 dot-depth-n

3.5 starfree

3.6 locally modulo testable

3.7 R-trivial

3.8 endlich / co-endlich

3.9 endwise testable

4 ω-Sprachklassen

4.1 Büchi Automat

Ein Automat $\mathcal{A} = (Q, \Sigma, E, I, F)$ **Büchi-akzeptiert** ein Wort $w = (a_0, a_1, a_2, ...) \in \Sigma^\omega$ genau dann, wenn es einen unendlichen Pfad $q_0 \rightarrow^{a_0} q_1 \rightarrow^{a_1} q_2 \rightarrow^{a_2} q_3 ...$ gibt mit $q_0 \in I$ und $\{q_i | q_i \in F\}$ unendlich, also der unendlich oft einen Zustand F erreicht.

Die Sprache $L^\omega(\mathcal{A})$ ist definiert als die Menge aller unendlichen Wörter, die von $\mathcal{A}$ Büchi-akzeptiert werden.

Man bezeichnet einen Automaten $\mathcal{A}$ als Büchi Automat, wenn man von der Büchi-Akzeptanz ausgeht.

4.2 Muller Automat

Ein Muller Automat $\mathcal{A}$ ist ein endlicher, deterministischer Automat mit Muller Akzeptanzbedingung und einer Menge $\mathcal{T} \in 2^Q$, genannt die Tabelle des Automaten (anstatt der Menge F). Dabei wird ein Wort $w \in \Sigma^\omega$ akzeptiert genau dann, wenn es einen entsprechenden Pfad p gibt mit $\text{Inf}(p) \in \mathcal{T}$, wobei $\text{Inf}(p)$ die Menge der unendlich oft besuchten Zustände ist.

Wir schreiben $\mathcal{A} = (Q, \Sigma, E, i, \mathcal{T})$.

4.3 Rabin Automat

Ein Rabin Automat ist ein Tuple $\mathcal{A} = (Q, \Sigma, E, i, \mathcal{R})$, wobei (Q, Σ, E) ein deterministischer Automat ist, i ist der Startzustand und $\mathcal{R} = \{(L_j, U_j) | j \in J\}$ ist eine Familie von Paren von Zustandsmengen. Ein Pfad p ist erfolgreich, wenn er in i beginnt und wenn es einen Index $j\ inJ$ gibt, so dass p unendlich oft U_j besucht und nur endlich oft L_j. Ist der Automat endlich, so ist dies äquivalent mit

$$\text{Inf}(p) \cap L_j = \emptyset \text{ und } \text{Inf}(p) \cap U_j \neq \emptyset.$$

4.4 Staiger Wagner Klasse zu $\mathcal{K}$

5 Operationen: von $*$-Sprache K zu ω-Sprache $L_\omega(K)$

5.1 ...

a) * alle Sprachen $K\dot{\Sigma}^\omega = \text{ext}(K)$, $K \in \mathcal{K}$
* offene G
* Staiger Wagner Klasse http://de.wikipedia.org/wiki/Staiger-Wagner-Automat Erich Grädel, Wolfgang Thomas und Thomas Wilke (Herausgeber), Automata, Logics, and Infinite Games, LNCS 2500, 2002, Seite 20 (auf englisch) http://www.automata.rwth-aachen.de/material/skripte/areas-english.pdf - s.53

a') dual $\overline{K} = \omega$-Wörter, deren alle Präfixe in K sind

b) Sprachen $\lim \mathcal{K}$ BC Muller-erkennbare (BC: boolean closure ?)

b') von einer Stelle an alle Prefixe in K

c) Kleene-Closure

alle der Form $\cup_{i=1}^n U_i \dot{V}_i^\omega$, $U_i, V_i \in \mathcal{K}$

d) $\mathcal{K}$ nicht suffix sensitiv

$K \in \mathcal{K} \Rightarrow K\dot{\Sigma}^* \in \mathcal{K}$

Hauptfrage: Für welche $\mathcal{K}$ ergibt sich eine andere Sprache als bei $\mathcal{K} = \text{Reg}$.

6 $*$-Sprachklassen

6.1 regular

6.2 piece-wise testable

6.3 k-locally testable

6.4 dot-depth-n

6.5 starfree

6.6 locally modulo testable

6.7 R-trivial

6.8 endlich / co-endlich

6.9 endwise testable

7 ω-Sprachklassen

7.1 Staiger Wagner Klasse zu $\mathcal{K}$

8 Operationen: von $*$-Sprache K zu ω-Sprache $L_\omega(K)$

8.1 ...

a) * alle Sprachen $K\dot{\Sigma}^\omega = \text{ext}(K)$, $K \in \mathcal{K}$
* offene G
* Staiger Wagner Klasse http://de.wikipedia.org/wiki/Staiger-Wagner-Automat Erich Grädel, Wolfgang Thomas und Thomas Wilke (Herausgeber), Automata, Logics, and Infinite Games, LNCS 2500, 2002, Seite 20 (auf englisch) http://www.automata.rwth-aachen.de/material/skripte/areas-english.pdf - s.53

a') dual $\overline{K} = \omega$-Wörter, deren alle Präfixe in K sind
b) Sprachen $\lim \mathcal{K}$ BC Muller-erkennbare (BC: boolean closure ?)
b') von einer Stelle an alle Prefixe in K
c) Kleene-Closure
alle der Form $\cup_{i=1}^n U_i \dot{V}_i^\omega$, $U_i, V_i \in \mathcal{K}$
d) $\mathcal{K}$ nicht suffix sensitiv
$K \in \mathcal{K} \Rightarrow K\dot{\Sigma}^* \in \mathcal{K}$

9 Lemmas

9.1 piece-wise testable

Theorem 9.1.

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\text{piece-wise testable}) = \mathrm{BC}\lim\mathcal{L}^*(\text{piece-wise testable})$$

Proof. L piece-wise testable $\Leftrightarrow$ L is a boolean algebra of $\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*$

$\subseteq$: It is sufficient to show $\mathrm{ext}(\mathcal{L}^*(\text{piece-wise testable})) \subseteq \mathrm{BC}\lim\mathcal{L}^*(\text{piece-wise testable})$. By complete induction:

$$\begin{aligned}
&\mathrm{ext}(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*) = \Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^\omega = \lim(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*)\\
&\mathrm{ext}(\neg(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*)) = \Sigma^\omega = \lim(\Sigma^*)\\
&\mathrm{ext}(\emptyset) = \emptyset = \lim(\emptyset)
\end{aligned}$$

It is sufficient to show negation only for such ground terms because we can always push the negation down.

$$\begin{aligned}
\mathrm{ext}(A \cup B) &= \mathrm{ext}(A) \cup \mathrm{ext}(B)\\
\mathrm{ext}(A \cap B) &= \mathrm{ext}(A) \cap \mathrm{ext}(B)
\end{aligned}$$

This makes the induction complete.

$\supseteq$: It is sufficient to show $\lim(\mathcal{L}^*(\text{piece-wise testable})) \subseteq \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\text{piece-wise testable})$.

$$\lim(\emptyset) = \mathrm{ext}(\emptyset), \lim(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*) = \mathrm{ext}(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*) \qquad \text{(see above)}$$

$$\begin{aligned}
\lim(\neg(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*)) &= \{\alpha \in \Sigma^\omega | \exists^\omega n : \alpha[0,n] \notin \Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*\}\\
&= \{\alpha \in \Sigma^\omega | \forall n : \alpha[0,n] \notin \Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*\}\\
&= \neg\,\mathrm{ext}(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*)
\end{aligned}$$

$$\begin{aligned}
\lim(A \cup B) &= \{\alpha \in \Sigma^\omega | \exists^\omega n : \alpha[0,n] \in A \cup B\} && = \lim(A) \cup \lim(B)\\
\lim(A \cap B) &= \{\alpha \in \Sigma^\omega | \exists^\omega n : \alpha[0,n] \in A \cap B\}
\end{aligned}$$

and because A, B are piece-wise testable

$$= \{\alpha \in \Sigma^\omega | \exists n : \forall m > n : \alpha[0,m] \in A \cap B\} \qquad = \lim(A) \cap \lim(B)$$

$\square$

Literatur

ooo