

endliche Sprache -> unendliche Sprache -> endliche Sprache

Albert Zeyer

10. Februar 2011

Inhaltsverzeichnis

1	Introduction	3
2	Automat	4
2.1	Pfad	4
2.2	Akzeptanz von endlichen Wörtern	4
3	*-Sprachklassen	4
3.1	reguläre Sprachen	4
3.2	piece-wise testable	5
3.3	k -locally testable	5
3.4	dot-depth- n	5
3.5	starfree	5
3.6	locally modulo testable	5
3.7	R -trivial	5
3.8	endlich / co-endlich	5
3.9	endwise testable	5
4	ω-Sprachklassen	5
4.1	Büchi Automat	5
4.2	Muller Automat	5
4.3	Rabin Automat	5
4.4	Staiger Wagner Klasse zu	
	5
5	Operationen: von *-Sprache K zu ω-Sprache $L_\omega(K)$	5
5.1	5
6	*-Sprachklassen	7
6.1	regular	7
6.2	piece-wise testable	7
6.3	k -locally testable	7

6.4	dot-depth- n			7
6.5	starfree			7
6.6	locally modulo testable			7
6.7	R -trivial			7
6.8	endlich / co-endlich			7
6.9	endwise testable			7
7	ω-Sprachklassen			7
7.1	Staiger	Wagner	Klasse	zu
				7
8	Operationen: von $*$-Sprache K zu ω-Sprache $L_\omega(K)$			7
8.1	...			7

1 Introduction

...

2 Automat

Ein Automat 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: $\text{Automat} = (Q, \Sigma, E, I, F)$.

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

Der Automat ist deterministisch, wenn E eine Menge von Funktionen $Q \times A \rightarrow Q$ 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 ist eine Folge von aufeinanderfolgenden Transitionen, geschrieben als: $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \dots$

2.2 Akzeptanz von endlichen Wörtern

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

Die Sprache $L^*(\text{Automat})$ ist definiert als die Menge aller Wörter, die von Automat 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 $= (Q, \Sigma, E, I, F)$ Büchi-akzeptiert ein Wort $w = (a_0, a_1, a_2, \dots) \in \Sigma^\omega$ genau dann, wenn es einen unendlichen Pfad $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \dots$ gibt mit $q_0 \in I$ und $\{q_i | q_i \in F\}$ unendlich, also der unendlich oft einen Zustand F erreicht.

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

Man bezeichnet einen Automaten als Büchi Automat, wenn man von der Büchi-Akzeptanz ausgeht.

4.2 Muller Automat

Ein Muller Automat ist ein endlicher, deterministischer Automat mit Muller Akzeptanzbedingung und einer Menge $\in 2^Q$, 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 $(p) \in$, wobei $Inf(p)$ die Menge der unendlich oft besuchten Zustände ist.

Wir schreiben $= (Q, \Sigma, E, i,)$.

4.3 Rabin Automat

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

4.4 Staiger Wagner Klasse zu

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

5.1 ...

a) $*$ alle Sprachen $K \dot{\Sigma}^\omega = (K)$, $K \in$

* 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 $\lim BC$ Muller-erkennbare (BC: boolean closure ?)

b') von einer Stelle an alle Prefixe in K

c) Kleene-Closure

alle der Form $\bigcup_{i=1}^n U_i V_i^\omega$, $U_i, V_i \in$

d) nicht suffix sensitiv

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

Hauptfrage: Für welche ergibt sich eine andere Sprache als bei $=$.

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

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

8.1 ...

a) * alle Sprachen $K\dot{\Sigma}^\omega = (K)$, $K \in$

* 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 $\lim BC$ Muller-erkennbare (BC: boolean closure ?)

b') von einer Stelle an alle Präfixe in K

c) Kleene-Closure

alle der Form $\bigcup_{i=1}^n U_i V_i^\omega$, $U_i, V_i \in$

d) nicht suffix sensitiv

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

Literatur