

# Contributions to the structure theory of $\omega$ -languages

Albert Zeyer

10. Februar 2011

## Inhaltsverzeichnis

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Automat</b>	<b>4</b>
2.1	Pfad . . . . .	4
2.2	Akzeptanz von endlichen Wörtern . . . . .	4
<b>3</b>	<b>*-Sprachklassen</b>	<b>4</b>
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
<b>4</b>	<b><math>\omega</math>-Sprachklassen</b>	<b>5</b>
4.1	Büchi Automat . . . . .	5
4.2	Muller Automat . . . . .	5
4.3	Rabin Automat . . . . .	5
4.4	Staiger Wagner Klasse zu $\mathcal{K}$ . . . . .	6
<b>5</b>	<b>Operationen: von *-Sprache <math>K</math> zu <math>\omega</math>-Sprache <math>L_\omega(K)</math></b>	<b>6</b>
5.1	... . . . .	6
<b>6</b>	<b>*-Sprachklassen</b>	<b>7</b>
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

<b>7</b>	<b><math>\omega</math>-Sprachklassen</b>	<b>7</b>
7.1	Staiger Wagner Klasse zu $\mathcal{K}$ . . . . .	7
<b>8</b>	<b>Operationen: von *-Sprache <math>K</math> zu <math>\omega</math>-Sprache <math>L_\omega(K)</math></b>	<b>7</b>
8.1	... . . . .	7
<b>9</b>	<b>Lemmas</b>	<b>8</b>
9.1	piece-wise testable . . . . .	8
9.2	extension of FO[+1] . . . . .	8

# 1 Introduction

...

## 2 Automat

Ein **Automat**  $\mathcal{A}$  auf dem Alphabet  $\Sigma$  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  $\Sigma$  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  $\mathcal{A}$  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  $\mathcal{A} = (Q, \Sigma, E, I, F)$  **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  $L^*(\mathcal{A})$  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 $\omega$ -Sprachklassen

### 4.1 Büchi Automat

Ein Automat  $\mathcal{A} = (Q, \Sigma, E, I, F)$  **Büchi-akzeptiert** ein Wort  $w = (a_0, a_1, a_2, \dots) \in \Sigma^\omega$  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  $L^\omega(\mathcal{A})$  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  $(Q, \Sigma, E)$  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  $j \in J$  gibt, so dass  $p$  unendlich oft  $U_j$  besucht und nur endlich oft  $L_j$ . Ist der Automat endlich, so ist dies äquivalent mit

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

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

### 5 Operationen: von \*-Sprache $K$ zu $\omega$ -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 $\omega$ -Sprachklassen

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

## 8 Operationen: von \*-Sprache $K$ zu $\omega$ -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  $\lim_{\mathcal{K}} \text{BC}$  Muller-erkennbare (BC: boolean closure ?)

b') von einer Stelle an alle Präfixe 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.**

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

*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  $\text{ext}(\mathcal{L}^*(\text{piece-wise testable})) \subseteq \text{BC lim } \mathcal{L}^*(\text{piece-wise testable})$ .

By complete induction:

$$\text{ext}(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*) = \Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^\omega = \lim(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*)$$

$$\text{ext}(\neg(\Sigma^* a_1 \Sigma^* a_2 \cdots a_n \Sigma^*)) = \Sigma^\omega = \lim(\Sigma^*)$$

$$\text{ext}(\emptyset) = \emptyset = \lim(\emptyset)$$

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

$$\text{ext}(A \cup B) = \text{ext}(A) \cup \text{ext}(B)$$

$$\text{ext}(A \cap B) = \text{ext}(A) \cap \text{ext}(B)$$

This makes the induction complete.

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

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

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

$$\lim(A \cup B) = \{\alpha \in \Sigma^\omega \mid \exists^\omega n: \alpha[0, n] \in A \cup B\} = \lim(A) \cup \lim(B)$$

$$\lim(A \cap B) = \{\alpha \in \Sigma^\omega \mid \exists^\omega n: \alpha[0, n] \in A \cap B\}$$

and because  $A, B$  are piece-wise testable

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

□

### 9.2 extension of FO[+1]

**Theorem 9.2.**

$$\mathcal{L}^\omega(\text{FO}[+1]) = \text{BC ext } \mathcal{L}^*(\text{FO}[+1])$$

*Proof.* The statement “sphere  $\sigma \in \Sigma^+$  occurs  $\geq n$  times” can be expressed by a sentence of the form

$$\psi := \exists \overline{x_1} \cdots \exists \overline{x_n} \phi(\overline{x_1}, \dots, \overline{x_n})$$

where each  $\overline{x_i}$  is a  $|\sigma|$ -tuple of variables and the formula  $\phi$  states: ...

□



## Literatur

ooo