Language Operations and a Structure Theory of ω-Languages

Diploma Thesis
in Computer Science

by
Albert Zeyer

submitted to the
Faculty of Mathematics, Computer Science and Natural Science of
RWTH Aachen University

July 2012

Supervisor: Prof. Dr. Dr.h.c. Wolfgang Thomas
Second examiner: PD Dr. Christof Löding

written at the
Chair of Computer Science 7
Logic and Theory of Discrete Systems
Prof. Dr. Dr.h.c. Wolfgang Thomas

Erklärung

Hiemit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, den 10. Juli 2012

Contents

Chapter 1

Introduction

The study of formal languages and finite-state automata theory is very old and fundamental in theoretical computer science. Regular expressions were introduced by Kleene in 1956 ([Kle56]). Research on the connection between formal languages, automata theory and mathematical logic began in the early 1960's by Büchi ([Büc60]). Good introductions into the theory are [Str94] and [Tho96].

We call languages over finite words the $*$-languages. Likewise, ω-languages are over infinite words.

The class of regular $*$-languages is probably the most well studied language class. Its expressiveness is exactly equivalent to the class of finite-state automata. For many applications, less powerful subsets of the regular $*$-languages are interesting, like star-free $*$-languages, locally testable $*$-languages, etc., as well as more powerful supersets, like context-free $*$-languages.

The research on ω-languages and their connection to finite-state automata began a bit later by Büchi [Büc62] and [Mul63]. As for the $*$-languages, the most well studied ω language class are the regular ω-languages. Good introductions into these theories are [Tho90], [Tho10], [Sta97] and [PP04].

The acceptance-condition in automata for $*$-languages is straight-forward. If we look at ω-languages, several different types of automata and their acceptance have been thought of, like Büchi-acceptance or Muller-acceptance, or E-acceptance and A-acceptance.

For all types, we can also argue with equivalent language-theoretical operators which operate on a $*$-language, like $\lim$ or ext. We will study the equivalences in more detail.

Depending on the $* \to \omega$ language operator or the ω-automaton acceptance condition, we get different ω-language classes. This was studied earlier already in detail for the class of regular $*$-languages. E.g., we get the result $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{reg}) \subsetneq \mathrm{BC}\lim\mathcal{L}^*(\mathrm{reg})$ and $\lim \cap \overline{\lim}\,\mathcal{L}^*(\mathrm{reg}) = \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{reg})$.

When we look at other $*$-language classes and the different ways to transform them into ω-languages, we can get different results. E.g., $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}(\mathrm{PT}) = \mathrm{BC}\lim\mathcal{L}(\mathrm{PT})$. This study is the main topic of this thesis.

Chapter 2

Background results on regular ω-languages

2.1 Preliminaries

We introduce some common terminonoly used in this thesis.

The set of natural numbers $1, 2, 3, \ldots$ is denoted by $\mathbb{N}$, likewise $0, 1, 2, 3, \ldots$ by $\mathbb{N}_0$.

An **alphabet** is a finite set of **symbols**. We usually denote an alphabet by Σ and its elements by $a, b, c, \ldots$. A finite sequence of elements in Σ is also called a **finite word**, often named $u, v, w, \ldots$. The set of such words, including the **empty word** ϵ, is denoted by Σ^*. Likewise, Σ^+ is the set of non-empty words. Infinite sequences over Σ are called **infinite words**, often named α, β. The set of such infinite words is denoted by Σ^ω.

A subset $L \subseteq \Sigma^*$ is called a **language** of finite words or also called a $*$**-language**. Likewise, a subset $L^\omega \subseteq \Sigma^\omega$ is called an ω**-language**.

A set $\mathcal{L}$ of $*$-languages is called a $*$**-language class**. Likewise, a set $\mathcal{L}^\omega$ of ω-languages is called a ω**-language class**.

We can **concatenate** finite words with each other and also finite words with infinite words. For languages $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^*$, $L_3^\omega \subseteq \Sigma^\omega$, we define the concatenation $L_1 \cdot L_2 := \{v \cdot w \mid v \in L_1, w \in L_2\}$ and $L_1 \cdot L_3^\omega := \{v \cdot \alpha \mid v \in L_1, \alpha \in L_3^\omega\}$. Exponentation of languages is defined naturally: For $L \subseteq \Sigma^*$, we define $L^0 := \{\epsilon\}$ and $L^{i+1} := L^i \cdot L$ for all $i \in \mathbb{N}_0$. The union of all such sets, is called the **Kleene closure** or the **Kleene star** operator, defined as $L^* := \cup_{i \in \mathbb{N}_0} L^i$. The **positive Kleene closure** is defined as $L^+ := \cup_{i \in \mathbb{N}} L^i$. The **infinite Kleene closure** is defined by $L^\omega := \{w_1 \cdot w_2 \cdot w_3 \cdots \mid w_i \in L\}$.

2.2 The class of regular $*$-languages

A **regular expression** is representing a language over an alphabet Σ. Regular expressions are defined recursively based on the ground terms $\emptyset$, ϵ and a for $a \in \Sigma$ denoting the languages $\emptyset$, $\{\epsilon\}$ and $\{a\}$. Then, if r and s are regular expressions representing $R, S \subseteq \Sigma^*$, then also $r + s$ (written also as $r|s$, $r \vee s$, $r \cup s$), rs (written also as $r \cdot s$) and r^* are regular expressions, representing $R \cup S$, $R \cdot S$ and R^*. Let $\mathcal{L}^*(\text{RE})$ be the set of languages which can be represented as regular expressions.

We extend these expressions also by $r \wedge s$ (written also as $r \cap s$) and $-r$ (written also as $\neg r$), representing the language $R \cap S$ and $-R := \{w \in \Sigma^* \mid w \notin R\}$. Some basic result of the study of formal languages, as can be seen in e.g. [Str94], is the equivalence of the class of these extended regular expression languages and $\mathcal{L}^*(\text{RE})$.

A **non-deterministic finite-state automaton** (NFA) $\mathcal{A}$ over an alphabet Σ is given by a finite set Q of **states** and a subset $\Delta \subseteq Q \times \Sigma^* \times Q$ of **transitions**. In most cases we also have an **initial states** $q_0 \in Q$ and a subset $F \subseteq Q$ of **final states**.

We write:

$$\mathcal{A} = (Q, \Sigma, q_0, \Delta, F).$$

The automaton is **deterministic** (a DFA) iff Δ is a function $Q \times \Sigma \to Q$. In that case, we often call the function δ and we write

$$\mathcal{A} = (Q, \Sigma, q_0, \delta, F).$$

Two transitions $(p, a, q), (p', a', q') \in E$ are **consecutive** iff $q = p'$.

A **run** in the automaton $\mathcal{A}$ is a finite sequence of consecutive transitions, written as:

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \dots$$

An automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ **accepts** a finite word $w = (a_0, a_1, \dots, a_n) \in \Sigma^*$ iff there is a run $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \cdots \xrightarrow{a_n} q_{n+1}$ with $q_0 \in I$ und $q_{n+1} \in F$.

The $*$-language $L^*(\mathcal{A})$ is defined as set of all finite words which are accepted by $\mathcal{A}$.

The set of $*$-languages accepted by a NFA is called $\mathcal{L}^*(\text{NFA})$. Likewise, $\mathcal{L}^*(\text{DFA})$ is the set of $*$-languages accepted by a DFA. A basic result (see for example [Str94] or [PP04]) is

$$\mathcal{L}^*(\text{DFA}) = \mathcal{L}^*(\text{NFA}) = \mathcal{L}^*(\text{RE}).$$

This class of $*$-languages is called the class of **regular $*$-languages**. We name it $\mathcal{L}^*(\text{reg})$ from now on.

2.3 The class of regular ω-languages

The class of regular ω-languages can be defined in many different ways. We will use one common definition and show some equivalent descriptions.

$$\mathcal{L}^\omega(\text{reg}) := \left\{ \bigcup_{i=1}^{n} U_i \cdot V_i^\omega \;\middle|\; U_i, V_i \in \mathcal{L}^*(\text{reg}), n \in \mathbb{N}_0 \right\}$$

This is also called the **Kleene closure**.

2.3.1 ω regular expressions

For a regular expression r representing a $*$-language $R \subseteq \Sigma^*$, we can introduce a corresponding ω regular expression r^ω which represents the ω-language R^ω. This ω regular expression can be combined with other ω regular expressions as usual and prefixed by standard regular expressions. We call all these combinations ω regular expressions.

We see that $\mathcal{L}^\omega(\text{reg})$ is closed under union (obviously), intersection and complement.

Thus, the class of languages accepted by ω regular expressions is exactly $\mathcal{L}^\omega(\text{reg})$.

2.3.2 ω-automata

A different, very common description is in terms of automata.

An automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ **Büchi-accepts** an infinite word $\alpha = (a_0, a_1, a_2, ...) \in \Sigma^\omega$ iff there is an infinite run $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \cdots$ in $\mathcal{A}$ with $\{i \in \mathbb{N}_0 \mid q_i \in F\}$ infinite, i.e. which reaches a state in F infinitely often.

The language $L^\omega(\mathcal{A})$ is defined as the set of all infinite words which are Büchi-accepted by $\mathcal{A}$. To make clear that we use the Büchi acceptance condition, we sometimes will also write $L^\omega_{\text{Büchi}}(\mathcal{A})$.

A basic result of the study of this language class is: The set of all languages accepted by a non-deterministic Büchi automaton is exactly $\mathcal{L}^\omega(\text{reg})$ (see [Tho10] or others). Deterministic Büchi automata are less powerful, e.g. they cannot recognise $(a+b)^* b^\omega$.

There are some different forms of ω-automata which differ in their acceptance condition. Noteable are the **Muller condition**, the **Rabin condition**, the **Streett condition** and the **Parity condition**. With such an acceptance condition, we call it **Muller automaton**, etc. The *main theorem of deterministic ω-automata* states:

- Non-deterministic Büchi automata,
- a boolean combination of deterministic Büchi automata,
- deterministic Muller automata,
- deterministic Rabin automata,
- deterministic Streett automata,
- deterministic Parity automata

all recognize the same languages. See [Tho10], [Tho96], [PP04] and others. The main part of this theorem is the **McNaughton's Theorem** which states the equivalence of non-deterministic Büchi automata and deterministic Muller automata.

Muller automata are interesting for us in the rest of this thesis. The acceptance component of a Muller automaton is a set $\mathcal{F} \subseteq 2^Q$, also called the **table** of the automaton (instead of a single set $F \subseteq Q$). A word $w \in \Sigma^\omega$ is accepted iff there is an infinite run ρ with $\mathrm{Inf}(\rho) \in \mathcal{F}$, where $\mathrm{Inf}(\rho)$ is the set of infinitely often reached states of the run ρ.

We write:

$$\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{F}).$$

2.3.3 Language operators

Büchi acceptance is closely connected to the language operator

$$\lim(L) := \{\alpha \in \Sigma^\omega \mid \exists^\omega n \colon \alpha[0,n] \in L\}.$$

We define the language class operator

$$\lim(\mathcal{L}) := \{\lim(L) \mid L \in \mathcal{L}\}.$$

We see that $\lim(\mathcal{L}^*(\mathrm{reg}))$ is equal to the languages accepted by deterministic Büchi automata ([Tho10]). Thus:

$$\mathrm{BC} \lim \mathcal{L}^*(\mathrm{reg}) = \mathcal{L}^\omega(\mathrm{reg}),$$

where BC means all boolean combinations (union, intersection, complement).

Another classification is

$$\mathcal{L}^\omega(\mathrm{reg}) = \left\{ \bigcup_{i=0}^{n} U_i \cdot \lim V_i \;\middle|\; U_i, V_i \in \mathcal{L}^*(\mathrm{reg}), n \in \mathbb{N}_0 \right\}.$$

2.3.4 Logic on infinite words

Let $L_2(\Sigma)$ be the set of formulas $\mathrm{MSO}(<)$ over Σ. The interpretation of such formulas over infinite words is straight-forward. In [Tho81], we can see that

$$\mathcal{L}^\omega(\mathrm{reg}) = \{A \subseteq \Sigma^\omega \mid A \text{ definable in } L_2(\Sigma)\} \,.$$

2.3.5 Some properties

Lemma 2.1. $\lim \mathcal{L}^*(reg)$ *is closed under intersection.*

Proof. In [AH04, Chapter 12, Remark 12.4], this is shown via a special product automata construction of deterministic Büchi automata. □

2.4 Language Operators: Transformation of $*$-languages to ω-languages

We already introduced $\lim$. We can define a family of language operators, partly also derived from the study of $\mathcal{L}^\omega(\mathrm{reg})$. Some of these operators operate on a single language and not on the class. Let $\mathcal{L}$ be a $*$-language class. Let $L \in \mathcal{L}$.

1. $\mathrm{ext}(L) := \{\alpha \in \Sigma^\omega \mid \exists n\colon \alpha[0,n] \in L\} = L \cdot \Sigma^\omega$
2. $\overline{\mathrm{ext}}(L) := \{\alpha \in \Sigma^\omega \mid \forall n\colon \alpha[0,n] \in L\}$ (also called the dual-ext)
3. $\mathrm{BC}\,\mathrm{ext}$
4. $\lim(L) := \{\alpha \in \Sigma^\omega \mid \forall N\colon \exists n > N\colon \alpha[0,n] \in L\} = \{\alpha \in \Sigma^\omega \mid \exists^\omega n\colon \alpha[0,n] \in L\}$
5. $\overline{\lim}(L) := \{\alpha \in \Sigma^\omega \mid \exists N\colon \forall n > N\colon \alpha[0,n] \in L\}$ (also called dual-lim)
6. $\mathrm{BC}\lim$
7. $\widehat{\mathrm{Kleene}}(\mathcal{L}) := \{\bigcup_{i=1}^n U_i \cdot V_i^\omega \mid U_i, V_i \in \mathcal{L}, n \in \mathbb{N}_0\}$
8. $\widehat{\lim}(\mathcal{L}) := \{\bigcup_{i=1}^n U_i \cdot \lim V_i \mid U_i, V_i \in \mathcal{L}, n \in \mathbb{N}_0\}$

From language operators, we get language class operators in a canonical way, e.g. $\lim(\mathcal{L}) := \{\lim L \mid L \in \mathcal{L}\}$. BC denotes always all boolean combinations (union, intersection, complement) of a language class, i.e. for $\mathcal{X} \subseteq \Sigma^\omega \,\dot\cup\, \Sigma^*$, $\mathrm{BC}\,\mathcal{X}$ is defined as the smallest set such that

- $\mathcal{X} \subseteq \mathrm{BC}\,\mathcal{X}$,
- $-X \in \mathrm{BC}\,\mathcal{X}$ for all $X \in \mathrm{BC}\,\mathcal{X}$,
- $X_1 \cup X_2 \in \mathrm{BC}\,\mathcal{X}$ for all $X_1, X_2 \in \mathrm{BC}\,\mathcal{X}$,
- $X_1 \cap X_2 \in \mathrm{BC}\,\mathcal{X}$ for all $X_1, X_2 \in \mathrm{BC}\,\mathcal{X}$.

For ext and $\overline{\mathrm{ext}}$, we can also introduce equivalent ω automata acceptance conditions (as in [Tho10]). Let $L \subseteq \Sigma^*$ be a regular $*$-language and $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ be an automaton which accepts exactly L. Let ρ be an infinte run in $\mathcal{A}$.

- $\mathcal{A}$ **E-accepts** ρ iff $\exists i\colon \rho(i) \in F$,
- $\mathcal{A}$ **A-accepts** ρ iff $\forall i\colon \rho(i) \in F$.

We define

$$L_E^\omega(\mathcal{A}) := \{\alpha \in \Sigma^\omega \mid \alpha \text{ is E-accepted in } \mathcal{A}\},$$
$$L_A^\omega(\mathcal{A}) := \{\alpha \in \Sigma^\omega \mid \alpha \text{ is A-accepted in } \mathcal{A}\},$$

and we have the equalities

$$L_E^\omega(\mathcal{A}) = \mathrm{ext}(L),$$
$$L_A^\omega(\mathcal{A}) = \overline{\mathrm{ext}}(L).$$

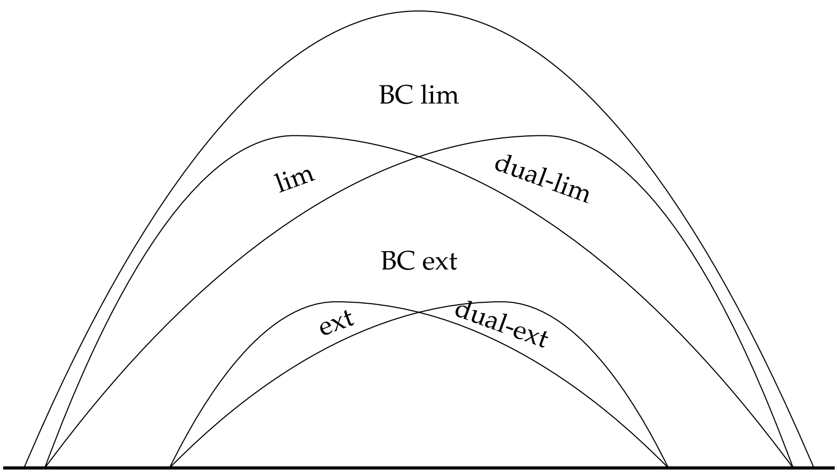
Note that $\mathcal{A}$ can be both deterministic or non-deterministic for this property (see 3.2).

Given these language operators, we are interested in the relations between them. For the class $\mathcal{L}^*(\mathrm{reg})$ of regular languages, we already know that

$$\mathcal{L}^\omega(\mathrm{reg}) = \widehat{\mathrm{Kleene}}(\mathcal{L}^*(\mathrm{reg})) = \mathrm{BC}\lim(\mathcal{L}^*(\mathrm{reg})) = \widehat{\lim}(\mathcal{L}^*(\mathrm{reg})).$$

2.5 Classification of regular ω-languages

Considering $\mathcal{R} := \mathcal{L}^*(\mathrm{reg})$, we get a language diagram like:



where all inclusions are strict. In more detail:

Lemma 2.2. *1* $\mathrm{ext}\,\mathcal{R} \cap \overline{\mathrm{ext}}\,\mathcal{R} \neq \emptyset$

2a. $\mathrm{ext}\,\mathcal{R} \cap \overline{\mathrm{ext}}\,\mathcal{R} \subsetneq \mathrm{ext}\,\mathcal{R}$

2b. $\operatorname{ext}\mathcal{R} \cap \overline{\operatorname{ext}}\,\mathcal{R} \subsetneqq \overline{\operatorname{ext}}\,\mathcal{R}$

3. $\operatorname{ext}\mathcal{R} \neq \overline{\operatorname{ext}}\,\mathcal{R}$

4. $\operatorname{ext}\mathcal{R} \cup \overline{\operatorname{ext}}\,\mathcal{R} \subsetneqq \operatorname{BC}\operatorname{ext}\mathcal{R}$

5. $\operatorname{BC}\operatorname{ext}\mathcal{R} = \lim\mathcal{R} \cap \overline{\lim}\,\mathcal{R}$ *(Staiger-Wagner class)*

6a. $\lim\mathcal{R} \cap \overline{\lim}\,\mathcal{R} \subsetneqq \lim\mathcal{R}$

6b. $\lim\mathcal{R} \cap \overline{\lim}\,\mathcal{R} \subsetneqq \overline{\lim}\,\mathcal{R}$

7. $\lim\mathcal{R} \neq \overline{\lim}\,\mathcal{R}$

8. $\lim\mathcal{R} \cup \overline{\lim}\,\mathcal{R} \subsetneqq \operatorname{BC}\lim\mathcal{R}$

and we have the additional properties

9. $\operatorname{BC}\lim\mathcal{R} = \widehat{\operatorname{Kleene}}(\mathcal{R})$

10. $\operatorname{BC}\lim\mathcal{R} = \widehat{\lim}(\mathcal{R})$

11. $\operatorname{BC}\lim\mathcal{R} = \left\{ L^{\omega}_{\textit{Büchi}}(\mathcal{A}) \,\middle|\, \mathcal{A} \textit{ automaton so that } L^*(\mathcal{A}) \in \mathcal{R} \right\}$

Proof. 1 $\tilde{L}_1 := a\Sigma^\omega \in \operatorname{ext} \cap \overline{\operatorname{ext}}\,\mathcal{R}$ with $\tilde{L}_1 = \operatorname{ext}(a)$ and $\tilde{L}_1 = \overline{\operatorname{ext}}(a\Sigma^*)$. ([Tho10, prop, p.38])

2a. $\tilde{L}_{2a} := \operatorname{ext}(a^*b) = a^*b\Sigma^\omega \in \operatorname{ext}\mathcal{R}$. Assume some A-automaton $\mathcal{A}$ with n states accepts $\tilde{L}_{2a}$. $\mathcal{A}$ would also accept $a^n b^\omega$. I.e. the $(n+1)$th state after the run of a^n would also accept a, i.e. $\mathcal{A}$ would accept a^{n+1}. By inclusion, $\mathcal{A}$ would accept a^ω. That is a contradiction. Thus, there is no such A-automat. Thus, $\tilde{L}_{2a} \notin \overline{\operatorname{ext}}\,\mathcal{R}$.

2b. $\tilde{L}_{2b} := -\tilde{L}_{2a} \in \overline{\operatorname{ext}}\,\mathcal{R}$, $\tilde{L}_{2b} \notin \operatorname{ext}\mathcal{R}$.

3. Follows directly from P2a and P2b.

4. $\tilde{L}_4 := \Sigma^* a \Sigma^\omega \cap -(\Sigma^* b \Sigma^\omega)$, $\Sigma = \{a, b, c\}$. Then we have $\tilde{L}_4 \notin \operatorname{ext} \cup \overline{\operatorname{ext}}\,\mathcal{R}$, $\tilde{L}_4 \in \operatorname{BC}\operatorname{ext}\mathcal{R}$. ([Tho10, p.38])

5. A language in this class is also said to have the **obligation property**. Staiger and Wagner have introduced a **Staiger-Wagner automaton** (also called a **weak Muller automaton**; see definition 3.14) which can accept exactly this language class. This class of languages is called the **Staiger-Wagner-recognizable** languages. This is stated in theorem 3.15.

   A generic proof of the equality $\mathrm{BC}\,\mathrm{ext}\,\mathcal{R} = \lim\mathcal{R} \cap \overline{\lim}\,\mathcal{R}$ is given in lemma 3.16.

6a. $\tilde{L}_{6a} := \lim(\Sigma^* a) = (\Sigma^* a)^\omega$. Assume there is $L \subseteq \Sigma^*$ with $\lim(L) = -\tilde{L}_{6a}$. Let $(w_0, w_1, w_2, \dots) \in (\Sigma^*)^{\mathbb{N}}$ so that $w_0 \in L, w_0 a w_1 \in L, \dots, w_0 \prod_{i=0}^{n} a w_i \in L\ \forall n \in \mathbb{N}$. Thus, $\alpha := w_0 \prod_{i\in\mathbb{N}} a w_i \in \lim L$. But $\alpha \notin -\tilde{L}_{6a}$. That is a contradiction. Thus, $-\tilde{L}_{6a} \notin \lim\mathcal{R}$. Because $\mathcal{R}$ is closed under complement, we get $\tilde{L}_{6a} \notin \overline{\lim}\,\mathcal{R}$.

6b. Analog to 6a with $\tilde{L}_{6b} := -\tilde{L}_{6a}$.

7. Follows directly from 6a and 6b.

8. $\tilde{L}_8 := (\Sigma^* a)^\omega \cap -(\Sigma^* b)^\omega$. Then $\tilde{L}_8 \notin \lim \cup \overline{\lim}\,\mathcal{R}$ but $\tilde{L}_8 \in \mathrm{BC}\lim\mathcal{R}$. ([Tho10, prop, p.38])

9.-11. This is explained already in chapter 2.3 and in more detail in [Tho10] or [Tho81, Theorem 3.1].

□

This relation diagram was studied in detail for $\mathcal{L}^*(\mathrm{reg})$. We are interested wether we get the same properties for other $*$-language classes under the given language operators.

In chapter 3, we will reformulate many proofs of the properties given in chapter 2.2 in a generic way. The results will give us an understanding about when such ω-language class relations hold, when inclusions are strict and when they are not.

These base theorems are then used in chapter 4 to study some concrete $*$-language classes.

Chapter 3

General results

Let $\mathcal{L}$ be a $*$-language class. We start with some very basic results on language operators.

3.1 Background

Lemma 3.1. *Let* $L, A, B \in \mathcal{L}$.

1. $\operatorname{ext} L = L \cdot \Sigma^\omega$
2. $\operatorname{ext} L = \lim(L \cdot \Sigma^*)$
3. $\operatorname{ext} L = \overline{\lim}(L \cdot \Sigma^*)$
4. $-\lim(-L) = \overline{\lim}(L)$
5. $\overline{\lim}\, L \subseteq \lim L$
6. $\lim A \cup \lim B = \lim(A \cup B)$
7. $\overline{\lim}\, A \cup \overline{\lim}\, B \subseteq \overline{\lim}(A \cup B)$
   There is no equality in general: $A = (00)^*$, $B = (00)^*0$.

Proof. 1.-5. They all follow directly from the definition.

6.

$$\begin{aligned}
&\alpha \in \lim A \cup \lim B \\
\Leftrightarrow\ &\exists^\omega n\colon \alpha[0,n] \in A \ \vee\ \exists^\omega n\colon \alpha[0,n] \in B \\
\Leftrightarrow\ &\exists^\omega n\colon \alpha[0,n] \in A \cup B \\
\Leftrightarrow\ &\alpha \in \lim A \cup B
\end{aligned}$$

7.

$$
\begin{aligned}
&\alpha \in \overline{\lim}\, A \cup \overline{\lim}\, B \\
\Leftrightarrow\ &\exists N \colon \forall n \geq N \colon \alpha[0,n] \in A \ \vee\ \exists N \colon \forall n \geq N \colon \alpha[0,n] \in B \\
\Rightarrow\ &\exists N \colon \forall n \geq N \colon \alpha[0,n] \in A \cup B
\end{aligned}
$$

□

For ω automata, we already know that non-determinism can be more powerful than determinism (see 2.3): The class of non-deterministic Büchi automata can accept clearly more languages than the class of determinstic Büchi automata. E.g. $L^\omega := (a+b)^* b^\omega \in \mathcal{L}^\omega(\text{reg})$ cannot be recognised by deterministic Büchi automata, i.e. $L^\omega \notin \lim \mathcal{L}^*(\text{reg})$.

Luckily, for E- and A-acceptance, this is not the case as we see below. This matches the intuition that E/A-acceptance doesn't really tell something about infinite properties of words but Büchi/Muller does. And when talking about finite words, we already know that non-deterministic and deterministic automata are equaly powerful (see chapter 2.2).

Lemma 3.2. *The ω-language-class accepted by deterministic E-automata is equal to non-deterministic E-automata. I.e., for every non-deterministic E-automaton, we can construct an equivalent deterministic E-automaton. The same goes for A-automata.*

Proof. Let $\mathcal{A}^N$ be any non-deterministic automaton and $\mathcal{A}^D$ an ($*$-)equivalent deterministic automaton. Then:

$$
\begin{aligned}
&\alpha \in L_E^\omega(\mathcal{A}^N) \\
\Leftrightarrow\ &\exists n \colon \alpha[0,n] \in L(\mathcal{A}^N) \\
\Leftrightarrow\ &\exists n \colon \alpha[0,n] \in L(\mathcal{A}^D) \\
\Leftrightarrow\ &\alpha \in L_E^\omega(\mathcal{A}^D)
\end{aligned}
$$

$\mathcal{A}^N$ can be interpreted as an arbitary E-automata and we have shown that we get an equivalent deterministic E-automata.

For A-automata, the proof is analogue. □

We are interested in relations like $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L} \overset{?}{\subsetneq} \mathrm{BC}\lim\mathcal{L}$ or $\mathrm{ext}\,\mathcal{L} \overset{?}{\subsetneq} \lim\mathcal{L}$. With $\mathcal{L} = \{\{a\}\}$, we realize that even $\mathrm{ext}\,\mathcal{L} \subseteq \lim\mathcal{L}$ is not true in general ($\mathrm{ext}\,\{\{a\}\} = \{a\Sigma^\omega\} \neq \emptyset = \lim\{\{a\}\}$). In lemma 3.3, we see a sufficient condition for this property, though.

We want to study all the properties we have shown for $\mathcal{L}^*(\mathrm{reg})$ in lemma 2.2.

We will formulate some properties of interest in a general form for a $*$-language class $\mathcal{L}$ which all hold for $\mathcal{L}^*(\mathrm{reg})$. We get some general results based on these properties later in this chapter.

Let $L, A, B \in \mathcal{L}$. Then there are the following properties on $\mathcal{L}$:

1. **Closure under suffix-independence**: $L \cdot \Sigma^* \in \mathcal{L}$

2a. **Closure under union**: $A \cup B \in \mathcal{L}$

2b. **Closure under intersection**: $A \cap B \in \mathcal{L}$

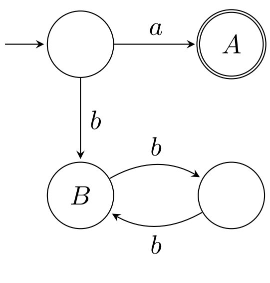
3. **Closure under complementation/negation**: $-L \in \mathcal{L}$

4. **Closure under change of final states**: In some proofs, e.g. in 3.16 or 3.17, we have an automaton based on some language of the language class and we do some modifications on it, e.g. we modify the acceptance component. If there is a way to stay in the language class, the class is called to be **closed under change of final states**. Formally:

   There is a deterministic automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ with $L^*(\mathcal{A}) = L$ such that for all $F' \subseteq Q$, we have $L^*((Q, \Sigma, q_0, \delta, F')) \in \mathcal{L}$.

   For $\mathcal{L}^*(\mathrm{reg})$, this property holds obviously.

   Note that we cannot just take any automaton. For $\mathcal{L}^*(\mathrm{FO}[<])$ and the automaton below, it does not hold:



   This is a deterministic automaton for the language $\{a\} \in \mathcal{L}^*(\mathrm{FO}[<])$. If you make B also a final state, we get the language $a + b(bb)^* \notin \mathcal{L}^*(\mathrm{FO}[<])$.

5. **Closure under alphabet permutation**: For all permutations $\sigma : \Sigma \to \Sigma$, we have $L_\sigma := \{\sigma(w) \mid w \in L\} \in \mathcal{L}$. ($\sigma$ on words is defined canonically.)

   If $L = L_\sigma$ for all permutations σ, we call L **alphabet permutation invariant**.

3.2 Classification for arbitrary language classes

Lemma 3.3. *If $\mathcal{L}$ is closed under suffix-independence,*

$$\operatorname{ext}\mathcal{L} \subseteq \lim\mathcal{L} \cap \overline{\lim}\,\mathcal{L}.$$

Proof. For $L \in \mathcal{L}$, we have $\operatorname{ext} L = L\Sigma^\omega = \lim(L\Sigma^*) = \overline{\lim}(L\Sigma^*)$. □

Lemma 3.4. *If we have* $\operatorname{ext}\mathcal{L} \subseteq \lim\mathcal{L}$*, then we also have*

$$\mathrm{BC}\operatorname{ext}\mathcal{L} \subseteq \mathrm{BC}\lim\mathcal{L}.$$

Proof. From $\operatorname{ext}\mathcal{L} \subseteq \lim\mathcal{L}$, it directly follows $\{-\operatorname{ext} L \mid L \in \mathcal{L}\} \subseteq \{-\lim L \mid L \in \mathcal{L}\}$. Thus, it also follows the claimed inequality. □

Lemma 3.5. *If we have* $\operatorname{ext}\mathcal{L} \subseteq \lim\mathcal{L}$ *and let* $\mathcal{L}$ *be closed under negation. Then we have*

$$\overline{\operatorname{ext}}\,\mathcal{L} \subseteq \overline{\lim}\,\mathcal{L}.$$

Proof. Let $L \in \mathcal{L}$. Then $\overline{\operatorname{ext}}\,L = -\operatorname{ext}(-L)$. Because of the negation closure, we also have $-L \in \mathcal{L}$ and $\operatorname{ext}(-L) \in \operatorname{ext}\mathcal{L}$.

Thus $\operatorname{ext}(-L) \in \lim\mathcal{L}$. Thus, $\overline{\operatorname{ext}}(L) = -\operatorname{ext}(-L) \in \{-\lim A \mid A \in \mathcal{L}\} = \left\{\overline{\lim}\,A \,\middle|\, -A \in \mathcal{L}\right\}$. Because of the negation closure, we have

$$\left\{\overline{\lim}\,A \,\middle|\, -A \in \mathcal{L}\right\} = \left\{\overline{\lim}\,A \,\middle|\, A \in \mathcal{L}\right\} = \overline{\lim}\,\mathcal{L}.$$

Thus,

$$\overline{\operatorname{ext}}\,L \in \overline{\lim}\,\mathcal{L}.$$

□

Note that we needed the negation closure in the proof. This is in contrast to 3.4, where it directly follows. We have to be careful about the difference $-\operatorname{ext}\mathcal{L} := \{-\operatorname{ext} L \mid L \in \mathcal{L}\} \neq \overline{\operatorname{ext}}\,\mathcal{L}$ (in general, if $\mathcal{L}$ is not closed under negation).

Analogously:

Lemma 3.6. *If we have* $\operatorname{ext}\mathcal{L} \subseteq \overline{\lim}\,\mathcal{L}$ *and let* $\mathcal{L}$ *be closed under negation. Then we have*

$$\overline{\operatorname{ext}}\,\mathcal{L} \subseteq \lim \mathcal{L}.$$

Proof. Let $L \in \mathcal{L}$. Then $\overline{\operatorname{ext}}\,L = -\operatorname{ext}(-L)$. Because of the negation closure, we also have $-L \in \mathcal{L}$ and $\operatorname{ext}(-L) \in \operatorname{ext}\mathcal{L}$.

Thus $\operatorname{ext}(-L) \in \overline{\lim}\,\mathcal{L}$. Thus, $\overline{\operatorname{ext}}(L) = -\operatorname{ext}(-L) \in \left\{-\overline{\lim}\,A \,\middle|\, A \in \mathcal{L}\right\} = \{\lim A \mid -A \in \mathcal{L}\}$. Because of the negation closure, we have

$$\{\lim A \mid -A \in \mathcal{L}\} = \{\lim A \mid A \in \mathcal{L}\} = \lim \mathcal{L}.$$

Thus,

$$\overline{\operatorname{ext}}\,L \in \lim \mathcal{L}.$$

□

Summerized:

Lemma 3.7. *Let* $\mathcal{L}$ *be closed under suffix-independence and negation. Then we have*

$$\operatorname{ext} \cup \overline{\operatorname{ext}}\,\mathcal{L} \subseteq \lim \cap \overline{\lim}\,\mathcal{L}.$$

Proof. This is lemma 3.3, 3.5 and 3.6. □

We also need the *negation closure* for the natural expected inclusion of the dual operators in their boolean closures.

Lemma 3.8. *Let* $\mathcal{L}$ *be closed under negation. Then*

$$\operatorname{ext} \cup \overline{\operatorname{ext}}\,\mathcal{L} \subseteq \mathrm{BC}\operatorname{ext}\mathcal{L},$$

$$\lim \cup \overline{\lim}\,\mathcal{L} \subseteq \mathrm{BC}\lim \mathcal{L}.$$

Proof. Because of the negation closure, we have

$$\mathrm{BC}\operatorname{ext}\mathcal{L} \supseteq \{-\operatorname{ext} A \mid A \in \mathcal{L}\} = \left\{\overline{\operatorname{ext}}\,A \,\middle|\, -A \in \mathcal{L}\right\} = \overline{\operatorname{ext}}\,\mathcal{L},$$

$$\mathrm{BC}\lim \mathcal{L} \supseteq \{-\lim A \mid A \in \mathcal{L}\} = \left\{\overline{\lim}\,A \,\middle|\, -A \in \mathcal{L}\right\} = \overline{\lim}\,\mathcal{L}.$$

□

Lemma 3.9.
- *$\mathcal{L}$ closed under union $\Rightarrow \bigcup \operatorname{ext} \mathcal{L} \subseteq \operatorname{ext} \mathcal{L}$.*
- *$\mathcal{L}$ closed under intersection $\Rightarrow \bigcap \operatorname{ext} \mathcal{L} \subseteq \operatorname{ext} \mathcal{L}$.*

Proof. Let $A, B \in \mathcal{L}$. Then we have $\operatorname{ext}(A \cup B) = \operatorname{ext}(A) \cup \operatorname{ext}(B)$ and $\operatorname{ext}(A \cap B) = \operatorname{ext}(A) \cap \operatorname{ext}(B)$. □

We present some common examples which would separate $\operatorname{ext} \cup \overline{\operatorname{ext}} \mathcal{L}$ from $\operatorname{BC} \operatorname{ext} \mathcal{L}$ and similarily $\lim \cup \overline{\lim} \mathcal{L}$ from $\operatorname{BC} \lim \mathcal{L}$.

Example 3.10. *Let $\{a, b\} \subseteq \Sigma$. Define $L_a := \Sigma^* a$, $L_b := \Sigma^* b$. Let $L_a, L_b \in \mathcal{L}$. Let $\mathcal{L}$ be closed under negation. Then*

$$\operatorname{ext} \cup \overline{\operatorname{ext}} \mathcal{L} \subsetneqq \operatorname{BC} \operatorname{ext} \mathcal{L},$$

$$\lim \cup \overline{\lim} \mathcal{L} \subsetneqq \operatorname{BC} \lim \mathcal{L}.$$

Proof. The inclusion follows from lemma 3.8.

$\tilde{L}_1 := \operatorname{ext}(L_a) \cap -\operatorname{ext}(L_b)$. Then $\tilde{L}_1 \notin \operatorname{ext} \cup \overline{\operatorname{ext}} \mathcal{L}$ but $\tilde{L}_1 \in \operatorname{BC} \operatorname{ext} \mathcal{L}$. (Lemma 2.2)

$\tilde{L}_2 := \lim(L_a) \cap -\lim(L_b)$. Then $\tilde{L}_2 \notin \lim \cup \overline{\lim} \mathcal{L}$ but $\tilde{L}_2 \in \operatorname{BC} \lim \mathcal{L}$. (Lemma 2.2) □

This exampe can be generalized a bit. We first introduce M-invariance on a language for $M \subseteq \Sigma$.

Definition 3.11. A language $L \subseteq \Sigma^* \cup \Sigma^\omega$ is called **M-invariant** for $M \subseteq \Sigma$ iff for all $w \in \Sigma^* \cup \Sigma^\omega$,

$$w \in L \quad \Leftrightarrow \quad w|_M \in L,$$

where $w|_M$ is the word w with all letters from M removed.

There is always exactly one **maximum invariant alphabet set** $M_m \subseteq \Sigma$ **of** L such that L is M_m-invariant. Then call $\Sigma - M_m$ the **non-invariant alphabet set of** L.

For the final lemma 3.13, we need another small lemma:

Lemma 3.12. *Let $L \subseteq \Sigma^*$ and let L be $\{a, b\} \subseteq \Sigma$ invariant. Then*

$$\operatorname{ext} L \notin \overline{\operatorname{ext}}\, \mathcal{L}^*(\mathit{reg}) \quad \Rightarrow \quad \operatorname{ext} L|_{\Sigma - \{a\}} \notin \overline{\operatorname{ext}}\, \mathcal{L}^*(\mathit{reg})$$

and

$$\lim L \notin \overline{\lim}\, \mathcal{L}^*(\mathit{reg}) \quad \Rightarrow \quad \lim L|_{\Sigma - \{a\}} \notin \overline{\lim}\, \mathcal{L}^*(\mathit{reg}).$$

Proof. In any automata for L (no matter if L, $\operatorname{ext} L$, $\overline{\operatorname{ext}}\, L$, $\lim L$ or $\overline{\lim}\, L$), we can assume without restriction that a, b never changes the state and is everywhere accepted. I.e. we always have the transition set $T := \{(q, \{a, b\}) \mapsto q \mid \text{for all states } q\}$. If we restrict L on $\Sigma - \{a\}$, those are all exactly the same automata with the only difference that the transition set becomes $T|_{\Sigma - \{a\}} = \{(q, \{b\}) \mapsto q \mid \text{for all states } q\}$. I.e. the set of possible automata we are interested about is isomorphic in both cases. Thus, saying that there doesn't exist some kind of automata is independent from wether we say it for L or $L|_{\Sigma - \{a\}}$. □

Theorem 3.1. *Let $\mathcal{L}$ be closed under negation and under alphabet permutation. Let $\{a, b, c\} \subseteq \Sigma$. Let there be $L_a \in \mathcal{L}$. Let $\{a\}$ be the* non-invariant alphabet set *of L_a and let L_a be $\{b, c\}$-*invariant. *Then*

$$\operatorname{ext} L_a \notin \overline{\operatorname{ext}}\, \mathcal{L}^*(\mathit{reg}) \quad \Rightarrow \quad \operatorname{ext} \cup \overline{\operatorname{ext}}\, \mathcal{L} \subsetneq \mathrm{BC} \operatorname{ext} \mathcal{L}$$

and

$$\lim L_a \notin \overline{\lim}\, \mathcal{L}^*(\mathit{reg}) \quad \Rightarrow \quad \overline{\lim} \cup \overline{\lim}\, \mathcal{L} \subsetneq \mathrm{BC} \lim \mathcal{L}.$$

Proof. Let op be either ext or $\lim$ and let $\overline{\operatorname{op}}$ be the dual version of the operator ($\overline{\operatorname{ext}}$ or $\overline{\lim}$). Assume that $\operatorname{op} L_a \notin \operatorname{op} \mathcal{L}^*(\text{reg})$.

The inclusion follows from lemma 3.8.

Define $\tilde{L}_a := \operatorname{op} L_a$. Define the alphabet permutation $\sigma := \{a \mapsto b, b \mapsto a\}$. L_a is an alphabet permutation non-invariant language because $\sigma(L) \neq L$. Define $L_b := \sigma(L_a)$. Because of the alphabet permutation closure, $L_b \in \mathcal{L}$ and $\tilde{L}_b := \operatorname{op} L_b \notin \overline{\operatorname{op}}\, \mathcal{L}^*(\text{reg})$. I.e., $-\tilde{L}_b = \overline{\operatorname{op}}(-L_b) \notin \operatorname{op} \mathcal{L}^*(\text{reg})$. Also, L_b is $\{a, c\}$-invariant and $\{b\}$ is the non-invariant alphabet set of L_b.

Then,

$$L_\omega := \tilde{L}_a \cap -\tilde{L}_b \in \mathrm{BC} \operatorname{op} \mathcal{L}.$$

Assume there is $L \in \mathcal{L}^*(\mathrm{reg})$ such that $\operatorname{op} L = L_\omega$. Then $\operatorname{op} L|_{\Sigma-\{a\}} = \overline{\operatorname{op}} - L_b|_{\Sigma-\{a\}}$. However, because of lemma 3.12, $\overline{\operatorname{op}} - L_b|_{\Sigma-\{a\}} \notin \operatorname{op} \mathcal{L}^*(\mathrm{reg})$. That is a contradiction. Thus, $L_\omega \notin \operatorname{op} \mathcal{L}^*(\mathrm{reg})$.

Assume there is $\overline{L} \in \mathcal{L}^*(\mathrm{reg})$ such that $\overline{\operatorname{op}}\, \overline{L} = L_\omega$. Then $\overline{\operatorname{op}}\, \overline{L}|_{\Sigma-\{b\}} = \operatorname{op} L_a|_{\Sigma-\{b\}}$. However, because of lemma 3.12, $\operatorname{op} L_a|_{\Sigma-\{b\}} \notin \overline{\operatorname{op}}\, \mathcal{L}^*(\mathrm{reg})$. That is a contradiction. Thus, $L_\omega \notin \overline{\operatorname{op}}\, \mathcal{L}^*(\mathrm{reg})$. □

The lemma could be generalized even more by using a generic M-non-invariant language $L \in \mathcal{L}$ with $K \subseteq \Sigma$ maximum invariant alphabet set such that $\#K > \#M$.

Definition 3.13. A **Staiger-Wagner automaton** (also called **weak Muller automaton**) is of the same form $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ with acceptance component $\mathcal{F} \subseteq 2^Q$ like a Muller automaton with the acceptance condition that a run ρ in $\mathcal{A}$ is accepting if and only if $\mathrm{Occ}(\rho) := \{q \in Q \mid q \text{ occurs in } \rho\} \in \mathcal{F}$. ([Tho10, Def.61, p.43])

Theorem 3.14. *We see that the class of Staigner-Wagner-recognized languages is exactly the class* $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathit{reg})$ *and also* $\lim \cap \overline{\lim}\, \mathcal{L}^*(\mathit{reg})$. *And thus:*

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathit{reg}) = \lim \cap \overline{\lim}\, \mathcal{L}^*(\mathit{reg}).$$

Proof. See [Tho10, Theorem 63+64, p.44]. □

We are now formulating a more general and direct proof for the $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L} = \lim \cap \overline{\lim}\, \mathcal{L}$ equality without Staiger-Wagner-automata (where some of the ideas are loosely based on [Tho10, Theorem 63+64, p.44]).

Lemma 3.15. *Let $\mathcal{L}$ be closed under suffix-independence, negation, union and change of final states. Then*

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L} = \lim \cap \overline{\lim}\, \mathcal{L}.$$

Proof. First, we show $\lim \cap \overline{\lim}\, \mathcal{L} \subseteq \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}$.

Let $\tilde{L} \in \lim \cap \overline{\lim}\, \mathcal{L}$, i.e. there are deterministic automaton $\mathcal{A}$ and $\overline{\mathcal{A}}$ so that $L^\omega_{\text{Büchi}}(\mathcal{A}) = L^\omega_{\text{co-Büchi}}(\overline{\mathcal{A}}) = \tilde{L}$. Let $Q, \overline{Q}$ be the states of $\mathcal{A}, \overline{\mathcal{A}}$. Now look at the product automaton $\mathcal{A} \times \overline{\mathcal{A}} =: \overset{\times}{\mathcal{A}}$ with states $Q \times \overline{Q}$ and final states $F \times \overline{F} \subseteq Q \times \overline{Q}$. $\overset{\times}{\mathcal{A}}$ is also deterministic.

In $\overset{\times}{\mathcal{A}}$, we have

$$\begin{aligned}
&\alpha \in \tilde{L} \\
\Leftrightarrow\ &\forall N\colon \exists n \geq N\colon \overset{\times}{\rho}(\alpha)[n] \in F \times \overline{Q} \\
\Leftrightarrow\ &\exists N\colon \forall n \geq N\colon \overset{\times}{\rho}(\alpha)[n] \in Q \times \overline{F}
\end{aligned}$$

Look at a strongly connected component (SCC) S in $\overset{\times}{\mathcal{A}}$. We have $S \cap F \times \overline{Q} \neq \emptyset$, iff S accepts. It follows that all states in S are finite states in $\overline{\mathcal{A}}$, i.e. $S \cap Q \times \overline{F} = S$.

Single $\overset{\times}{q} \in \overset{\times}{Q}$ which are not part of a SCC can be ignored. For the acceptance of infinte words, only SCCs are relevant. For S, define $S_+ := \left\{ \overset{\times}{q} \in \overset{\times}{Q} - S \,\middle|\, \overset{\times}{q} \text{ can be visited after } S \right\}$.

Then we have

$$\begin{aligned}
\tilde{L} = \bigcup_{\text{SCC } S} &S \text{ will be visited} \wedge \\
&\text{all states of } S \text{ will be visited forever after some step} \wedge \\
&S_+ \text{ will not be visited.}
\end{aligned}$$

S will be visited: Let S exactly be the finite states. This interpreted as an E-automaton $\mathcal{A}_S^E$ is exactly the condition.

Only the allowed states will be visited but nothing followed after S: Mark S and all states on all paths to S as finite states. This as an A-automaton $\mathcal{A}_S^A$ is exactly the condition.

A similar negated condition might be simpler: Let S_+ be exactly the finite states. Interpret this as an E-automaton $\mathcal{A}_{S_+}^E$.

Then we have

$$\begin{aligned}
\tilde{L} &= \bigcup_{\text{SCC } S} L_E^\omega(\mathcal{A}_S^E) \cap L_A^\omega(\mathcal{A}_S^A) \\
&= \bigcup_{\text{SCC } S} L_E^\omega(\mathcal{A}_S^E) \cap -L_E^\omega(\mathcal{A}_{S_+}^E).
\end{aligned}$$

Thus,

$$\tilde{L} \in \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\text{reg}).$$

Given the *closure under change of final states*, we have $L^*(\mathcal{A}_S^E), L^*(\mathcal{A}_{S_+}^E) \in \mathcal{L}$, i.e.

$$\tilde{L} \in \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}.$$

Now let us show $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L} \subseteq \lim \mathcal{L}$.

With the *closure under suffix-independence*, we get $\mathrm{ext}\,\mathcal{L} \subseteq \lim \mathcal{L}$ and $\mathrm{ext}\,\mathcal{L} \subseteq \overline{\lim}\,\mathcal{L}$. I.e. $\mathrm{ext}\,\mathcal{L} \subseteq \lim \cap \overline{\lim}\,\mathcal{L}$. Let us show that $\lim \cap \overline{\lim}\,\mathcal{L}$ is closed under boolean closure.

Let $\tilde{L}_a, L_b \in \lim \cap \overline{\lim}\,\mathcal{L}$, i.e. $\exists L_{a1}, L_{a2}, L_{b1}, L_{b2} \in \mathcal{L}\colon \tilde{L}_a = \lim L_{a1} = \overline{\lim}\,L_{a2}, \tilde{L}_b = \lim L_{b1} = \overline{\lim}\,L_{b2}$. Let us show 1. $-\tilde{L}_a \in \lim \cap \overline{\lim}\,\mathcal{L}$, 2. $\tilde{L}_a \cup \tilde{L}_b \in \lim \cap \overline{\lim}\,\mathcal{L}$.

1. $-\tilde{L}_a = -\lim L_{a1} = \overline{\lim} -L_{a1}$, $-\tilde{L}_b = -\overline{\lim}\,L_{a2} = \lim -L_{a2}$. With the *closure under negation*, we get

   $$-\tilde{L}_a \in \lim \cap \overline{\lim}\,\mathcal{L}.$$

2. $\tilde{L}_a \cup \tilde{L}_b = \lim L_{a1} \cup \lim L_{b1} = \lim L_{a1} \cup L_{b1}$ (3.2). Thus, with *closure under union*, we have

   $$\tilde{L}_a \cup \tilde{L}_b \in \lim \mathcal{L}.$$

   The $\overline{\lim}\,\mathcal{L}$ case is harder. Let $\mathcal{A}_a$, $\mathcal{A}_b$ be deterministic automaton, so that $L^\omega_{\text{Büchi}}(\mathcal{A}_a) = L^\omega_{\text{co-Büchi}}(\mathcal{A}_a) = \tilde{L}_a$, $L^\omega_{\text{Büchi}}(\mathcal{A}_b) = L^\omega_{\text{co-Büchi}}(\mathcal{A}_b) = \tilde{L}_b$. Look at the product automaton $\mathcal{A}_a \times \mathcal{A}_b =: \overset{\times}{\mathcal{A}}$. Then we have $L^\omega_{\text{Büchi}}(\overset{\times}{\mathcal{A}}) = L^\omega_{\text{co-Büchi}}(\overset{\times}{\mathcal{A}}) = \tilde{L}_a \cup \tilde{L}_b$.

   Thus,

   $$\tilde{L}_a \cup \tilde{L}_b \in \overline{\lim}\,\mathcal{L}^*(\mathrm{reg}).$$

   Again, given the *closure under change of final states*, we have $L^*(\overset{\times}{\mathcal{A}}) \in \mathcal{L}$.

□

Lemma 3.16. *Let $\mathcal{L}$ be closed under change of final states. Then*

$$\mathrm{BC}\lim \mathcal{L} = \widehat{\mathrm{Kleene}}\,\mathcal{L}.$$

Proof. Let $U, V \in \mathcal{L}$. Look at the non-deterministic automaton $\mathcal{A}$ defined as:

$$\longrightarrow U \xrightarrow{\epsilon} V\odot$$

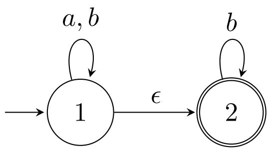
Then we have $L^{\omega}_{\text{Büchi}}(\mathcal{A}) = U \cdot V^{\omega}$.

Let us construct deterministic automata for $\mathcal{A}$ so that we can formulate 'V will be visited and not be left anymore' and 'finite states of the V-related automaton will be visited infinitely often' (or 'UV^* will be visited infinitely often').

In a constructed automaton, we must be able to tell wether we are in U or we deterministically have been in U the previous state. In a state power set construction, we can tell wether we are deterministically in U or not. If we are non-determinstic and we may be in both U or V and we get an input symbol which determines that we have been in U, we might not be able to tell from the following power set.

Example:

Let $U = (a+b)^*$, $V = \{b\}$. I.e. $UV^{\omega} = \{\alpha \in \{a,b\}^{\omega} \mid \text{at one point in } \alpha\text{, there are only } b\text{s}\}$. The non-deterministic automaton is:



Powerset construction: The initial state is $\{1,2\}$. Then we have:

- $\{1,2\} \xrightarrow{a} \{1,2\}$
- $\{1,2\} \xrightarrow{b} \{1,2\}$

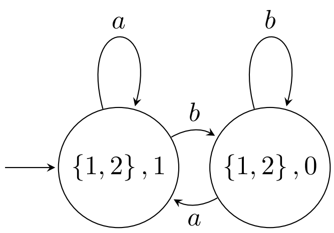
This gives the $*$-language $\{a,b\}^*$ and we cannot formulate UV^{ω} in any way from there.

In the construction, when we got the a from $\{1,2\}$, we knew that we have been deterministically in 1, i.e. in U. We loose this information. To keep it, we introduce another state flag which exactly says wether we have determined that we have been in U. Thus, we construct an automaton with the states $\mathcal{P}(Q) \times \mathbb{B}_{\text{det. been in } U}$, where Q are the states from $\mathcal{A}$.

For the example, we get the initial state $(\{1,2\}, 1)$. Then we have:

- $(\{1,2\}, 1) \xrightarrow{a} (\{1,2\}, 1)$
- $(\{1,2\}, 1) \xrightarrow{b} (\{1,2\}, 0)$
- $(\{1,2\}, 0) \xrightarrow{a} (\{1,2\}, 1)$
- $(\{1,2\}, 0) \xrightarrow{b} (\{1,2\}, 0)$

This is the automaton



When we mark all states from V and where we have not been deterministically in U as final, this as a co-Büchi automaton gives exactly the condition 'V will be visited and not be left anymore'. Let L_E be the $*$-language of this automata. Note that $L_E \neq UV^*$ in general and esp. in the example.

When we mark the final states as in the original non-deterministic automata, no matter about $\mathbb{B}_{\text{det. been in } U}$, with Büchi-acceptance, we get the condition 'UV^* will be visited infinitely often'. This is just $\lim UV^*$.

Together, we get UV^ω, i.e.:

$$\lim UV^* \cap \overline{\lim}\, L_E = UV^\omega$$

Given the *closure under change of final states*, we have $L_E \in \mathcal{L}$. Then, it follows

$$\left\{ \bigcup_{i=1}^{n} U_i \cdot V_i^\omega \;\middle|\; U_i, V_i \in \mathcal{L} \right\} = \widehat{\text{Kleene}}\, \mathcal{L} \subseteq \text{BC} \lim \mathcal{L}.$$

We also need to show $\mathrm{BC}\lim \mathcal{L} \subseteq \widehat{\mathrm{Kleene}}\,\mathcal{L}$.

Show: $\lim \mathcal{L} \subseteq \widehat{\mathrm{Kleene}}\,\mathcal{L}$.

Proof: Let $\mathcal{A}$ be a deterministic Büchi automaton for some language $\tilde{L} = L^{\omega}_{\text{Büchi}}(\mathcal{A}) \in \mathcal{L}$ with final states F.

For all finite states $q \in F$: If q is not part of a strongly connected component (SCC), we can ignore it. Let S be the SCC where $q \in S$. Then the set of all $\alpha \in \Sigma^{\omega}$ which are infinitely often in q can be described as $U_q \cdot V_q^{\omega}$, where U_q is the set of words so that we arrive in q and V_q is the set of words so that we get from q to q. Both sets are regular.

Thus,

$$\tilde{L} = L^{\omega}_{\text{Büchi}}(\mathcal{A}) = \bigcup_{q \in F} U_q V_q^{\omega}.$$

Obviously, the Kleene-Closure is closed under union.

TODO: Show that Kleene-Closure is closed under negation. (S306.5) (Follows with non-det Büchi complementation but a more generic proof might be useful.) $\square$

3.3 Congruence based language classes

3.3.1 Introduction

Definition 3.17. We define $\mathcal{L}(R)$ for an equivalence relation $R \subseteq \Sigma^* \times \Sigma^*$

$$\mathcal{L}^*(R) := \{L \subseteq \Sigma^* \mid L \text{ is finite union of } R\text{-equivalence-classes}\}.$$

Examples of such language classes are locally testable (LT, section 4.5), locally threshold testable (LTT, section 4.6) or piece-wise testable (PT, section 4.4) languages. At their definition, the word-relation basically tells wether a local test / piece-wise test can see a difference between two words.

If a language class $\mathcal{L}(R)$ is defined as finite union of equivalence classes of a relation $R \subseteq \Sigma^* \times \Sigma^*$ and

- the set of equivalent classes of R is finite,
- R is a congruence relation, i.e. also $(v, w) \in R \;\Leftrightarrow\; (va, wa) \in R \;\; \forall a \in \Sigma$

then we can construct a canonical deterministic automaton $\mathcal{A}_R$ which has $S_R := \Sigma^*/R$ as states, $\langle\epsilon\rangle_R$ is the initial state and the transitions are according to concatenation. Call this an R-automaton.

The LT, LTT and PT language classes have the above properties and thus such related canonical automaton.

The set of all such R-automata, varying in the final state set, is isomprophic to $\mathcal{L}(R)$. We have

$$\mathcal{L}^*(R) = \{L^*(\mathcal{A}_R(F)) \mid F \subseteq S_R\} =: \mathcal{L}^*(\mathcal{A}_R).$$

Obviously, by construction, such language classes are all *closed under change of final states*. Obviously, $\mathcal{L}^*(R)$ is also closed under *negation, union and intersection* (via negating, merging or intersecting the final state set of related automata). *Closure under suffix-independence* doesn't directly follow from this — we see some counter example later.

Definition 3.18. Analogously for ω, we get the set of R-E-automata with the ω-language-class

$$\mathcal{L}^\omega_E(\mathcal{A}_R) := \left\{L^\omega(\mathcal{A}^E_R(F)) \,\middle|\, F \subseteq S_R\right\},$$

R-Büchi-automata and

$$\mathcal{L}^\omega_{\text{Büchi}}(\mathcal{A}_R) := \left\{L^\omega(\mathcal{A}^{\text{Büchi}}_R(F)) \,\middle|\, F \subseteq S_R\right\},$$

R-Muller-automata and

$$\mathcal{L}^\omega_{\text{Muller}}(\mathcal{A}_R) := \left\{L^\omega(\mathcal{A}^{\text{Muller}}_R(\mathcal{F})) \,\middle|\, \mathcal{F} \subseteq 2^{S_R}\right\}.$$

Definition 3.19. For a relation R on Σ^*, there are various ways to construct a relation on Σ^ω. For now, we mainly study $R^\omega := \overline{\mathrm{ext}}\, R$, i.e.

$$(\alpha, \beta) \in R^\omega \;\;:\Leftrightarrow\;\; \forall n\colon (\alpha[0,n], \beta[0,n]) \in R.$$

Analogously to $\mathcal{L}(R)$, define the ω-language-class

$$\mathcal{L}^\omega(R^\omega) := \{L^\omega \subseteq \Sigma^\omega \mid L^\omega \text{ is finite union of } R^\omega\text{-equivalence-classes}\}.$$

3.3.2 Classification

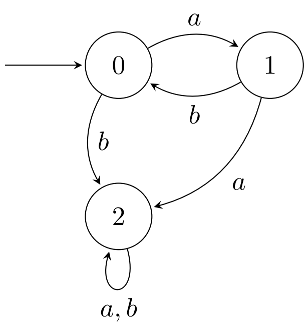
With this preparation, we get some obvious results:

Lemma 3.20. *$\mathcal{L}(R)$ is closed under negation, union, intersection and change of final states.*

Proof. This all follows directly from manipulations on final states of the canonical R-automata via negation, union, intersection or general change of final states. $\square$

Example 3.21. *There is an R such that $\mathcal{L}(R)$ is **not** closed under suffix-independence.*

Proof. Look at the transition graph over $\Sigma := \{a, b\}$:



Define the congruence relation $R \subseteq (\Sigma^*, \Sigma^*)$ as $(v, w) \in R \ :\Leftrightarrow\ v, w$ end up in the same state. Then, the above transition graph is exactly the R-automaton.

If 1 is the only final state, this is the language $L_1 = a(ba)^*$. With suffix independence, we get the language $L_1 \cdot \Sigma^* = a\Sigma^*$.

$a\Sigma^* \notin \mathcal{L}(R)$, thus $\mathcal{L}(R)$ is not *closed under suffix-independence*.

Furthermore, $\operatorname{ext} L_1 = a\Sigma^\omega$. Marking 0 or 1 as final state in a R-Büchi-automaton accepts the language $\tilde{L}_1 := \{(ab)^\omega\}$. Marking 2 as final state accepts the language $\tilde{L}_2 := (ab)^*(b|aa)\Sigma^\omega$. Then, $\lim \mathcal{L}(R) = \left\{\emptyset, \tilde{L}_1, \tilde{L}_2, \tilde{L}_1 \cup \tilde{L}_2\right\}$. And we have $\operatorname{ext} L_1 \notin \lim \mathcal{L}(R)$. But we also have $\tilde{L}_1 \notin \operatorname{ext} \mathcal{L}(R)$. $\square$

In the rest of this section, we show some equalities:

- $\mathcal{L}_E^\omega(\mathcal{A}_R) = \operatorname{ext} \mathcal{L}(R)$ (lemma 3.23)

- $\mathcal{L}^\omega_{\text{Büchi}}(\mathcal{A}_R) = \lim \mathcal{L}(R)$ (lemma 3.24)
- $\mathcal{L}^\omega_{\text{Muller}}(\mathcal{A}_R) = \mathrm{BC} \lim \mathcal{L}(R)$ (lemma 3.25)
- $\mathcal{L}^\omega(R^\omega) = \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}(R)$ (lemma 3.26)
- $\mathrm{BC} \lim \mathcal{L}(R) \cap \mathrm{ext}\,\mathcal{L}^*(\text{reg}) = \mathrm{ext}\,\mathcal{L}(R)$ (lemma 3.27)
- $\lim \mathcal{L}(R) \cap \overline{\lim}\,\mathcal{L}(R) = \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}(R)$ (lemma 3.39)

We will see that all those equations hold for all R, i.e. also for $\mathcal{L}(LT_n)$, $\mathcal{L}(LTT_n^k)$ and $\mathcal{L}(PT_n)$.

We will also see that $\mathrm{BC} \lim \mathcal{L}(R) \cap \lim \mathcal{L}^*(\text{reg}) = \lim \mathcal{L}(R)$ is not always the case. We will give an alternative characterization of this property (theorem 3.38).

Lemma 3.22.

$$\mathcal{L}^\omega_E(\mathcal{A}_R) = \mathrm{ext}\,\mathcal{L}(R)$$

Proof. Let $L = \bigcup_i \langle w_i \rangle_R$, $L \in \mathcal{L}(R)$. Then

$$\begin{aligned}
&L^\omega = \mathrm{ext}\,L\\
\Leftrightarrow\ &L^\omega = \left\{ \alpha \in \Sigma^\omega \,\middle|\, \exists n\colon \alpha[0,n] \in \bigcup_i \langle w_i \rangle_R \right\}\\
\Leftrightarrow\ &L^\omega = \{ \alpha \in \Sigma^\omega \mid \exists n\colon \delta_{\mathcal{A}_R}(\alpha[0,n]) \in \{\langle w_i \rangle_R \subseteq S_R \mid i\}\}\\
\Leftrightarrow\ &L^\omega = L^\omega(\mathcal{A}_R^E(\{\langle w_i \rangle_R \subseteq S_R \mid i\}))
\end{aligned}$$

□

Lemma 3.23.

$$\mathcal{L}^\omega_{\textit{Büchi}}(\mathcal{A}_R) = \lim \mathcal{L}(R)$$

Proof. Let $L = \bigcup_i \langle w_i \rangle_R$, $L \in \mathcal{L}(R)$. Then

$$\begin{aligned}
&L^\omega = \lim L\\
\Leftrightarrow\ &L^\omega = \left\{ \alpha \in \Sigma^\omega \,\middle|\, \exists^\infty n\colon \alpha[0,n] \in \bigcup_i \langle w_i \rangle_R \right\}\\
\Leftrightarrow\ &L^\omega = \{ \alpha \in \Sigma^\omega \mid \exists^\infty n\colon \delta_{\mathcal{A}_R}(\alpha[0,n]) \in \{\langle w_i \rangle_R \subseteq S_R \mid i\}\}\\
\Leftrightarrow\ &L^\omega = L^\omega(\mathcal{A}_R^{\text{Büchi}}(\{\langle w_i \rangle_R \subseteq S_R \mid i\}))
\end{aligned}$$

□

Lemma 3.24.

$$\mathcal{L}^{\omega}_{\mathit{Muller}}(\mathcal{A}_R) = \mathrm{BC}\lim\mathcal{L}(R)$$

Proof. Any $L^\omega \in \mathrm{BC}\lim\mathcal{L}(R)$ can be described by $\mathrm{BC}\,2^{S_R}$. $2^{2^{S_R}}$ is also finite. Thus, any $A \in \mathrm{BC}\,2^{S_R}$ can be represented in $2^{2^{S_R}}$. This is exactly an acceptance condition in Muller. □

Lemma 3.25.

$$\mathcal{L}^{\omega}(R^\omega) = \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}(R)$$

Proof. TODO... □

When we compare the outer $\mathrm{ext}\,\mathcal{L}^*(\mathrm{reg})$ (inside $\mathrm{BC}\lim\mathcal{L}^*(\mathrm{reg})$, i.e. all regular ω-languages) which is clearly a superset of $\mathrm{ext}\,\mathcal{L}$ and the inner whole class $\mathrm{BC}\lim\mathcal{L}$, a natural question is wether $\mathrm{BC}\lim\mathcal{L}\cap\mathrm{ext}\,\mathcal{L}^*(\mathrm{reg}) = \mathrm{ext}\,\mathcal{L}$. This is the case for $\mathcal{L}(R)$ as shown below.

Lemma 3.26.

$$\mathrm{BC}\lim\mathcal{L}(R)\cap\mathrm{ext}\,\mathcal{L}^*(\mathit{reg}) = \mathrm{ext}\,\mathcal{L}(R)$$

Proof. We have $\mathrm{ext}\,\mathcal{L}(R) \subseteq \mathrm{ext}\,\mathcal{L}^*(\mathrm{reg})$ and $\mathrm{ext}\,\mathcal{L}(R) \subseteq \mathrm{BC}\lim\mathcal{L}(R)$. Thus, "$\supseteq$" is shown.

Now, we show "$\subseteq$". Let $L^\omega \in \mathrm{BC}\lim\mathcal{L}(R)\cap\mathrm{ext}\,\mathcal{L}^*(\mathrm{reg})$. Because $L^\omega \in \mathrm{ext}\,\mathcal{L}^*(\mathrm{reg})$, there is an E-automaton $\mathcal{A}^E$ which accepts L^ω. We can assume that $\mathcal{A}^E$ is deterministic (with 3.2).

We must find an R-E-automaton which accepts L^ω. We will call it the $\overline{\mathcal{A}}^M$ E-automaton and will construct it in the following.

Let $\mathcal{A}^M$ be the deterministic R-Muller-automaton for L^ω (according to 3.3.1 and 3.25). Without restriction, there are no final state sets in $\mathcal{A}^M$ which are not loops. Then, $\overline{\mathcal{A}}^M$ has the same states and transitions as $\mathcal{A}^M$.

Look at a final state q^E of $\mathcal{A}^E$. Without restriction, we can assume that there is no path that we can reach multiple final states at once. Let L_{q^E} be all words which reach q^E exactly once at the end.

Let $w \in L_{q^E}$. Let q be the state in $\mathcal{A}^M$ which is reached after w. Let S be the set of states in $\mathcal{A}^M$ which can be reached from q.

Then, $\mathcal{A}^M$ accepts all words in $L_q \cdot L^\omega_{q,S}$, where L_q is the set of words to q and $L^\omega_{q,S}$ is the set of words of possible infinite postfixes after q in S so that they are accepted. Any word with a prefix in L_q, which is not in $L_q \cdot L^\omega_{q,S}$, will not be accepted by $\mathcal{A}^M$ because $\mathcal{A}^M$ is deterministic. Also, because $L_{q^E} \cap L_q \neq \emptyset$ and $L_{q^E} \cdot \Sigma^\omega \subseteq L^\omega$ and $L_q \cdot L^\omega_{q,S} \subseteq L^\omega$, we get $L^\omega_{q,S} \neq \emptyset$.

Assuming $L^\omega_{q,S} \neq \Sigma^\omega$. Then we would have $L^\omega \notin \operatorname{ext} \mathcal{L}^*(\text{reg})$, which is a contradiction. I.e. $L^\omega_{q,S} = \Sigma^\omega$.

Thus, $\mathcal{A}^M$ accepts all words in $L_q \cdot \Sigma^\omega$. Mark q as a final state in $\overline{\mathcal{A}}^M$. Thus, $\overline{\mathcal{A}}^M$ E-accepts all words in $L_q \cdot \Sigma^\omega \subseteq L^\omega$.

Because we did this for all final states in $\mathcal{A}^E$, there is no $\alpha \in L^\omega$ which is not accepted by $\overline{\mathcal{A}}^M$. I.e., the R-E-automata $\overline{\mathcal{A}}^M$ accepts exactly L^ω. I.e. $L^\omega \in \operatorname{ext} \mathcal{L}(R)$. □

In fact, we actually have shown $\mathrm{BC} \lim \mathcal{L} \cap \operatorname{ext} \mathcal{L}^*(\text{reg}) = \operatorname{ext} \mathcal{L}$ for any $\mathcal{L} \subseteq \mathcal{L}^*(\text{reg})$.

We are also interested in the equality $\mathrm{BC} \lim \mathcal{L} \cap \lim \mathcal{L}^*(\text{reg}) = \lim \mathcal{L}$ for some $*$-language class $\mathcal{L} \subseteq \mathcal{L}^*(\text{reg})$. This is a connection between the outer $\lim \mathcal{L}^*(\text{reg})$ (inside $\mathrm{BC} \lim \mathcal{L}^*(\text{reg})$) which is clearly a superset of $\lim \mathcal{L}$ and the inner whole class $\mathrm{BC} \lim \mathcal{L}$. It turns out that this is not always the case and not as straightforward as in the ext case in lemma 3.27.

We have $\lim \mathcal{L} \subseteq \lim \mathcal{L}^*(\text{reg})$ and $\lim \mathcal{L} \subseteq \mathrm{BC} \lim \mathcal{L}$. Thus, "$\supseteq$" does hold in all cases.

Example 3.27. *The equality does not hold for any $\mathcal{L}$.*

Proof. Let $\Sigma = \{a, b\}$,

$$L_a := (b^* a^+)(b^+ a^+)^*, \quad L_b := b^*(a^+ b^+)^* \quad \text{and} \quad \mathcal{L} := \{L_a, L_b\}.$$

Then, $\lim L_a$ is the set of words where a occurs infintely often and $\lim L_b$ is the set of words where b occurs infinitely often. Then, $L_\omega := \lim L_a \cap \lim L_b \in \mathrm{BC} \lim \mathcal{L}$. Also, let

$$L_{ab} := (a^* b^+ a)^*.$$

Then, $\lim L_{ab}$ is the set of words where both a and b occurs infintely often. Thus, $\lim L_{ab} = L_\omega$. Obviously, we also have $L_{ab} \in \mathcal{L}^*(\text{reg})$. Thus, $L_\omega \in \mathrm{BC} \lim \mathcal{L} \cap \lim \mathcal{L}^*(\text{reg})$. But we can also see that $L_\omega \notin \lim \mathcal{L}$. □

Thus, we need some conditions on $\mathcal{L}$ for the equality. Here, we will study the class $\mathcal{L}(R)$. We will introduce a property on $\mathcal{L}(R)$ where we can show the equality. The idea of this property is comming from the study of this equality in terms of automata. Let $L_\omega \in \mathrm{BC}\lim\mathcal{L}(R)$. Then there is a representing R-Muller-automaton $\mathcal{A}_M$ for L_ω. Let also $L_\omega \in \lim\mathcal{L}^*(\mathrm{reg})$. Then there is representing deterministic Büchi automaton $\mathcal{A}_r$ for L_ω. We want to show that $L_\omega \in \lim\mathcal{L}(R)$. I.e. we are searching for a representing deterministic automaton whose language is in $\mathcal{L}(R)$ and where the Büchi-acceptance gives us L_ω. Because the R-Büchi-automaton is the canonical deterministic Büchi automaton for $\mathcal{L}(R)$, we must be able to construct such R-Büchi-automaton $\mathcal{A}_B$ for L_ω. Let us look at the product automaton $\mathcal{A}_M \times \mathcal{A}_r$ and determine from there the final state set of $\mathcal{A}_B$. $\mathcal{A}_M$ already has the right transition graph. $\mathcal{A}_r$ has the Büchi acceptance. So, when looking at the product automaton, we try to find the loops in $\mathcal{A}_M$ which match a final state in $\mathcal{A}_r$. $\mathcal{A}_r$ might be bigger than $\mathcal{A}_M$ and it doesn't seem clear wether the Muller final state sets of $\mathcal{A}_M$ can be translated to a Büchi final state set. However, when we say that each SCC in $\mathcal{A}_M$ has exactly one loop, there is no inconclusiveness about wether there is a Büchi final state in this SCC in $\mathcal{A}_B$ or not. We will formulate this formally below. So, if we have that property on $\mathcal{A}_M$, i.e. on $\mathcal{L}(R)$, we can construct $\mathcal{A}_B$ and thus we have the equality.

Definition 3.28. For $\alpha \in \Sigma^\omega$, let $\overrightarrow{\alpha} \in (\Sigma^*/R)^\omega$ be the state sequence we run through with α and thus $\mathrm{Inf}(\overrightarrow{\alpha}) \subseteq \Sigma^*/R$ those states which are visited infinitely often.

If for every $L \in \mathcal{L}(R)$, there is an inclusion function $B_L \colon \Sigma^*/R \to \mathbb{B}$ such that for every $\alpha \in \Sigma^\omega$, we have

$$\alpha \in L \quad\Leftrightarrow\quad \exists q \in \mathrm{Inf}(\overrightarrow{\alpha})\colon B_L(q) = 1.$$

Also, for every $s \notin \mathrm{Inf}(\overrightarrow{\alpha})$,

$$B_L(s) = B_L(q) \quad \forall q \notin \mathrm{Inf}(\overrightarrow{\alpha})$$

and

$$B_L(s) \neq B_L(q) \quad \forall q \in \mathrm{Inf}(\overrightarrow{\alpha}).$$

If such B_L always exists, we call $\mathcal{L}$ **infinity-postfix-independent**.

This definition is as general as possible. It is also well defined if there are an infinity number of equivalence classes of R. Thus, $\mathcal{L}(R)$ doesn't need to be regular. Also, in that case, there might be an $\alpha \in \Sigma^\omega$ with $\mathrm{Inf}(\overrightarrow{\alpha}) = \emptyset$.

If $\mathcal{L}(R)$ is regular and we look at an R-ω-automaton, it basically says that when we visit some state infinitely often, it determines in what loop we are and we cannot switch the loop. Formally:

Lemma 3.29. *Let R be a congruence relation with a finite number of equivalence classes. $\mathcal{L}(R)$ is* infinity-postfix-independent *exactly if and only if every SCC Q in the R-automata (as defined in section 3.3.1) has exactly one looping subset, i.e. Q itself is the only loop in Q.*

Proof. $S_R := \Sigma^*/R$ are the states of the R-automata. Let $Q \subseteq S_R$ be any SCC. Let $\tilde{L} \in \mathrm{BC}\lim\mathcal{L}(R)$. Let $\mathcal{A}_M$ be the R-Muller-automaton accepting $\tilde{L}$.

'$\Rightarrow$': Let $\mathcal{L}(R)$ be *infinity-postfix-independent*. Then, for L, we have an inclusion function $B_L\colon S_R \to \mathbb{B}$. If there is an accepting loop $Q' \subseteq Q$ in $\mathcal{A}_M$, it means that every $\alpha \in \tilde{L}$ which ends up in Q' is accepted, thus there is a $q' \in Q'$ with $B_L(q') = 1$. Because we can loop through all of Q and thus construct $\beta \in \Sigma^\omega$ with $\mathrm{Inf}(\overrightarrow{\beta}) = Q$, we get $B_L(q) = 1$ for all $q \in Q$. Thus, all possible loops in Q will accept. This was general for any SCC and any language $\tilde{L} \in \mathrm{BC}\lim\mathcal{L}(R)$. Because this is a Muller-automaton, this can only be if Q itself is the only loop. Otherwise we can have both an accepting loop and a non-accepting loop in Q which is a contradiction.

'$\Leftarrow$': The SCC Q has exactly one looping subset. This is Q itself. Assuming Q is accepting in $\mathcal{A}_M$. Then define $B_L(q) = 1$ for all $q \in Q$, otherwise $B_L(q) = 0$. This $B_L\colon S_R \to \mathbb{B}$ has the needed properties, thus $\mathcal{L}(R)$ is *infinity-postfix-independent*. □

For piecewise testable languages, this is the case.

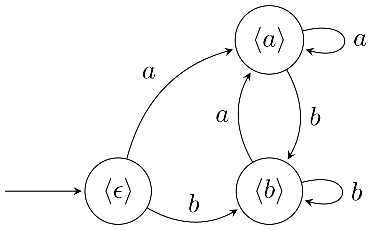
For locally testable languages, this is *not* the case. Depending on the ending of $\alpha[0, n]$, we can switch through different equivalence classes and visit different loops.

Example 3.30. *There is an R so that $\mathcal{L}(R)$ is not* infinity-postfix-independent *and*

$$\mathrm{BC}\lim\mathcal{L}(R) \cap \lim\mathcal{L}^*(\mathit{reg}) \neq \lim\mathcal{L}(R).$$

Proof. If we take the example 3.28: For $v, w \in \{a, b\}^*$, let $v =_R w \;:\Leftrightarrow\; v, w$ end up with the same symbol. I.e., the equivalence classes are $\langle\epsilon\rangle, \langle a\rangle, \langle b\rangle$.

The R-automata is



From example 3.28, we have $L_a = \langle a \rangle$ and $L_b = \langle \epsilon \rangle \cup \langle b \rangle$ and $\mathcal{L} = \{L \in \mathcal{L}(R) \mid \#L = \infty\}$ (only the infinity $L \in \mathcal{L}(R)$ matter for $\lim$). We also see from the R-automata that there is no way to mark states as final states for Büchi-acceptance so that we get the condition "both a and b occur infinitely often". Via Muller, we just mark the loop $\{\langle a \rangle, \langle b \rangle\}$ as final. I.e. $\lim L_{ab} \notin \lim \mathcal{L}(R)$ but $\lim L_{ab} \in \mathrm{BC} \lim \mathcal{L}(R)$ and as shown in example 3.28, $\lim L_{ab} \in \lim \mathcal{L}^*(\mathrm{reg})$. Thus, $\mathrm{BC} \lim \mathcal{L}(R) \cap \lim \mathcal{L}^*(\mathrm{reg}) \neq \lim \mathcal{L}(R)$. □

This example can actually be extended to be the $\mathcal{L}(LT_2)$ class and generalized to any $\mathcal{L}(LT_n)$. I.e. for all $n \in \mathbb{N}$, $\mathcal{L}(LT_n)$ is not *infinty-postfix-independent* and

$$\mathrm{BC} \lim \mathcal{L}(LT_n) \cap \lim \mathcal{L}^*(\mathrm{reg}) \neq \lim \mathcal{L}(LT_n).$$

When studying the *infinity-postfix-independence* property in more detail, we get the surprising result:

Lemma 3.31. *Let R be a congruence relation with a finite number of equivalence classes. Let $\mathcal{L}(R)$ be* infinity-postfix-independent. *Then*

$$\mathrm{BC} \lim \mathcal{L}(R) = \lim \mathcal{L}(R).$$

Proof. $S_R := \Sigma^*/R$ are the states of the R-automata. Let $Q \subseteq S_R$ be any SCC. Let $\tilde{L} \in \mathrm{BC} \lim \mathcal{L}(R)$. Let $\mathcal{A}_M$ be the R-Muller-automaton accepting $\tilde{L}$.

For L, we have an inclusion function $B_L \colon S_R \to \mathbb{B}$. If there is an accepting loop $Q' \subseteq Q$ in $\mathcal{A}_M$, it means that every $\alpha \in \tilde{L}$ which ends up in Q' is accepted, thus there is a $q' \in Q'$ with $B_L(q') = 1$. Because we can loop through all of Q and thus construct $\beta \in \Sigma^\omega$ with $\mathrm{Inf}(\overrightarrow{\beta}) = Q$, we get $B_L(q) = 1$ for all $q \in Q$. Thus, all possible loops in Q will accept. In an R-Büchi-automata $\mathcal{A}_B$, we can mark all states of Q as final states. This was for any SCC Q, thus $\mathcal{A}_B$ will accept exactly iff $\mathcal{A}_M$ accepts. Thus we have $\tilde{L} \in \lim \mathcal{L}(R)$. This was for any $\tilde{L}$, i.e. $\mathrm{BC} \lim \mathcal{L}(R) = \lim \mathcal{L}(R)$. □

Lemma 3.32. *Let R be a congruence relation with a finite number of equivalence classes. Let $\mathcal{L}(R)$ be* infinity-postfix-independent. *Then*

$$\mathrm{BC} \lim \mathcal{L}(R) \cap \lim \mathcal{L}^*(\mathit{reg}) = \lim \mathcal{L}(R).$$

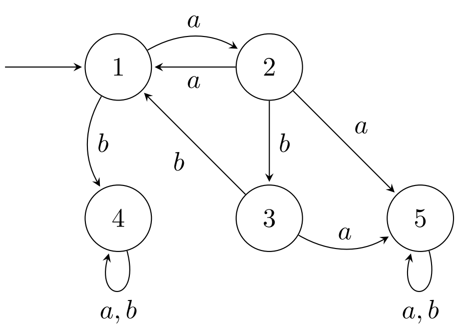
Proof. In lemma 3.32, we showed that we have $\mathrm{BC} \lim \mathcal{L}(R) = \lim \mathcal{L}(R)$. Because $\lim \mathcal{L}(R) \subseteq \lim \mathcal{L}^*(\mathrm{reg})$, we directly get the claimed equality. □

The question arises wether *infinity-postfix-independence* on $\mathcal{L}(R)$ is equivalent to the equality $\mathrm{BC} \lim \mathcal{L}(R) \cap \lim \mathcal{L}^*(\mathrm{reg}) = \lim \mathcal{L}(R)$. We have shown in lemma 3.30 that if $\mathcal{L}(R)$ is not *infinity-postfix-independent*, there must be a SCC $Q \subseteq S_R$ with more than one loop.

Example 3.33. *There is a congruence relation R with a finite number of equivalence classes where $\mathcal{L}(R)$ is not* infinity-postfix-independent *but we still have*

$$\mathrm{BC} \lim \mathcal{L}(R) \cap \lim \mathcal{L}^*(\mathit{reg}) = \lim \mathcal{L}(R).$$

Proof. Look at the fully connected transition graph over $\Sigma := \{a, b\}$:



Define the congruence relation $R \subseteq (\Sigma^*, \Sigma^*)$ via the transition graph: $(v, w) \in R \;:\Leftrightarrow\; v, w$ end up in the same state. Then, the R-automata is exactly the transition graph. Look at the SCC $Q := \{1, 2, 3\}$. Q has two loops $P_1 := \{1, 2\}$ and $P_2 := Q$, where $P_1 \subsetneq P_2$. Thus, $\mathcal{L}(R)$ is not *infinity-postfix-independent*. Let $\tilde{L}$ be the language accepted by the R-Muller-automaton which only accepts the loop P_1. Then, $\tilde{L}$ is not recognizable by a deterministic Büchi automaton. Thus, we also have $\mathrm{BC} \lim \mathcal{L}(R) \neq \lim \mathcal{L}(R)$.

The R-Muller-automaton only accepting P_2 is equivalent to the R-Büchi-automaton only accepting state 3. The R-Muller-automaton accepting both P_1 and P_2 is equivalent to the R-Büchi-automaton accepting Q.

All other R-Büchi-automata can be constructed canonically from that. Thus we have

$$\mathrm{BC}\lim\mathcal{L}(R)\cap\lim\mathcal{L}^*(\mathrm{reg})=\lim\mathcal{L}(R).$$

□

However, if we get stricter on the possible subloops, we can show the inequality.

Definition 3.34. Let R be a congruence relation with a finite number of equivalence classes.

If there is a SCC $Q\subseteq S_R$ including two loops $P_1,P_2\subseteq Q$, $P_1\neq P_2$ with $P_1\not\subseteq P_2$, $P_2\not\subseteq P_1$, then call $\mathcal{L}(R)$ **postfix-loop-deterministic**.

Lemma 3.35. *Let R be a congruence relation with a finite number of equivalence classes. And let $\mathcal{L}(R)$ be* postfix-loop-deterministic. *Then*

$$\mathrm{BC}\lim\mathcal{L}(R)\cap\lim\mathcal{L}^*(\mathit{reg})\neq\lim\mathcal{L}(R).$$

Proof. There is a SCC $Q\subseteq S_R$ including two loops $P_1,P_2\subseteq Q$, $P_1\neq P_2$ with $P_1\not\subseteq P_2$ and $P_2\not\subseteq P_1$. They are in the same SCC Q, thus there is an outer loop $P\subseteq Q$ with $P_1,P_2\subseteq P$. In the R-Muller-automaton, let P be the only final state set. Let $\tilde{L}$ be the language accepted by this. For every $q\in Q$, look at the R-Büchi-automaton where q is the only final state. The intersection of all these is recognized by a deterministic Büchi automaton (lemma 2.1). And the intersection accepts exactly $\tilde{L}$. Thus, $\tilde{L}\in\mathrm{BC}\lim\mathcal{L}(R)\cap\lim\mathcal{L}^*(\mathrm{reg})$. However, there is no way in the R-Büchi-automaton to mark a subset of Q as the final states such that we accept $\tilde{L}$. Thus, $\tilde{L}\notin\lim\mathcal{L}(R)$. □

Now, the question arises wether *postfix-loop-determinism* is equivalent to the inequality.

Lemma 3.36. *Let $\mathcal{L}(R)$ be not* postfix-loop-deterministic. *Then*

$$\mathrm{BC}\lim\mathcal{L}(R)\cap\lim\mathcal{L}^*(\mathit{reg})=\lim\mathcal{L}(R).$$

Proof. For all SCC Q and subloops $P_1, P_2 \subseteq Q$, we either have $P_1 = P_2$ or $P_1 \subseteq P_2$ or $P_2 \subseteq P_1$. If we have always $P_1 = P_2$ for this Q, it means that Q has only one loop. Then, if an R-Muller-automaton accepts Q, we can just mark any state $q \in Q$ final in a R-Büchi-automaton and every α going through Q would be accepted by the R-Büchi-automata exactly if it would be accepted by the R-Muller-automata.

Now, assume that there is $P_1 \subseteq P_2$. If R-Muller would accept P_1 but not P_2, the resulting language would not be recognizable by deterministic Büchi automata, thus we would be out of $\lim \mathcal{L}^*(\text{reg})$. If R-Muller accepts P_2 but not P_1, we mark some state from $P_2 - P_1$ as final in the R-Büchi automaton. If it accepts both, we mark some state from P_1 as fina in R-Büchi. In either case, we are in $\lim \mathcal{L}(R)$. If there are other loops $P' \subseteq Q$, they are either supersets of P_2 or subsets of P_1 and thus we can use the same argumentation.

We showed that for all SCC of the R-automata. Thus we have shown the claimed equality. □

Thus, we get the final result

Theorem 3.37. *$\mathcal{L}(R)$ is* postfix-loop-deterministic *exactly if and only if*

$$\mathrm{BC} \lim \mathcal{L}(R) \cap \lim \mathcal{L}^*(\textit{reg}) = \lim \mathcal{L}(R).$$

Proof. Lemma 3.36 and lemma 3.37. □

Lemma 3.38.

$$\lim \mathcal{L}(R) \cap \overline{\lim} \mathcal{L}(R) = \mathrm{BC}\, \mathrm{ext}\, \mathcal{L}(R)$$

Proof. TODO... □

Chapter 4

Results on concrete $*$-language classes

We already showed many results for $\mathcal{L}^*(\mathrm{reg})$.

4.1 Starfree regular languages

This class is also equivalent to the set of $\mathrm{FO}[<]$-definable languages.

Theorem 4.1.

$$\mathcal{L}^\omega(\mathrm{FO}[<]) = \mathrm{BC}\lim \mathcal{L}^*(\mathrm{FO}[<])$$

Proof. Let $\varphi \in \mathrm{FO}[<]$. By the [Tho81, Normal Form Theorem (4.4)] there are bounded formulas $\varphi_1(y), \cdots, \varphi_r(y), \psi_1(y), \cdots, \psi_r(y)$ such that for all $\alpha \in \Sigma^\omega$:

$$\alpha \models \varphi \Leftrightarrow \alpha \models \bigvee_{i=1}^{r} (\forall x \exists y > x \colon \varphi_i(y)) \wedge \neg (\forall x \exists y > x \colon \psi_i(y))$$

Thus:

$$\begin{aligned}\alpha \models \varphi \Leftrightarrow \bigvee_{i=1}^{r} &\underbrace{(\alpha \models \forall x \exists y > x \colon \varphi_i(y))} \wedge \neg (\alpha \models \forall x \exists y > x \colon \psi_i(y)) \\ &\Leftrightarrow \forall x \exists y > x \colon \alpha[0,n] \models \varphi_i(\omega) \\ &\Leftrightarrow \exists^\omega n \colon \alpha[0,n] \models \varphi_i(\omega) \\ &\Leftrightarrow \alpha \in \lim L^*(\varphi_i(\omega))\end{aligned}$$

where $\varphi_i(\omega)$ stands for φ_i with all bounds removed. I.e. we have

$$L^\omega(\varphi) = \bigcup_{i=1}^{r} \lim(L^*(\varphi_i(\omega)) \cap \neg \lim(L^*(\psi_i(\omega))),$$

and thus

$$L^\omega(\varphi) \in \mathrm{BC}\lim \mathcal{L}^*(\mathrm{FO}[<]).$$

We have prooved the $\subseteq$-direction. For $\supseteq$:

$$
\begin{aligned}
&\alpha \in \lim(L^*(\varphi))\\
\Leftrightarrow\ &\exists^\omega n\colon \alpha[0,n] \models \varphi\\
\Leftrightarrow\ &\alpha \models \forall x \exists y > x\colon \varphi(y)\\
\Leftrightarrow\ &\alpha \in L^\omega(\forall \exists y > x\colon \varphi(y))
\end{aligned}
$$

where $\varphi(y)$ stands for φ with all variables bounded by y. I.e.

$$\lim \mathcal{L}^*(\mathrm{FO}[<]) \subseteq \mathcal{L}^\omega(\mathrm{FO}[<]),$$

and thus also

$$\mathrm{BC} \lim \mathcal{L}^*(\mathrm{FO}[<]) \subseteq \mathcal{L}^\omega(\mathrm{FO}[<]).$$

Thus we have prooved the equality. □

Theorem 4.2.

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{FO}[<]) \subsetneq \mathrm{BC}\lim \mathcal{L}^*(\mathrm{FO}[<])$$

Proof. $\subseteq$: $L \subset \Sigma^\omega$ starfree $\;\Rightarrow\;$ $L\Sigma^\omega \in \lim(\mathcal{L}^*(\mathrm{FO}[<]))$ □

Proof. $\neq$:

$$
\begin{aligned}
&L := (\Sigma^* a)^\omega\\
\Rightarrow\ &L = \lim((\Sigma^* a)^*)\\
\Rightarrow\ &L = L^\omega(\exists^\omega x : Q_a x)
\end{aligned}
$$

And we have $L \notin \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{FO}[<])$. □

With 3.3, we get $\mathrm{ext}\,\mathcal{L} \subseteq \lim \mathcal{L}$.

$\tilde{L} := \lim(\Sigma^* a) = (\Sigma^* a)^\omega \in \lim \mathcal{L}$ but $\tilde{L} \notin \mathrm{ext}\,\mathcal{L}$ as shown in chapter 2.2.

- P1: $\{a\} \in \mathcal{L}$. $a\Sigma^* \in \mathcal{L}$, thus $a\Sigma^\omega = \mathrm{ext}(\{a\}) = \overline{\mathrm{ext}}\, a\Sigma^*$.
- P2a: $\tilde{L}_{2a} := \mathrm{ext}(a^*b) = a^*b\Sigma^\omega$, $a^*b \in \mathcal{L}$. Then $\tilde{L}_{2a} \notin \mathrm{ext}\,\mathcal{L}^*(\mathrm{reg}) \supseteq \mathcal{L}^*(\mathrm{FO}[<])$.

- P2b: $-\tilde{L}_{2a} := \overline{\mathrm{ext}}(-a^*b)$, $-a^*b \in \mathcal{L}$. Then $-\tilde{L}_{2a} \notin \mathrm{ext}\,\mathcal{L}$.
- P3: Follows directly from P2a and P2b.
- P4: $\tilde{L}_4 := \mathrm{ext}(\Sigma^* a) \cap \overline{\mathrm{ext}}(-\Sigma^* b) = \Sigma^* a \Sigma^\omega \cap -(\Sigma^* b \Sigma^\omega)$, whereby $\Sigma^* a \in \mathcal{L}$, $-\Sigma^* b \in \mathcal{L}$. $\tilde{L}_4 \notin \mathrm{ext} \cup \overline{\mathrm{ext}}\,\mathcal{L}^*(\text{reg}) \supseteq \mathcal{L}^*(\mathrm{FO}[<])$ but $\tilde{L}_4 \in \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}$.
- P5: TODO
- P6a/P6b/P7/P8: $\Sigma^* a \in \mathcal{L}$. We can use the same arguments as for $\mathcal{L}^*(\text{reg})$.
- P9: TODO
- P10: TODO

4.2 FO[+1]

Theorem 4.3.

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

Proof. From [Tho96, Theorem 4.8], we know that each formular in FO[+1] is equivalent (for both finite and infinite words) to a boolean combination of statements "sphere $\sigma \in \Sigma^+$ occurs $\geq n$ times". That statement can be expressed by a sentence of the form

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

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

$$\bigwedge_{\substack{i,j \in \underline{n}, \\ i \neq j, \\ k,l \in \underline{|\sigma|}}} x_{i,k} \neq x_{j,l} \ \wedge \bigwedge_{\substack{i \in \underline{n}, \\ k \in \underline{|\sigma|-1}}} x_{i,k+1} = x_{i,k} + 1 \ \wedge \bigwedge_{\substack{i \in \underline{n}, \\ k \in \underline{|\sigma|}}} Q_{\sigma_k} x_{i,k}$$

For ψ, we have:

$$\alpha \models \psi \Leftrightarrow \exists n \colon \alpha[0,n] \models \psi \ \text{ for all } \alpha \in \Sigma^\omega,$$

i.e.

$$L^\omega(\psi) = \mathrm{ext}\, L^*(\psi).$$

Any formular in FO[+1] can be expressed as a boolean combination of ψ-like formular. With

$$L^\omega(\neg\psi) = \neg L^\omega(\psi)$$
$$L^\omega(\psi_1 \wedge \psi_2) = L^\omega(\psi_1) \cap L^\omega(\psi_2)$$
$$L^\omega(\psi_1 \vee \psi_2) = L^\omega(\psi_1) \cup L^\omega(\psi_2)$$

we get:

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

□

4.3 FO[]

4.4 Piece-wise testable languages

Theorem 4.4.

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{PT}) = \mathrm{BC}\lim\mathcal{L}^*(\mathrm{PT})$$

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 $\mathrm{ext}(\mathcal{L}^*(\mathrm{PT})) \subseteq \mathrm{BC}\lim\mathcal{L}^*(\mathrm{PT})$.

By complete induction:

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

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

$$\mathrm{ext}(A \cup B) = \mathrm{ext}(A) \cup \mathrm{ext}(B)$$
$$\mathrm{ext}(A \cap B) = \mathrm{ext}(A) \cap \mathrm{ext}(B)$$

This makes the induction complete.

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

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

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

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

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

and because A, B are piece-wise testable

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

□

For positive piece-wise testable (pos-PT) languages, we get the same result.

Theorem 4.5.

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\text{pos-PT}) = \mathrm{BC}\,\lim\,\mathcal{L}^*(\text{pos-PT})$$

Proof. $\subseteq$: Exactly like the proof for PT except that we leave out the negated part. $\supseteq$: Also like the proof for PT. □

We also have a relation between pos-PT and PT.

Theorem 4.6.

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\text{pos-PT}) = \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{PT})$$

Proof. In the proof of $\lim\mathcal{L}^*(\mathrm{PT}) \subseteq \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{PT})$ we actually proved $\mathrm{BC}\lim\mathcal{L}^*(\mathrm{PT}) \subseteq \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\text{pos-PT})$. Similiarly we also proved $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{PT}) \subseteq \mathrm{BC}\lim\mathcal{L}^*(\text{pos-PT})$.

With 4.4 and 4.4 we get the claimed equality. □

4.5 Locally testable languages

Theorem 4.7.

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{LT}) \subsetneq \mathrm{BC}\lim\mathcal{L}^*(\mathrm{LT})$$

Proof. Let $w \in \Sigma^+$.

$$\mathrm{ext}(w\Sigma^*) = \lim(w\Sigma^*)$$
$$\mathrm{ext}(\Sigma^* w) = \Sigma^* w \Sigma^\omega = \lim(\Sigma^* w \Sigma^*)$$
$$\mathrm{ext}(\Sigma^* w \Sigma^*) = \Sigma^* w \Sigma^\omega = \lim(\Sigma^* w \Sigma^*)$$

Thus we have

$$\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{LT}) \subseteq \mathrm{BC}\lim\mathcal{L}^*(\mathrm{LT}).$$

But we also have

$$\lim(\Sigma^*) = (\Sigma^* w)^\omega \notin \mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(\mathrm{LT}).$$

$\square$

4.6 Locally threshold testable languages

4.7 Endwise testable languages

- $\mathrm{BC}\,\mathrm{ext}\,\mathcal{L}^*(endwise) \neq \mathrm{BC}\lim\mathcal{L}^*(endwise)$ because $\Sigma^* a \in \mathcal{L}^*(endwise)$.
- $\mathrm{ext}(a\Sigma^* a) = a\Sigma^* a\Sigma^\omega \notin \mathrm{BC}\lim\mathcal{L}^*(endwise)$

4.8 Local languages

4.9 Finite / Co-finite languages

- $\lim\mathcal{L}^*(finite) = \{\emptyset\}$
- $\mathrm{ext}\,\mathcal{L}^*(finite) = \mathcal{L}^*(finite) \cdot \Sigma^\omega$

- $\lim \mathcal{L}^*(co - finite) = \{\Sigma^\omega\}$
- $\operatorname{ext} \mathcal{L}^*(co - finite) = \{\Sigma^\omega\}$

4.10 Regular languages of dot-depth-n

4.11 L-trivial languages

4.12 R-trivial languages

4.13 Locally modulo testable languages

4.14 Context free languages

Chapter 5

Conclusion

Chapter 6

References

[AH04] Rajeev Alur and Thomas A. Henzinger. Computer-aided verification. `http://mtc.epfl.ch/courses/CAV_WS2004/`, 2004.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik und Grundl. Math.*, 6:66–92, 1960.

[Büc62] J. Richard Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr .)*, pages 1–11. Stanford Univ. Press, Stanford, Calif., 1962.

[Kle56] S. C. Kleene. Representation of events in nerve nets and finite automata. In *Automata studies*, pages 3–41. Princeton University Press, Princeton, N. J., 1956. Annals of mathematics studies, no. 34.

[Mul63] David E. Muller. Infinite sequences and finite machines. In *Switching Theory and Logical Design, Proc. Fourth Annual Symp. IEEE*, pages 3–16. IEEE, 1963.

[PP04] D. Perrin and J.É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Number Bd. 141 in Pure and Applied Mathematics. Elsevier, 2004.

[Sta97] Ludwig Staiger. ω-languages. In *Handbook of formal languages, Vol. 3*, pages 339–387. Springer, Berlin, 1997.

[Str94] Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Birkhäuser Boston Inc., Boston, MA, 1994.

[Tho81] Wolfgang Thomas. A combinatorial approach to the theory of omega-automata. *Information and Control*, 48(3):261–283, 1981.

[Tho90] Wolfgang Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume vol. B, Formal models and semantics, pages 135–191. Elsevier, 1990.

[Tho96] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

[Tho10] Wolfgang Thomas. Infinite computations. Technical report, RWTH Aachen, February 2010.