I F I G

RESEARCH

REPORT

INSTITUT FÜR INFORMATIK

HIERARCHY OF SUBREGULAR LANGUAGE FAMILIES

Bianca Truthe

IFIG RESEARCH REPORT 1801

FEBRUARY 2018

Institut für Informatik
JLU Gießen
Arndtstraße 2
35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

IFIG Research Report

IFIG Research Report 1801, February 2018

Hierarchy of Subregular Language Families

Bianca Truthe[1]

Institut für Informatik, Universität Giessen
Arndtstraße 2, 35392 Giessen, Germany

Abstract. In the area of formal languages and automata theory, regular languages and finite automata are widely studied. Several classes of specific finite automata and their accepted languages have been investigated, for example, definite automata and non-counting automata. Subfamilies of the family of the regular languages can also be motivated by their specific representations as regular expressions, for example, the family of the union-free languages or the family of the star-free languages. Another line of research is to consider subfamilies of the family of the regular languages which are based on ressources needed for generating or accepting them (like the number of non-terminal symbols, production rules, or states).
In this paper, we prove inclusion relations and incomparabilities of subregular language families which are based on structural properties (like the set of all suffix-closed or commutative regular languages) or on descriptional complexity measures.

MSC Classification: 68Q45 Formal languages and automata

Additional Key Words and Phrases: Subregular Language Families, Inclusion Relations, Hierarchy

[1] E-mail: bianca.truthe@informatik.uni-giessen.de

1 Introduction

In the area of formal languages and automata theory, regular languages and finite automata are widely studied. Several classes of specific finite automata and their accepted languages have been investigated separately, for example, definite automata by M. Peres, M. O. Rabin, and E. Shamir in [23], non-counting automata by R. McNaughton and S. Papert in [21], and communicating automata by R. Laing and J. B. Wright in [17]. Subfamilies of the family of the regular languages can also be motivated by their specific representations as regular expressions, for example, the family of the union-free languages which are obtained by concatenation and the Kleene star operation (but without union) or the family of the star-free languages which are obtained by concatenation, union, and complement (but without the Kleene star operation) and which are exactly those languages accepted by non-counting automata ([21]).

In the last years, several papers have been published in which, for different problems, the decrease of descriptional or computational complexity was studied when going from arbitrary regular languages to special ones. Especially the effect of subregular control was studied for

- *tree controlled grammars with subregular control languages* by J. Dassow, R. Stiebe, and B. Truthe ([10], [9], [7], [8]),
- *generating and accepting networks of evolutionary processors with subregular communication filters* by J. Dassow, F. Manea, and B. Truthe ([11], [12], [1], [5] for NEPs and [18], [20], [6] for ANEPs),
- *external and internal contextual grammars with subregular selection* by J. Dassow, F. Manea, and B. Truthe ([26] as well as [2], [3] for the external case and [19], [4] for the internal case), and
- *splicing systems with splicing rules which are taken from a subregular set* by J. Dassow and B. Truthe ([13]).

In these papers above, subfamilies of the family of the regular languages have been considered independently of each other. Especially, subfamilies based on structural properties (like the set of all suffix-closed or commutative regular languages) and subfamilies based on ressources needed for generating or accepting them were not related to each other and, hence, also the various devices controlled by such languages were not related to each other.

In this paper, we start to fill this gap by proving inclusion relations and incomparabilities of subfamilies based on different properties.

2 General Definitions and Notation

An alphabet is a finite and non-empty set of symbols (called letters). A word is a finite sequence of letters; the length of a word w is denoted by $|w|$. The empty word does not contain any letter (has the length zero) and is denoted by λ. Let $x_1, x_2, \ldots, x_n$ for some natural number n be letters and $w = x_1 x_2 \cdots x_n$. Then we denote by w^R the mirror word of w (where the letters occur in reversed order): $w^R = x_n x_{n-1} \cdots x_1$.

A set of words over some alphabet V is called a language over the alphabet V. Let V be an alphabet. We use the following notations for sets of words over the alphabet V:

- V^* denotes the set of all words over the alphabet V,
- V^+ denotes the set of all non-empty words: $V^+ = V^* \setminus \{\lambda\}$,
- V^k for a natural number $k \geq 0$ denotes the set of all words with the length k,
- $V^{\leq k}$ for a natural number $k \geq 0$ denotes the set of all words with a length of at most k.

For a word $w \in V^*$ and a set $A \subseteq V$, we denote by $|w|_A$ the number of all occurrences of letters $a \in A$ in the word w. If such a set A consists of a letter a only, we write simply $|w|_a$. The cardinality of a set A is denoted by $|A|$.

The concatenation of two languages U and V is the set of all words obtained by concatenating a word of the language U with a word of the language V:

$$U \cdot V = \{\, uv \mid u \in U \text{ and } v \in V \,\}.$$

For a language L and a natural number $i > 1$, we denote by L^i the concatenation of the language L^{i-1} with the language L (note that $L^1 = L$). Furthermore, $L^0 = \{\lambda\}$. For a language L, we use the notations

$$L^* = \bigcup_{i \geq 0} L^i \quad \text{and} \quad L^+ = \bigcup_{i \geq 1} L^i$$

analogously to the same notation as for alphabets.

Let $V = \{a_1, a_2, \ldots, a_n\}$ be an alphabet with an order

$$a_1 \prec a_2 \prec \cdots \prec a_n.$$

We define the alphabetical order $\prec$ of the words over the alphabet V as follows: For any two numbers n and m and letters $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m$, we say

$$x_1 x_2 \cdots x_n \prec y_1 y_2 \cdots y_m$$

if and only if there is a number k with $k \leq n$ and $k < m$ such that

$$x_1 x_2 \cdots x_k = y_1 y_2 \cdots y_k$$

and if $k < n$ then $x_{k+1} \prec y_{k+1}$.

A phrase structure grammar is a quadruple

$$G = (N, T, P, S)$$

where N and T are two disjoint alphabets (the elements of the alphabet N are called non-terminal symbols; the elements of the alphabet T are called terminal symbols), P is a non-empty and finite subset of $(N \cup T)^* \setminus T^* \times (N \cup T)^*$ (its elements are called rules and are written as $\alpha \to \beta$ instead of (α, β)), and $S \in N$ is the so-called start symbol (also called axiom). A phrase structure grammar is called right-linear if

$$P \subset N \times (T^* N \cup T^*)$$

and regular if

$$P \subset N \times (TN \cup T).$$

Let $G = (N, T, P, S)$ be a phrase structure grammar and let $V = N \cup T$. A word $w \in V^*$ is derived to a word $w' \in V^*$ by the grammar G, written as

$$w \underset{G}{\Longrightarrow} w'$$

or $w \Longrightarrow w'$ if the grammar is known from the context, if there are a decomposition of the word w into three subwords u, α, v such that $w = u\alpha v$, $\alpha \to \beta \in P$, and $w' = u\beta v$ (a subword α

is replaced by a word β if the grammar contains the rule $\alpha \to \beta$). For a natural number k, we say that a word w is derived to a word w' in k steps, written as

$$w \overset{k}{\underset{G}{\Longrightarrow}} w',$$

if there exist words $w_1, w_2, \ldots, w_{k-1}$ such that there is the derivation

$$w \underset{G}{\Longrightarrow} w_1 \underset{G}{\Longrightarrow} w_2 \underset{G}{\Longrightarrow} \cdots \underset{G}{\Longrightarrow} w_{k-1} \underset{G}{\Longrightarrow} w'.$$

The reflexive and transitive closure of the relation $\underset{G}{\Longrightarrow}$ is denoted by

$$\overset{*}{\underset{G}{\Longrightarrow}} .$$

The language $L(G)$ generated by the grammar G is the set of all words that are derivable from the axiom S:

$$L(G) = \left\{ w \;\middle|\; S \overset{*}{\underset{G}{\Longrightarrow}} w \right\}.$$

It is well known that the family of all languages generated by right-linear grammars coincides with the family of all languages generated by regular grammars. The languages of this family are called regular languages.

Regular languages can also be described by regular expressions. Let V be an alphabet. A regular expression is defined inductively as follows:

1. $\emptyset$ is a regular expression;
2. for every element $x \in V$ is x a regular expression;
3. if R and S are regular expressions, so are the concatenation $R \cdot S$, the union $R \cup S$, and the Kleene closure R^*;
4. for every regular expression, there is a natural number n such that the regular expression is obtained from the atomic elements $\emptyset$ and $x \in V$ by n operations concatenation, union, or star.

The language $L(R)$ which is described by a regular expression R is also inductively defined:

1. $L(\emptyset) = \emptyset$;
2. for every element $x \in V$, we have $L(x) = \{x\}$;
3. if R and S are regular expressions, then

$$\begin{aligned} L(R \cdot S) &= L(R) \cdot L(S), \\ L(R \cup S) &= L(R) \cup L(S), \text{ and} \\ L(R^*) &= L^*(R), \end{aligned}$$

where $L^*(R) = (L(R))^*$.

The operator sign $\cdot$ is often omitted; instead of the operator sign $\cup$, the sign $+$ is often used in the literature.

A finite automaton is a quintuple

$$\mathcal{A} = (V, Z, z_0, F, \delta)$$

where V is an alphabet called the input alphabet, Z is a non-empty finite set of elements which are called states, $z_0 \in Z$ is the so-called start state, $F \subseteq Z$ is the set of accepting states, and $\delta : Z \times V \to \mathcal{P}(Z)$ is a mapping which is also called the transition function. A finite automaton is called deterministic if every set $\delta(z, a)$ for $z \in Z$ and $a \in V$ is a singleton set. The transition function δ can be extended to a function $\delta^* : Z \times V^* \to \mathcal{P}(Z)$ where $\delta^*(z, \lambda) = \{z\}$ and

$$\delta^*(z, va) = \bigcup_{z' \in \delta^*(z,v)} \delta(z', a).$$

We will use the same symbol δ in both the original and extended version of the transition function.

Let $\mathcal{A} = (V, Z, z_0, F, \delta)$ be a finite automaton. A word $w \in V^*$ is accepted by the finite automaton $\mathcal{A}$ if and only if the automaton has reached an accepting state after reading the input word w:

$$\delta(z_0, w) \cap F \neq \emptyset.$$

The language $L(\mathcal{A})$ accepted by the finite automaton $\mathcal{A}$ is the set of all accepted words:

$$L(\mathcal{A}) = \{\, w \mid w \in V \text{ and } \delta(z_0, w) \cap F \neq \emptyset \,\}\,.$$

The language accepted by a finite automaton is always regular; on the other hand, for every regular language, there exists a finite automaton which accepts this language.

Let V be an alphabet and $L \subseteq V^*$ be a language over this alphabet. By D_xL for some word $x \in V^*$, we denote the set

$$D_xL = \{\, w \mid xw \in L \,\}.$$

We define a binary relation $\equiv_L \subseteq V^* \times V^*$ by

$$x \equiv_L y \text{ if and only if } D_xL = D_yL$$

for any two words $x \in V^*$ and $y \in V^*$. The relation $\equiv_L$ is an equivalence relation and it is called the Myhill-Nerode relation of the language L. The number of its equivalence classes is called the index of the relation. The following results by J. R. Myhill and A. Nerode can be found in the book [16] by J. E. Hopcroft and J. D. Ullman. A language L is regular if and only if the index of the relation $\equiv_L$ is finite. The minimal number of states which are necessary for accepting a regular language L by a deterministic finite automaton is the index of the relation $\equiv_L$. Up to isomorphism, the deterministic finite automaton generating a language L with the minimal number of states is unique. It is called the minimal deterministic finite automaton for the language L.

3 Definition of Subregular Language Families

We now define various subfamilies of the family of the regular languages and investigate relations between them. Those families are formed by regular languages with certain further properties. Such properties can be defined with respect to the single words (for instance, that every word has as a special last letter), with respect to the operations applied to atomic regular languages (the empty set and sets with a single letter), with respect to dependencies of words (the membership of a word implies the membership of other words), or with respect to the structure or complexity of grammars generating or auotmata accepting the languages. We call a subfamily of regular languages a subregular family of languages.

3.1 Subregular Families Defined by Structural Properties

We define and investigate here subregular families of languages which have common structural properties.

For a language L over an alphabet V, we set

$$Comm(L) = \{ a_{i_1} \dots a_{i_n} \mid a_1 \dots a_n \in L,\ n \geq 1,\ \{i_1, i_2, \dots, i_n\} = \{1, 2, \dots, n\} \}$$

as the commutative closure (the set of all permutations of words) of the language L,

$$Circ(L) = \{ vu \mid uv \in L,\ u, v \in V^* \}$$

as the circular closure (the set of all circular shifts of words) of the language L, and

$$Suf(L) = \{ v \mid uv \in L,\ u, v \in V^* \}$$

as the suffix closure (the set of all suffixes of words) of the language L.

We consider the following restrictions for regular languages. Let L be a language over an alphabet V. We say that the language L – with respect to the alphabet V – is

- *combinational* if and only if it has the form

$$L = V^* A$$

  for some subset $A \subseteq V$,
- *definite* if and only if it can be represented in the form

$$L = A \cup V^* B$$

  where A and B are finite subsets of V^*,
- *nilpotent* if and only if it is finite or its complement $V^* \setminus L$ is finite,
- *commutative* if and only if it contains with each word also all permutations of this word or equivalently,

$$L = Comm(L),$$

- *circular* if and only if it contains with each word also all circular shifts of this word or equivalently,

$$L = Circ(L),$$

- *suffix-closed* (or *fully initial* or *multiple-entry* language) if and only if the relation $xy \in L$ for some words $x \in V^*$ and $y \in V^*$ implies the relation $y \in L$ or equivalently,

$$L = Suf(L),$$

- *non-counting* (or *star-free*) if and only if there is a natural number $k \geq 1$ such that, for any words $x \in V^*$, $y \in V^*$, and $z \in V^*$, it holds

$$xy^k z \in L \text{ if and only if } xy^{k+1} z \in L,$$

- *power-separating* if and only if, there is a natural number $m \geq 1$ such that for any $x \in V^*$, either

$$J_x^m \cap L = \emptyset$$

  or

$$J_x^m \subseteq L$$

  where

$$J_x^m = \{ x^n \mid n \geq m \},$$

– *ordered* if and only if the language L is accepted by some finite automaton $\mathcal{A} = (V, Z, z_0, F, \delta)$ where $(Z, \preceq)$ is a totally ordered set and, for any $a \in V$, the relation

$$z \preceq z' \text{ implies } \delta(z,a) \preceq \delta(z',a),$$

– *union-free* if and only if L can be described by a regular expression which is only built by product and star,
– *monoidal* if and only if $L = V^*$.

We remark that combinational, definite, nilpotent, ordered, union-free, and monoidal languages are regular, whereas non-regular languages of the other types mentioned above exist. Here, we consider among the commutative, circular, suffix-closed, non-counting, and power-separating languages only those which are also regular. So, we do not necessarily mention the regularity then.

By *COMB*, *DEF*, *NIL*, *COMM*, *CIRC*, *SUF*, *NC*, *PS*, *ORD*, *UF*, *MON*, and *REG* we denote the families of all combinational, definite, nilpotent, regular commutative, regular circular, regular suffix-closed, regular non-counting, regular power-separating, ordered, union-free, monoidal, and regular languages, respectively. Moreover, we add the family *FIN* of all finite languages. We set

$$\mathcal{F} = \{\mathit{MON}, \mathit{FIN}, \mathit{COMB}, \mathit{NIL}, \mathit{DEF}, \mathit{ORD}, \mathit{NC}, \mathit{PS}, \mathit{SUF}, \mathit{COMM}, \mathit{CIRC}, \mathit{UF}\}.$$

Set-theoretic relations between families of the set $\mathcal{F}$ are investigated, e. g., in [14], [15], [24], [25], and [27]. Further relations will be proven in Section 4.

3.2 Subregular Families Defined by the Number of Resources

We now define families of regular languages by restricting the resources needed for generating or accepting them.

Let $G = (N, T, P, S)$ be a right-linear grammar, $\mathcal{A} = (V, Z, z_0, F, \delta)$ be a deterministic finite automaton, and L be a regular language. Then we define the following measures of descriptional complexity:

$$\begin{aligned} \mathit{Var}(G) &= |N|, \\ \mathit{Prod}(G) &= |P|, \\ \mathit{State}(A) &= |Z|. \end{aligned}$$

The descriptional complexity of a regular language L with respect to the number of nonterminals, production rules, or states needed for generating or accepting the language L is the minimal number of the respective resources necessary. For the generating case, we distinguish between generating the language L by a regular grammar or by an arbitrary right-linear grammar:

$$\begin{aligned} \mathit{Var}_{RL}(L) &= \min\{\, \mathit{Var}(G) \mid G \text{ is a right-linear grammar generating } L\,\}, \\ \mathit{Prod}_{RL}(L) &= \min\{\, \mathit{Prod}(G) \mid G \text{ is a right-linear grammar generating } L\,\}, \\ \mathit{Var}_{REG}(L) &= \min\{\, \mathit{Var}(G) \mid G \text{ is a regular grammar generating } L\,\}, \\ \mathit{Prod}_{REG}(L) &= \min\{\, \mathit{Prod}(G) \mid G \text{ is a regular grammar generating } L\,\}, \\ \mathit{State}(L) &= \min\{\, \mathit{State}(A) \mid A \text{ is a det. finite automaton accepting } L\,\}. \end{aligned}$$

For these complexity measures, we define the following families of languages (we abbreviate the measure *Var* by V, the measure *Prod* by P, and the measure *State* by Z):

$$\begin{aligned}
RL_n^V &= \{\, L \mid L \text{ is a regular language with } \mathit{Var}_{RL}(L) \leq n \,\}\,,\\
RL_n^P &= \{\, L \mid L \text{ is a regular language with } \mathit{Prod}_{RL}(L) \leq n \,\}\,,\\
REG_n^V &= \{\, L \mid L \text{ is a regular language with } \mathit{Var}_{REG}(L) \leq n \,\}\,,\\
REG_n^P &= \{\, L \mid L \text{ is a regular language with } \mathit{Prod}_{REG}(L) \leq n \,\}\,,\\
REG_n^Z &= \{\, L \mid L \text{ is a regular language with } \mathit{State}(L) \leq n \,\}\,.
\end{aligned}$$

Since every regular grammar is also right-linear, the number of resources needed by a regular grammar is not smaller (and could be greater) than the number of resources needed by an arbitrary right-linear grammar. Therefore, the inclusion

$$REG_n^K \subseteq RL_n^K$$

holds for every natural number $n \geq 1$ and complexity measure $K \in \{V, P\}$.

Regarding the ressources, we will consider here in this paper the families RL_n^K for $n \geq 1$ and $K \in \{V, P\}$ as well as REG_n^Z for $n \geq 1$.

4 Hierarchies of Subregular Families

In this section, we relate the subfamilies of regular languages which we have introduced in the previous section. We prove proper inclusions and incomparabilities between such families. The hierarchies obtained are presented graphically. We first deduce a hierarchy of subregular families which are defined by structural properties and then a hierarchy of subregular families which are defined by restricting the number of resources needed for generating or accepting the respective languages.

4.1 Structurally Defined Subregular Families

Set-theoretic relations between families of the set

$$\mathcal{F} = \{\mathit{MON}, \mathit{FIN}, \mathit{COMB}, \mathit{NIL}, \mathit{DEF}, \mathit{ORD}, \mathit{NC}, \mathit{PS}, \mathit{SUF}, \mathit{COMM}, \mathit{CIRC}, \mathit{UF}\}$$

are investigated, e. g., in [14], [15], [24], [25], and [27]. In these papers, proper inclusions and incomparabilities are proven.

It only remains to show the following statement.

Lemma 1. *The proper inclusion $\mathit{COMB} \subset \mathit{DEF}$ holds.*

Proof. The inclusion follows from the definition and was already stated in the paper [14]. By definition, every finite language is also definite but never combinational. This proves the properness of the inclusion. □

Summarizing, the hierarchy shown in Figure 1 is obtained.

Theorem 2. *The inclusion relations presented in Figure 1 hold. An arrow from an entry X to an entry Y depicts the proper inclusion $X \subset Y$; if two families are not connected by a directed path, then they are incomparable.*

Proof. The labels at the arrows in Figure 1 refer to the paper or the lemma where the respective inclusion is proven. The incomparabilities have all been proven in [15]. □

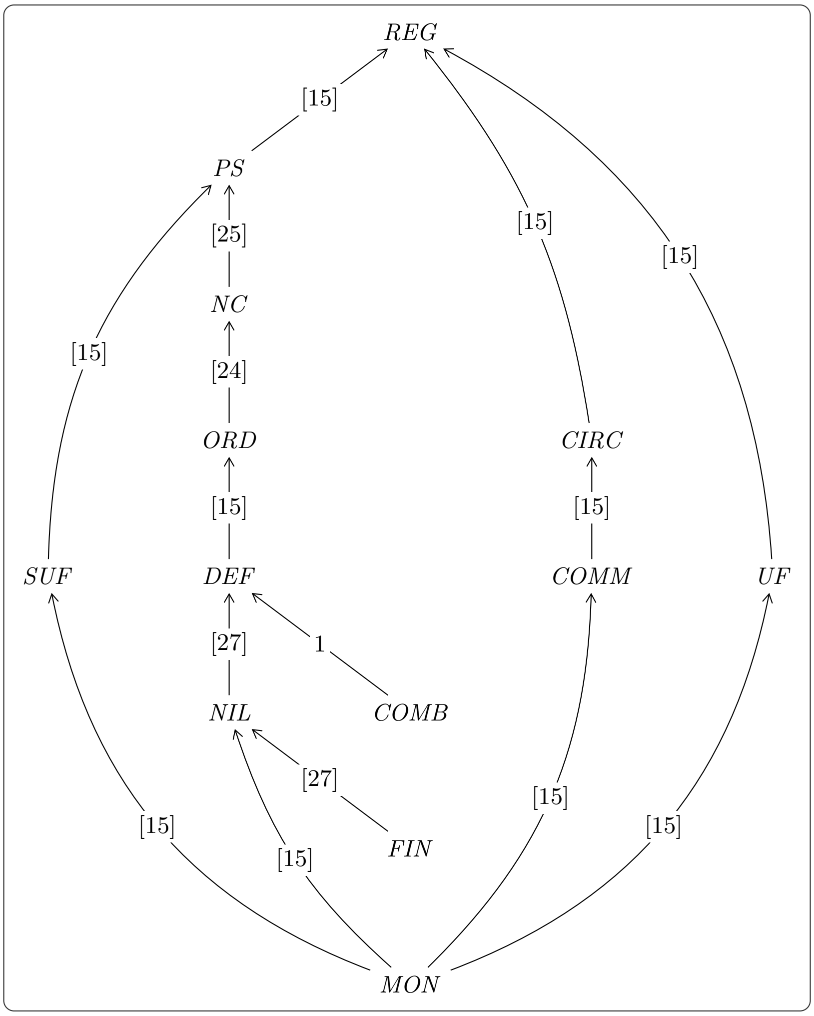


Figure 1: Hierarchy of subregular families defined by structural properties

4.2 Subregular Families Defined by the Number of Resources

We now state the inclusion relations between the families REG_n^Z and RL_n^K for the complexity measures $K \in \{V, P\}$ and $n \geq 1$. The hierarchy of the families is presented in Figure 2. An arrow from an entry X to an entry Y denotes the proper inclusion $X \subset Y$. If two families are not connected by a directed path, then they are incomparable.

We first prove the inclusion relations, then we present witness languages for their properness and for the incomparabilities, and finally, we prove the properness of every inclusion and each comparability.

Lemma 3. *For each number $n \geq 1$ and complexity measure $K \in \{V, P\}$, we have the inclusion $RL_n^K \subseteq RL_{n+1}^K$ and the inclusion $REG_n^Z \subseteq REG_{n+1}^Z$.*

Proof. Every language which is generated by a grammar with a certain number of resources can also be generated by a grammar with more resources (for instance, a grammar with the same resources and additional but unused resources). $\square$

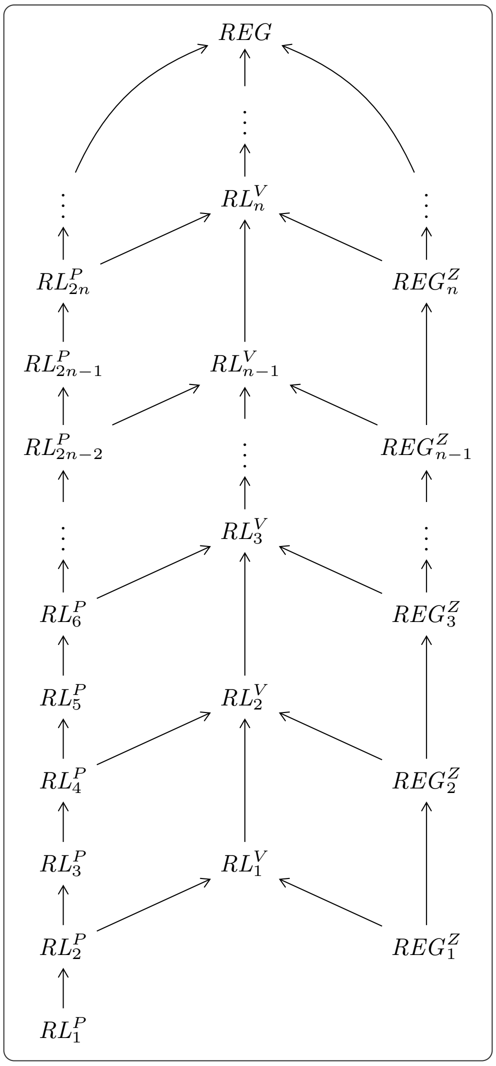


Figure 2: Hierarchy of subregular language families with bounded resources

Lemma 4. *For each number $n \geq 1$, the inclusion*

$$REG_n^Z \subseteq RL_n^V$$

holds.

Proof. Let n be a natural number with $n \geq 1$ and $L_n \in REG_n^Z$. Then there is a deterministic finite automaton $\mathcal{A}_n = (V, Z, z_0, F, \delta)$ which has at most n states and accepts the language L_n. We construct a regular grammar $G_n = (N, V, P, S_{z_0})$ to the automaton $\mathcal{A}_n$ where we assign a non-terminal to each state:

$$N = \{ S_z \mid z \in Z \}.$$

The rules correspond to the transitions:

$$P = \{ S_z \to xS_{z'} \mid \delta(z, x) = z' \} \cup \{ S_z \to \lambda \mid z \in F \}.$$

The grammar G_n generates the same language which is accepted by the automaton $\mathcal{A}_n$ because it simulates a transition step by the application of a rule. Since $|N| = |Z|$, we obtain the inclusion $REG_n^Z \subseteq RL_n^V$. $\square$

Lemma 5. *For each number $n \geq 1$, the inclusion*

$$RL^P_{2n} \subseteq RL^V_n$$

holds.

Proof. Let n be a natural number with $n \geq 1$ and $L_n \in RL^P_{2n}$. The language L_n is generated by a right-linear grammar which has not more than $2n$ rules. Let

$$G_n = (N, T, P, S)$$

be such a grammar with the minimal number of rules. Then, for every rule $A \to w \in P$ with $w \in T^*N \cup T^*$ and $A \neq S$, there is also another rule for the non-terminal A. Otherwise, in every rule where the non-terminal A occurs on the right-hand side, one could replace A by the word w which makes the rule $A \to w$ superflous. But then the grammar G_n would not be minimal with respect to the number of rules. Hence, for every non-terminal $A \neq S$ (for which a rule exists), there are two rules in the grammar. Thus, the number of the non-terminal symbols occurring on the left-hand sides of the rules is at most

$$\frac{|P|-1}{2} + 1 = \frac{|P|+1}{2}.$$

Since $|P| \leq 2n$, the number of sufficient non-terminals is at most

$$\frac{2n+1}{2}.$$

The number of non-terminals is a natural number, hence, n non-terminals are sufficient for generating the language L_n. Thus, $L_n \in RL^V_n$ and $RL^P_{2n} \subseteq RL^V_n$. □

For proving the properness of the inclusions and the incomparabilities, we use several witness languages for which we state membership properties in the sequel.

Lemma 6. *Let n be a natural number with $n \geq 1$ and V be an alphabet with n different letters $a_1, a_2, \ldots, a_n$. Further, let $L_n = V^*$. Then the relation*

$$L_n \in (RL^P_{n+1} \cap RL^V_1 \cap REG^Z_1) \setminus RL^P_n$$

holds.

Proof. Let n be a natural number with $n \geq 1$. The language L_n can be generated by a regular grammar with $n+1$ rules:

$$G_n = (\{S\}, V, \{\, S \to a_iS \mid 1 \leq i \leq n \,\} \cup \{S \to \lambda\}, S).$$

Hence, $L_n \in RL^P_{n+1} \cap RL^V_1$. In order to generate the language L_n, one needs at least a rule for every letter a_i and the empty word λ. Hence, $L_n \notin RL^P_n$.

The language L_n is accepted by a deterministic finite automaton

$$\mathcal{A}_n = (V, \{z\}, z, \{z\}, \delta)$$

with the transition mapping δ defined by $\delta(z, x) = z$ for $x \in V$. Hence, we have the last result $L_n \in REG^Z_1$. □

Lemma 7. *Let n be a natural number with $n \geq 1$ and $V = \{a_1, a_2, \ldots, a_{2n}\}$ be an alphabet with n different letters. Further, let $L_n = V^*$. Then $L_n \in RL_1^V \setminus RL_{2n}^P$.*

Proof. Let n be a natural number with $n \geq 1$. The language $L_n = V^*$ can be generated by the regular grammar

$$G_n = (\{S\}, V, \{\, S \to a_i S \mid 1 \leq i \leq 2n \,\} \cup \{S \to \lambda\}, S),$$

hence, with only one non-terminal. In order to generate the language L_n, one needs at least a rule for every letter a_i and the empty word λ. Hence, $2n$ rules are not sufficient. □

Lemma 8. *Let n be a natural number with $n \geq 1$ and $V = \{a, b\}$. Further, let*

$$L_n = ((\{b\}^*\{a\})^n\{b\}^*)^+.$$

Then $L_n \in (RL_{n+1}^V \cap REG_{n+1}^Z) \setminus (RL_n^V \cup REG_n^Z)$.

Proof. Let n be a natural number with $n \geq 1$. The language L_n can be generated by a regular grammar with $n+1$ non-terminal symbols:

$$G_n = (\{S_1, S_2, \ldots, S_{n+1}\}, V, P_n, S_1)$$

where the set P_n of rules is

$$P_n = \{\, S_i \to bS_i \mid 1 \leq i \leq n+1 \,\} \cup \{\, S_i \to aS_{i+1} \mid 1 \leq i \leq n \,\} \cup \{S_n \to aS_1, S_{n+1} \to \lambda\}.$$

Let us assume that the language L_n can be generated by a grammar with at most n non-terminals $A_1, A_2, \ldots, A_n$ where the start symbol is A_1. Then there is a derivation

$$\begin{aligned} A_1 &\stackrel{*}{\Longrightarrow} b^{\ell'_1} A_{i_1} \stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell'_2} A_{i_2} \stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell'_3} A_{i_3} \\ &\stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell'_n} A_{i_n} \stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell_n} a b^{\ell'_{n+1}} A_{i_{n+1}} \\ &\stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell_n} a b^{\ell_{n+1}} \end{aligned}$$

of a word with exactly n letters a for numbers $ij \in \{1, \ldots, n+1\}$ with $0 \leq j \leq n+1$ and $\ell'_i \geq 1$ and $\ell_i > \ell'_i$ with $1 \leq i \leq n+1$. Since there are only n different non-terminal symbols, there are two equal indices i_r and i_s with $1 \leq r < s \leq n+1$. If $r = 1$ and $s = n+1$, then there exists also the derivation

$$A_1 \stackrel{*}{\Longrightarrow} b^{\ell'_1} A_{i_1} \stackrel{*}{\Longrightarrow} b^{\ell'_1} b^{\ell_{n+1} - \ell'_{n+1}}.$$

Hence, a word of the set $\{b\}^+$ is generated which does not belong to the language L_n. Otherwise (if $r > 1$ or $s \leq n$), there exists also the derivation

$$\begin{aligned} A_1 &\stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell'_s} A_{i_s} \stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell'_s} b^{\ell_r - \ell'_r} a b^{\ell_{r+1}} \cdots a b^{\ell'_s} A_{i_s} \\ &\stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell'_s} b^{\ell_r - \ell'_r} a b^{\ell_{r+1}} \cdots a b^{\ell_s} \cdots a b^{\ell'_{n+1}} A_{i_{n+1}} \\ &\stackrel{*}{\Longrightarrow} b^{\ell_1} a b^{\ell_2} a b^{\ell_3} \cdots a b^{\ell'_s} b^{\ell_r - \ell'_r} a b^{\ell_{r+1}} \cdots a b^{\ell_s} \cdots a b^{\ell_{n+1}} \end{aligned}$$

to a word with exactly $n + r - s$ letters a (more than n but less than $2n$ letters a). Also such a word does not belong to the language L_n. This contradiction implies that the language L_n cannot be generated by a right-linear grammar with at most n non-terminal symbols.

The language L_n is accepted by a deterministic finite automaton

$$\mathcal{A}_n = (V, \{z_1, z_2, \ldots, z_{n+1}\}, z_1, \{z_{n+1}\}, \delta)$$

with the transition mapping δ defined by

$$\delta(z_i, a) = z_{i+1}$$

for $1 \leq i \leq n$ and

$$\delta(z_{n+1}, a) = z_2$$

as well as

$$\delta(z_i, b) = z_i$$

for $1 \leq i \leq n+1$. Hence, we have $L_n \in REG^Z_{n+1}$.

Let $c_i = a^{n-i}$ for $0 \leq i \leq n$. Then, for $0 \leq i \leq n$, we have

$$ba^i c_i \in L_n \quad \text{and} \quad ba^j c_i \notin L_n$$

for $0 \leq j < i \leq n$. Therefore, the words $b, ba, ba^2, \ldots, ba^n$ are pairwise not in the Myhill-Nerode relation. Thus, the minimal determinsitic finite automaton accepting the language L_n has at least $n+1$ states and, therefore, $L_n \notin REG^Z_n$. $\square$

Lemma 9. *Let n be a natural number with $n \geq 1$ and $V_n = \{a_1, a_2, \ldots, a_n, b\}$ be an alphabet with n different letters. Further, let*

$$L_n = \{b\}\{a_1\}^+\{a_2\}^+ \cdots \{a_n\}^+.$$

Then $L_n \in (RL^V_{n+1} \cap RL^P_{2n+1}) \setminus (RL^V_n \cup RL^P_{2n})$.

Proof. Let n be a natural number with $n \geq 1$. The language L_n can be generated by a regular grammar with $n+1$ non-terminal symbols and $2n+1$ rules:

$$G_n = (\{S_0, S_1, \ldots, S_n\}, V_n, P_n, S_0)$$

where the set P_n of rules is

$$P_n = \{S_0 \to bS_1\} \cup \{\, S_i \to a_i S_i \mid 1 \leq i \leq n \,\} \cup \{\, S_i \to a_i S_{i+1} \mid 1 \leq i \leq n-1 \,\} \cup \{S_n \to a_n\}.$$

Let us assume that the language L_n can be generated by a grammar with at most n non-terminal symbols $A_1, A_2, \ldots, A_n$. Then there is a derivation

$$\begin{aligned} A_{i_0} &\overset{*}{\Longrightarrow} ba_1^{\ell'_1} A_{i_1} \overset{*}{\Longrightarrow} ba_1^{\ell_1} a_2^{\ell'_2} A_{i_2} \overset{*}{\Longrightarrow} ba_1^{\ell_1} a_2^{\ell_2} a_3^{\ell'_3} A_{i_3} \overset{*}{\Longrightarrow} ba_1^{\ell_1} a_2^{\ell_2} a_3^{\ell_3} \cdots a_n^{\ell'_n} A_{i_n} \\ &\overset{*}{\Longrightarrow} ba_1^{\ell_1} a_2^{\ell_2} a_3^{\ell_3} \cdots a_n^{\ell_n} \end{aligned}$$

of a word of the language L_n (for certain numbers $i_j \in \{1, \ldots, n\}$ with $0 \leq j \leq n$ and $\ell'_j \geq 1$ and $\ell_j > \ell'_j$ with $1 \leq j \leq n$). Since there are only n different non-terminal symbols, there are two equal indices i_r and i_s with $0 \leq r < s \leq n$. If $r = 0$, then there exists also the derivation

$$A_{i_0} = A_{i_r} = A_{i_s} \overset{*}{\Longrightarrow} a_s^{\ell_s - \ell'_s} a_{s+1}^{\ell_{s+1}} \cdots a_n^{\ell_n}.$$

Hence, a word is generated which does not belong to the language L_n. Otherwise (if $r > 0$), there exists also the derivation

$$\begin{aligned} A_{i_0} &\overset{*}{\Longrightarrow} ba_1^{\ell_1}a_2^{\ell_2}a_3^{\ell_3}\cdots a_s^{\ell'_s}A_{i_s} \\ &\overset{*}{\Longrightarrow} ba_1^{\ell_1}a_2^{\ell_2}a_3^{\ell_3}\cdots a_s^{\ell'_s}a_r^{\ell_r-\ell'_r}a_{r+1}^{\ell'_{r+1}}A_{i_{r+1}} \\ &\overset{*}{\Longrightarrow} ba_1^{\ell_1}a_2^{\ell_2}a_3^{\ell_3}\cdots a_s^{\ell'_s}a_r^{\ell_r-\ell'_r}a_{r+1}^{\ell_{r+1}}\cdots a_n^{\ell_n} \end{aligned}$$

to a word which contains $a_s a_r$ as a subword but such a word does not belong to the language L_n because $r < s$. This contradiction implies that the language L_n cannot be generated by a right-linear grammar with at most n non-terminal symbols.

From Lemma 5, we know the inclusion

$$RL_{2n}^P \subseteq RL_n^V.$$

Since the language L_n does not belong to the class RL_n^V it does not belong to the class RL_{2n}^P either. □

Lemma 10. *Let n be a natural number with $n \geq 1$ and $V = \{a\}$. Further, let*

$$L_n = \{a^{n+1}\}^*.$$

Then $L_n \in (REG_{n+1}^Z \cap RL_1^V \cap RL_2^P) \setminus REG_n^Z$.

Proof. Let n be a natural number with $n \geq 1$. Let $V = \{a\}$ and $L_n = \{a^{n+1}\}^*$. This language is accepted by a deterministic finite automaton

$$\mathcal{A}_n = (V, \{z_0, z_1, \ldots, z_n\}, z_0, \{z_0\}, \delta)$$

with the transition mapping δ defined by

$$\delta(z_i, a) = z_{(i+1) \mod (n+1)}.$$

Hence, we have $L_n \in REG_{n+1}^Z$. Let $c_i = a^{n+1-i}$ for $1 \leq i \leq n+1$. Then

$$a^i c_i \in L_n \quad \text{and} \quad a^j c_i \notin L_n$$

for $1 \leq j < i \leq n+1$. Therefore, the words $a, a^2, \ldots, a^{n+1}$ are pairwise not in the Myhill-Nerode relation. Thus, the minimal determinsitic finite automaton accepting the language L_n has at least $n+1$ states and, therefore, $L_n \notin REG_n^Z$.

The language L_n can be generated by the right-linear grammar

$$G_n = (\{S\}, V, \{S \to a^{n+1}S, S \to \lambda\}, S)$$

with one non-terminal and two rules. Hence, $L_n \in RL_1^V \cap RL_2^P$. □

Lemma 11. *Let n be a natural number with $n \geq 1$ and let*

$$L_n = \{a^{n+1}\}.$$

Then $L_n \in (REG_{n+2}^Z \cap RL_1^V \cap RL_1^P) \setminus REG_{n+1}^Z$.

Proof. Let n be a natural number with $n \geq 1$. Let $V = \{a\}$ and $L_n = \{a^{n+1}\}$. This language is accepted by a deterministic finite automaton

$$\mathcal{A}_n = (V, \{z_0, z_1, \ldots, z_n, z_{n+1}, z_{n+2}\}, z_0, \{z_{n+1}\}, \delta)$$

with the transition mapping δ defined by

$$\delta(z_i, a) = z_{i+1}$$

for $0 \leq i \leq n$ and

$$\delta(z_{n+1}, a) = z_{n+1}.$$

Hence, we have $L_n \in REG_{n+2}^Z$. Let $c_i = a^{n+1-i}$ for $1 \leq i \leq n+1$. Then

$$a^i c_i \in L_n \quad \text{and} \quad a^j c_i \notin L_n$$

for $0 \leq j < i \leq n+1$. Therefore, the words $\lambda, a, a^2, \ldots, a^{n+1}$ are pairwise not in the Myhill-Nerode relation. Thus, the minimal determinsitic finite automaton accepting the language L_n has at least $n+2$ states and, therefore, $L_n \notin REG_{n+1}^Z$.

The language L_n can be generated by the right-linear grammar

$$G_n = (\{S\}, V, \{S \to a^{n+1}\}, S)$$

with one non-terminal and one rule. Hence, $L_n \in RL_1^V \cap RL_1^P$. □

We now prove the properness of every inclusion depicted in Figure 2.

Lemma 12. *For each number $n \geq 1$ and complexity measure $K \in \{V, P\}$, we have the proper inclusion $RL_n^K \subset RL_{n+1}^K$ and the proper inclusion $REG_n^Z \subset REG_{n+1}^Z$.*

Proof. The inclusions are shown in Lemma 3.

We now prove that all these inclusions are proper. Let n be a natural number with $n \geq 1$.

First, we consider the number of production rules. Let V be an alphabet with n different letters $a_1, a_2, \ldots, a_n$ and let

$$L_n = V^*.$$

According to Lemma 6, $L_n \in RL_{n+1}^P \setminus RL_n^P$.

Now, we consider the number of non-terminal symbols. Let

$$L_n = \{b\}\{a_1\}^+\{a_2\}^+ \cdots \{a_n\}^+.$$

According to Lemma 9, $L_n \in RL_{n+1}^V \setminus RL_n^V$.

Now, we consider the number of states. Let

$$L_n = \{a^{n+1}\}^*.$$

According to Lemma 10, $L_n \in REG_{n+1}^Z \setminus REG_n^Z$. □

Lemma 13. *For each number $n \geq 1$, the proper inclusion*

$$REG_n^Z \subset RL_n^V$$

holds.

Proof. The inclusion was shown in Lemma 4.

From Lemma 10, we know that the language

$$L_n = \{a^{n+1}\}^*$$

does not belong to the family REG_n^Z but it can be generated by the grammar with only one non-terminal symbol. Hence, it holds $L_n \in RL_1^V$ and, according to Lemma 12, we also have the relation $L_n \in RL_n^V$. Thus, $L_n \in RL_n^V \setminus REG_n^Z$ which proves the properness of the inclusion. □

Lemma 14. *For each number $n \geq 1$, the proper inclusion*

$$RL_{2n}^P \subset RL_n^V$$

holds.

Proof. The inclusion is shown in Lemma 5.

Now, let $V = \{a_1, a_2, \ldots, a_{2n}\}$ and $L_n = V^*$. According to Lemma 7, this language can be generated by a regular grammar with only one non-terminal but

$$L_n \notin RL_{2n}^P.$$

Since $L_n \in RL_1^V$, we know from Lemma 12 that also $L_n \in RL_n^V$ holds. Thus,

$$L_n \in RL_n^V \setminus RL_{2n}^P$$

which proves the properness of the inclusion. □

We now prove the incomparabilities of the hierarchy.

Lemma 15. *For each number $n \geq 1$, the family RL_n^V is incomparable to each family REG_m^Z with $m > n$.*

Proof. Let n and m be two natural numbers with $n \geq 1$ and $m > n$. The language

$$L_m = \{a^{m+1}\}^*$$

belongs to the family RL_1^V and, according to Lemma 12, also to the family RL_n^V but not to the family REG_m^Z (Lemma 10).

The language

$$K_n = ((\{b\}^*\{a\})^n\{b\}^*)^*$$

belongs to the family REG_{n+1}^Z (Lemma 8) and, according to Lemma 12, also to the family REG_m^Z but not to the family RL_n^V (Lemma 8). □

Lemma 16. *For each number $n \geq 1$, the family RL_n^V is incomparable to each of the families RL_m^P with $m > 2n$.*

Proof. Let n and m be two natural numbers with $n \geq 1$ and $m > 2n$. The language

$$L_m = \{a_1, a_2, \ldots, a_m\}^*$$

belongs to the family RL_1^V (Lemma 6) and, according to Lemma 12, also to the family RL_n^V but not to the family RL_m^P (Lemma 6).

The language

$$K_n = \{b\}\{a_1\}^+\{a_2\}^+ \cdots \{a_n\}^+$$

belongs to the family RL_{2n+1}^P (Lemma 9) and, according to Lemma 12, also to the family RL_m^P but not to the family RL_n^V (Lemma 9). □

Lemma 17. *For every two numbers $n \geq 1$ and $m \geq 1$, the families RL_n^P and REG_m^Z are incomparable.*

Proof. Let n and m be two natural numbers with $n \geq 1$ and $m \geq 1$. The language

$$L_m = \{a^{m+1}\}$$

belongs to the family RL_1^P (Lemma 11) and, according to Lemma 12, also to the family RL_n^P but not to the family REG_{m+1}^Z (Lemma 11) and, according to Lemma 12, also not to the family REG_m^Z.

The language

$$K_n = \{a_1, a_2, \ldots, a_n\}^*$$

belongs to the family REG_1^Z (Lemma 6) and, according to Lemma 12, also to the family REG_m^Z but not to the family RL_n^P (Lemma 6). □

Summarizing, the proper inclusions and incomparabilities shown in Figure 2 are proven.

Theorem 18. *The inclusion relations presented in Figure 2 hold. An arrow from an entry X to an entry Y depicts the proper inclusion $X \subset Y$; if two families are not connected by a directed path, then they are incomparable.*

Proof. The labels at the arrows in Figure 2 refer to the statement where the respective inclusion is proven. □

5 Comparing the Families of the Hierarchies

We have defined and investigated subregular families of languages which have common structural properties and families of regular languages defined by restricting the resources needed for generating or accepting them. We now relate the families of these two kinds. First, we present languages which will serve later as witness languages for proper inclusions or incomparabilities.

Lemma 19. *Let n be a natural number with $n \geq 1$ and let*

$$L_n = \{a^{n+1}\}.$$

Then $L_n \in (FIN \cap UF \cap COMM) \setminus REG_{n+1}^Z$.

Proof. Each language L_n is finite, commutative, can be represented as a finite concatenation of letters a, and can be generated by a regular grammar with one rule.

Let n be a natural number with $n \geq 1$. According to Lemma 11, the language L_n cannot be accepted by a finite automaton with at most $n+1$ states. □

Lemma 20. *Let n be a natural number with $n \geq 1$ and*

$$L_n = \{\, a^i \mid 0 \leq i \leq n \,\}.$$

Then $L_n \in (SUF \cap NIL \cap COMM) \setminus REG_{n+1}^Z$.

Proof. Let n be a natural number with $n \geq 1$ and

$$L_n = \{\, a^i \mid 0 \leq i \leq n \,\}.$$

The language is finite and, therefore, also nilpotent. For every word of the language L_n, also all suffixes and permutations of this word belong to the language. Hence,

$$L_n \in \mathit{SUF} \cap \mathit{NIL} \cap \mathit{COMM}.$$

Let $c_i = a^{n+1-i}$ for $0 \leq i \leq n+1$. Then $a^i c_i \notin L_n$ and $a^j c_i \in L_n$ for $0 \leq j < i \leq n+1$. Therefore, the words $\lambda, a, a^2, \ldots, a^{n+1}$ are pairwise not in the Myhill-Nerode relation. Thus, the minimal determinsitic finite automaton accepting the language L_n has at least $n+2$ states and, therefore, $L_n \notin \mathit{REG}^Z_{n+1}$. ☐

Lemma 21. *Let*

$$L = \{aa\}^*.$$

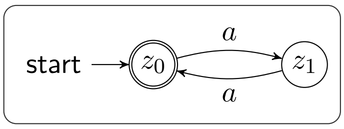
Then $L \in (\mathit{REG}^Z_2 \cap \mathit{RL}^V_1 \cap \mathit{RL}^P_2) \setminus \mathit{PS}$.

Proof. The language L is accepted by the finite automaton

$$\mathcal{A} = (\{a\}, \{z_0, z_1\}, z_0, \{z_0\}, \delta)$$

where the transition function δ is given by the following table (which is illustrated in the diagram next to it)

	z_0	z_1
a	z_1	z_0



which shows that the language L can be accepted by a deterministic finite automaton with at most two states.

The language L is generated by the right-linear grammar

$$G = (\{S\}, \{a\}, P, S)$$

where the set P consists of the rules $S \to aaS$ and $S \to \lambda$. Hence,

$$L \in \mathit{RL}^V_1 \cap \mathit{RL}^V_2.$$

The language L is not power-separating since for every natural number $m \geq 1$, it holds

$$J^m_a \cap L \neq \emptyset \quad \text{and} \quad J^m_a \not\subseteq L$$

with

$$J^m_a = \{\, a^n \mid n \geq m \,\}$$

(for every natural number $m \geq 1$, we have $a^{2m} \in L$ and $a^{2m+1} \in L$).

Thus, $L \in (\mathit{REG}^Z_2 \cap \mathit{RL}^V_1 \cap \mathit{RL}^V_2) \setminus \mathit{PS}$. ☐

Lemma 22. *Let*

$$L = \{ab\}^*.$$

Then $L \in (RL_1^V \cap RL_2^P) \setminus (SUF \cup DEF)$.

Proof. The language L is generated by the right-linear grammar

$$G = (\{S\}, \{a, b\}, P, S)$$

where the set P consists of the rules $S \to abS$ and $\to \lambda$. Hence,

$$L \in RL_1^V \cap RL_2^V.$$

The language is neither suffix-closed (because b is a suffix of the word $ab \in L$ but the word b does not belong to the language L) nor definite (because otherwise the language L would contain a sufficiently long word which starts with a letter b which is a contradiction). $\square$

Lemma 23. *Let*

$$L = \{a\} \cup \{a, b\}^*\{b\} \cup \{\lambda\}.$$

Then $L \in (SUF \cap DEF) \setminus (RL_1^V \cup RL_3^P)$.

Proof. For every word of the language L, also each of its suffixes is contained. Hence, the language L is suffix-closed. The language L can be written as $L = A \cup V^*B$ with

$$V = \{a, b\},$$
$$A = \{a, \lambda\},$$
$$B = \{b\}.$$

Hence, the language L is also definite.

Let $G = (N, \{a, b\}, P, S)$ be a right-linear grammar which generates the language L. Since the empty word λ and the word a belong to the language, there are derivations

$$S \overset{*}{\Longrightarrow} R \Longrightarrow \lambda \text{ and}$$
$$S \overset{*}{\Longrightarrow} T \Longrightarrow a.$$

For generating an arbitrarily long word of the set V^*B (with a length greater than one), a derivation of the form

$$S \overset{*}{\Longrightarrow} uS_1 \overset{*}{\Longrightarrow} uvS_1 \overset{*}{\Longrightarrow} uvwS_2 \overset{*}{\Longrightarrow} uvwzb$$

is necessary where $\{S, S_1, S_2\} \subseteq N$ (the non-terminal symbols S, S_1, S_2 are not necessarily different), $uvwz \in V^+$, and $v \in V^+$. The non-terminal symbols T and S_2 must be different because otherwise the word $uvwa$ could be derived which does not belong to the language L (because $v \in V^+$). Hence, one non-terminal is not sufficient for generating the language L.

Regarding the number of production rules: One needs terminating rules for the words λ and a as well as a terminating rule for producing the letter b at the end of every other word. Further, on needs at least one non-terminating rule for the loop part $S_1 \overset{*}{\Longrightarrow} vS_1$ of the derivation above. Hence, three rules are not sufficient.

Thus, we obtain $L \notin RL_1^V \cup RL_3^P$. $\square$

Lemma 24. *Let n be a natural number with $n \geq 1$ and let*

$$L_n = ((\{b\}^*\{a\})^n\{b\}^*)^*$$

Then $L_n \in (COMM \cap UF) \setminus RL_n^V$.

Proof. Let n be a natural number with $n \geq 1$. The representation of the language L_n as

$$L_n = ((\{b\}^*\{a\})^n\{b\}^*)^*$$

shows that the language L_n is union-free. The statement $L_n \notin RL_n^V$ is known already from Lemma 8. Let $V = \{a, b\}$. The language L_n can also be represented as

$$L_n = \{\, w \mid w \in V \text{ and } |w|_a = kn \text{ for some natural number } k \geq 0 \,\}.$$

This representation shows that the language L_n is also commutative. □

Lemma 25. *Let n be a natural number with $n \geq 1$ and let*

$$L_n = Suf(\{\, w_1aw_2aw_3 \cdots w_naw_{n+1} \mid w_i \in \{b\}^*, 1 \leq i \leq n+1 \,\}).$$

Then $L_n \in (SUF \cap ORD) \setminus RL_n^V$.

Proof. We start with the relation $L_n \notin RL_n^V$.

Let us assume that the language L_n can be generated by a grammar with at most n non-terminals $A_1, A_2, \ldots, A_n$ where the start symbol is A_1. Then there is a derivation

$$\begin{aligned} A_1 &\overset{*}{\Longrightarrow} b^{\ell'_1}A_{i_1} \overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell'_2}A_{i_2} \overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell'_3}A_{i_3} \\ &\overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell'_n}A_{i_n} \overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell_n}ab^{\ell'_{n+1}}A_{i_{n+1}} \\ &\overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell_n}ab^{\ell_{n+1}} \end{aligned}$$

for certain numbers $\ell'_i \geq 1$ and $\ell_i > \ell'_i$ with $1 \leq i \leq n+1$. Since there are only n different non-terminal symbols, there are two equal indices i_r and i_s with $1 \leq r < s \leq n+1$. Hence, there exists also the derivation

$$\begin{aligned} A_1 &\overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell'_s}A_{i_s} \overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell'_s}b^{\ell_r-\ell'_r}ab^{\ell'_{r+1}}A_{i_{r+1}} \\ &\overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell'_s}b^{\ell_r-\ell'_r}ab^{\ell_{r+1}} \cdots ab^{\ell'_s}A_{i_s} \\ &\overset{*}{\Longrightarrow} b^{\ell_1}ab^{\ell_2}ab^{\ell_3} \cdots ab^{\ell'_s}b^{\ell_r-\ell'_r}ab^{\ell_{r+1}} \cdots ab^{\ell_s} \cdots ab^{\ell_{n+1}} \end{aligned}$$

to a word which contains more than n letters a because the subderivation

$$A_{i_r} \overset{*}{\Longrightarrow} b^{\ell_r-\ell'_r}ab^{\ell_{r+1}} \cdots ab^{\ell'_s}A_{i_s}$$

is carried out twice and produces $s - r$ letters a. This contradiction implies that the language L_n cannot be generated by a right-linear grammar with at most n non-terminal symbols. Hence, we have $L_n \notin RL_n^V$.

Since the language L_n is the suffix-closure of some language, it is suffix-closed.

The language L_n is also ordered because it can be accepted by the following deterministic finite automaton which is ordered.

The automaton is defined as

$$\mathcal{A} = (V, Z, z_0, Z \setminus \{z_{n+2}\}, \delta)$$

with

$$V = \{a, b\},$$
$$Z = \{z_1, z_2, \ldots, z_{n+2}\},$$

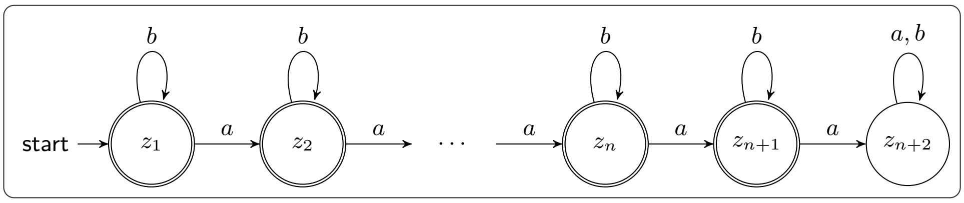
and the transition function δ given by

$$\begin{aligned} \delta(z_i, a) &= z_{i+1} \text{ for } i = 1, 2, \ldots, n+1, \\ \delta(z_{n+2}, a) &= z_{n+2}, \\ \delta(z_i, b) &= z_i \text{ for } i = 1, 2, \ldots, n+2 \end{aligned}$$

which is also shown in the table (where we see that the order $z_1 \prec z_2 \prec \cdots \prec z_{n+2}$ is preserved by the transition mapping)

	z_1	z_2	$\cdots$	z_n	z_{n+1}	z_{n+2}
a	z_2	z_3	$\cdots$	z_{n+1}	z_{n+2}	z_{n+2}
b	z_1	z_2	$\cdots$	z_n	z_{n+1}	z_{n+2}

and which is illustrated in the diagram below (where we see that the automaton accepts exactly the suffixes of a word of the language $(\{b\}^*\{a\})^n\{b\}^*$):



Thus, we have proven that $L_n \in (SUF \cap ORD) \setminus RL_n^V$ holds. $\square$

Lemma 26. *Let*

$$L = \{ab\}.$$

Then $L \in RL_1^P \setminus (COMB \cup SUF \cup CIRC)$.

Proof. The language can be generated by a right-linear grammar with a start symbol S and the rule $S \to ab$ as the only rule.

The language is not combinational because it is neither empty nor infinite. It is not suffix-closed because it does not contain the suffix b of the word $ab \in L$. The language is also not circular because it does not contain the circular shift ba of the word $ab \in L$. $\square$

Lemma 27. *Let*

$$L = \{a\}^+.$$

Then $L \in COMB \setminus RL_1^P$.

Proof. The language is combinational because it has the form V^*V with $V = \{a\}$.

Since a right-linear grammar with one production rule either generates nothing or exactly one word and the language L is infinite, one rule is not sufficient for generating the language L. $\square$

Lemma 28. *Let*

$$L = \{a\}^*.$$

Then $L \in MON \setminus RL_1^P$.

Proof. The language is monoidal because it can be represented as V^* with $V = \{a\}$.

Since a right-linear grammar with one production rule either generates nothing or exactly one word and the language L is infinite, one rule is not sufficient for generating the language L. $\square$

Lemma 29. *Let n be a natural number with $n \geq 2$ and let*

$$V_n = \{a_1, a_2, \ldots, a_n\}$$

be an alphabet with n letters. Then the relations

$$V_n^* \in MON \setminus RL_n^P,$$
$$V_{n+1} \in FIN \setminus RL_n^P,$$
$$V_n^+ \in COMB \setminus RL_n^P$$

hold.

Proof. Let n be a natural number with $n \geq 2$.

The language V_n^* is monoidal, the language V_{n+1} is finite, and the language V_n^+ is combinational.

For generating each of the languages V_n^*, V_{n+1}, and V_n^+, a grammar needs at least, for every letter x of its alphabet, a rule where the first letter of its right-hand side is this letter x. For generating the infinite languages V_n^* and V_n^+, such rules are necessary which do not terminate and at least one terminating rule is needed. Hence, in all three cases, $n+1$ rules are necessary for generating the language. $\square$

Lemma 30. *Let*

$$L = \{ab, bb\}.$$

Then $L \in RL_2^P \setminus (CIRC \cup UF)$.

Proof. The language can be generated by a right-linear grammar with a start symbol S and the rules $S \to ab$ and $S \to bb$ as the only two rules.

The language is not circular because it does not contain the circular shift ba of the word ab. The language is neither union-free because it contains two words of minimal length ([22]). $\square$

We now consider the relations between the families defined by structural properties and those defined by the number of resources.

The state of a deterministic finite automaton with exactly one state is either accepting or not accepting. If it is accepting, the automaton accepts every word over the input alphabet of the automaton, otherwise is does not accept any word. This yields the following equality.

Lemma 31. *It holds* $REG_1^Z = MON \cup \{\emptyset\}$.

From Lemma 31, we obtain the following statements.

Lemma 32. *For the family* REG_1^Z*, the relations*

$$MON \subset REG_1^Z$$

as well as

$$REG_1^Z \subset SUF, \ REG_1^Z \subset NIL, \ REG_1^Z \subset COMM, \ \text{and } REG_1^Z \subset UF$$

hold.

Proof. According to Lemma 31, we have $MON \subset REG_1^Z$.

Every language of the family REG_1^Z is suffix-closed, nil-potent, commutative, and union-free (for the empty set, this is obvious; the other languages are monoidal and, therefore, it follows from the inclusions of the set MON (Theorem 2)). The families SUF, NIL, $COMM$, and UF also contain non-empty finite languages which are not contained in the family REG_1^Z. This proves the properness of each inclusion considered here. □

Lemma 33. *The family* $COMB$ *is incomparable to the family* REG_1^Z *and strictly included in the family* REG_2^Z*. Furthermore,* $COMB \cap REG_1^Z = \{\emptyset\}$.

Proof. According to Theorem 2, the families $COMB$ and SUF are disjoint. Since $MON \subset SUF$, the families $COMB$ and MON are disjoint. The family $COMB$ contains languages which are not empty and not monoidal. Since the family MON is not empty, the families $COMB$ and REG_1^Z are incomparable. Furthermore,

$$COMB \cap REG_1^Z \subseteq \{\emptyset\}.$$

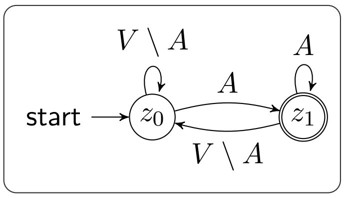
The empty set belongs to the family $COMB$ since it can be given as $\emptyset^*\emptyset$. Hence, the equality holds.

Every combinational language can be represented in the form V^*A for an alphabet V and a subset $A \subseteq V$. Such a language is accepted by the finite automaton

$$\mathcal{A} = (V, \{z_0, z_1\}, z_0, \{z_1\}, \delta)$$

where the transition function δ is given by the following table (which is illustrated in the diagram next to it)

	z_0	z_1
$x \in V \setminus A$	z_0	z_0
$a \in A$	z_1	z_1



which shows that every combinational language can be accepted by a deterministic finite automaton with at most two states. Hence, $COMB \subseteq REG_2^Z$. Since

$$\{\lambda\} \in REG_1^Z \setminus COMB \quad \text{and} \quad REG_1^Z \subset REG_2^Z,$$

we obtain the proper inclusion $COMB \subset REG_2^Z$. □

Lemma 34. *The family FIN is incomparable to each family REG_n^Z for $n \geq 1$.*

Proof. Let n be a natural number with $n \geq 1$ and $V = \{a\}$. Further, let

$$L_n = \{a^{n+1}\}.$$

According to Lemma 19, we have $L_n \in FIN \setminus REG_{n+1}^Z$. Hence, for every natural number $n \geq 1$, there is a finite language which cannot be accepted by a deterministic finite automaton with at most n states.

The language V^*, however, can be accepted with one state but is not finite. □

Besides the Lemma 33, we obtain the following results for the families REG_n^Z for every natural number $n \geq 2$.

Lemma 35. *Each of the families SUF, NIL, DEF, ORD, NC, and PS is incomparable to every family REG_n^Z for $n \geq 2$.*

Proof. The family *NIL* is a subset of each of the families *DEF*, *ORD*, *NC*, and *PS*. The family REG_2^Z is a subset of each of the families REG_n^Z with $n \geq 3$. According to Theorem 2, it suffices to show for the mentioned incomparabilities that, for every number $n \geq 2$, there is a language which is suffix-closed and nilpotent but cannot be accepted by a deterministic finite automaton with n states and that there is a language which is accepted by a deterministic finite automaton with two states but which is not power-separating.

Let n be a natural number with $n \geq 1$ and

$$L_n = \{\, a^i \mid 0 \leq i \leq n \,\}.$$

According to Lemma 20, we have $L_n \in SUF \cap NIL \setminus REG_{n+1}^Z$.

Let

$$L = \{aa\}^*.$$

According to Lemma 21, we have $L \in REG_2^Z \setminus PS$.

Thus, each of the families *SUF*, *NIL*, *DEF*, *ORD*, *NC*, and *PS* is incomparable to each family REG_n^Z for $n \geq 2$. □

Lemma 36. *Each of the families COMM, CIRC, and UF is incomparable to every family REG_n^Z for $n \geq 2$.*

Proof. Since the family *COMM* is a subset of the family *CIRC*, it suffices to show that, for every number $n \geq 2$, there is a commutative and union-free language which is not accepted by a deterministic finite automaton with n states and that there is a language which is accepted by a deterministic finite automaton with two states but which is neither circular nor union-free.

Let n be a natural number with $n \geq 1$ and let

$$L_n = \{a^{n+1}\}.$$

From Lemma 19, we know that $L_n \in (COMM \cap UF) \setminus REG_{n+1}^Z$.

According to Theorem 2, there exists a combinational language which is neither circular nor union-free. By Lemma 33, this language is accepted by a deterministic finite automaton with two states.

Thus, each of the families *COMM*, *CIRC*, and *UF* is incomparable to each of the families REG_n^Z for $n \geq 2$. □

We now consider the sugregular families defined by restricting the number of non-terminals.

Lemma 37. *The proper inclusions*

$$COMB \subset RL_1^V \quad \textit{and} \quad NIL \subset RL_1^V$$

hold.

Proof. Let L be a combinational language. Then $L = V^*A$ for an alphabet V and a subset $A \subseteq V$. This language is generated by the regular grammar

$$G_{COMB} = (\{S\}, V, P, S)$$

where the set P consists of the rules $S \to xS$ for every letter $x \in V$ and $S \to a$ for every letter $a \in A$. Since the grammar contains only one non-terminal, we obtain the inclusion

$$COMB \subseteq RL_1^V.$$

The family RL_1^V contains non-empty finite languages which are not combinational. This proves the properness of the inclusion.

Let L be a nilpotent language over some alphabet V. Then the language L is finite or its complement $V^* \setminus L$ is finite. If the language L is finite, it can be generated by the right-linear grammar

$$G_{NIL} = (\{S\}, V, P, S)$$

where the set P consists of the rules $S \to w$ for every word $w \in L$. If the language L is infinite, then its complement $V^* \setminus L$ is finite. Let

$$m = \max\{\, |w| \mid w \in V^* \setminus L \,\}$$

be the maximal length of the words of the complement set. Then all words with a length of more than m belong to the language L (and possibly further words). Hence, the language L can be represented in the form

$$L = A \cup V^*V^{m+1}$$

for a finite subset $A \subseteq V \leq m$. Any natural number $n \geq m+1$ (any length of a word $w \in L \setminus A$) is the sum $n = k(m+1) + r$ for natural numbers k and r with $k \geq 1$ and $0 \leq r < m+1$. Hence, the number n is also the sum

$$n = (k-1)(m+1) + (m+1) + r$$

and, with $k' = k-1$ and $r' = m+1+r$, we obtain $n = k'(m+1) + r'$ and $k' \geq 0$ and $m+1 \leq r < 2(m+1)$. The language L can be generated by the right-linear grammar

$$G_{NIL\infty} = (\{S\}, V, P, S)$$

where the set P consists of the rules

- $S \to w$ for every word $w \in A$,
- $S \to w$ for every word $w \in V^*$ with $m+1 \leq |w| < 2(m+1)$, and
- $S \to wS$ for every word $w \in V^{m+1}$.

By the rules of the first kind, exactly the words of the set A are generated. By the rules of the second and third kind, exactly the words of the set V^*V^{m+1} are generated. By the rules of the second kind alone, a finite subset of the set V^*V^{m+1} is generated. By the rules of the first and third kind together, an infinite subset of the set V^*V^{m+1} is generated.

Since the grammar $G_{NIL\infty}$ contains only one non-terminal symbol, we obtain the inclusion $NIL \subseteq RL_1^V$.

This proves the properness of the inclusion. □

Lemma 38. *The family DEF is incomparable to the family RL_1^V and strictly included in the family RL_2^V.*

Proof. Let
$$L = \{aa\}^*.$$
According to Lemma 21, we have $L \in RL_1^V \setminus PS$ and, therefore, also $L \in RL_1^V \setminus DEF$.

Let
$$L = \{a\} \cup \{a,b\}^*\{b\} \cup \{\lambda\}.$$
From Lemma 23, we know that $L \in DEF \setminus RL_1^V$.

Thus, the family DEF is incomparable to the family RL_1^V.

Now, let L be an arbitrary definite language $L = A \cup V^*B$ for some alphabet V and two finite subsets A and B of V^*. This language can be generated by the right-linear grammar
$$G = (\{S, S_\infty\}, V, P, S)$$
where the set P consists of the rules
- $S \to w$ for every word $w \in A$,
- $S \to S_\infty$,
- $S_\infty \to xS_\infty$ for every letter $x \in V$, and
- $S_\infty \to w$ for every word $w \in B$.

Hence, every definite language can be generated by a right-linear grammar with two non-terminal symbols.

The properness of the inclusion follows from Lemma 21 with the witness language $L = \{aa\}^*$ mentioned above. □

Lemma 39. *Each of the families SUF, ORD, NC, and PS is incomparable to every family RL_n^V for $n \geq 1$.*

Proof. Due to the inclusions $SUF \subseteq PS$ and $ORD \subseteq NC \subseteq PS$, it suffices to show that for every number $n \geq 1$, there is a suffix-closed and ordered language which is not generated by a right-linear grammar with n non-terminal symbols and that there is a language which is generated by a right-linear grammar with one non-terminal symbol but which is not power-separating.

For the second case, let
$$L = \{aa\}^*.$$
According to Lemma 21, we have $L \in RL_1^V \setminus PS$.

For the first case, let n be a natural number with $n \geq 1$ and let
$$L_n = Suf(\{\, w_1aw_2aw_3 \cdots w_naw_{n+1} \mid w_i \in \{b\}^*, 1 \leq i \leq n+1 \,\}).$$
According to Lemma 25, we have $L_n \in (SUF \cap ORD) \setminus RL_n^V$.

Hence, each of the families SUF, ORD, NC, and PS is incomparable to every family RL_n^V for $n \geq 1$. □

Lemma 40 ([15]). *Let*
$$L = \{a,b,c\}^*\{a,b\}.$$
Then the relation $L \in COMB \setminus (NIL \cup CIRC \cup UF)$ holds.

Proof. The language L can be represented as V^*A for the alphabet $V = \{a, b, c\}$ and its subset $A = \{a, b\}$. Hence, the language is combinational.

This language is infinite. Its complement is also infinite with respect to every alphabet V which is a superset of the alphabet $\{a, b, c\}$ (the complement contains in every case infinitely many words which have c as the last letter). Hence, the language L is not nilpotent.

The word cb belongs to the language L but not its circular shift bc. Hence, the language L is not circular.

In a union-free language, there are no two different words of the minimal length ([22]). But the minimal length of words in the language L is one and the language contains two words of this length (a and b). Hence, the language L is not union-free. □

Lemma 41. *Each of the families COMM, CIRC, and UF is incomparable to every family RL_n^V for $n \geq 1$.*

Proof. Due to the inclusion $COMM \subseteq CIRC$, it suffices to show that for every number $n \geq 1$, there is a commutative and union-free language which cannot be generated by a right-linear grammar with n non-terminal symbols and that there is a language which is generated by a right-linear grammar with one non-terminal symbol but which is neither circular nor union-free.

For the first case, let n be a natural number with $n \geq 1$ and

$$L_n = ((\{b\}^*\{a\})^n\{b\}^*)^*.$$

According to Lemma 24, it holds

$$L_n \in (COMM \cap UF) \setminus RL_n^V.$$

For the second case, let

$$L = \{a, b, c\}^*\{a, b\}.$$

From Lemma 40, we know the relation

$$L \in COMB \setminus (CIRC \cup UF),$$

which implies the relation

$$L \in RL_1^V \setminus (CIRC \cup UF)$$

by Lemma 37. □

We now consider the subregular families defined by restricting the number of production rules.

Lemma 42. *The proper inclusions*

$$RL_1^P \subset FIN \quad \textit{and} \quad RL_1^P \subset UF$$

hold.

Proof. A right-linear grammar with one production rule either generates nothing or exactly one word. The properness of the inclusions hold because there exist a finite language and a union-free language with more than one word. □

Lemma 43. *The family RL_1^P is incomparable to the family COMB. Furthermore, the relation*

$$RL_1^P \cap COMB = \{\emptyset\}$$

holds.

Proof. From Lemma 26, we know

$$\{ab\} \in RL_1^P \setminus COMB.$$

According to Lemma 27, we have

$$\{a\}^+ \in COMB \setminus RL_1^P.$$

These two witness languages prove the incomparability of the family RL_1^P to the family $COMB$.

Since a right-linear grammar with one production rule either generates nothing or exactly one word and all the combinational languages are either empty or infinite, the empty set is the only common language of the two families. □

Lemma 44. *The family RL_1^P is incomparable to the families MON, SUF, COMM, and CIRC.*

Proof. According to Theorem 2, it suffices to show for these incomparabilities that there are a monoidal language which cannot be generated by a right-linear grammar with one production rule only and a language which is generated by a right-linear grammar with one production rule only but which is not suffix-closed nor circular.

According to Lemma 28, we have

$$\{a\}^* \in MON \setminus RL_1^P.$$

From Lemma 26, we know

$$\{ab\} \in RL_1^P \setminus (SUF \cup CIRC).$$

These two witness languages prove the incomparability of the family RL_1^P with each of the families MON, SUF, $COMM$, and $CIRC$. □

Lemma 45. *Each family RL_n^P for $n \geq 2$ is incomparable to each of the families of the set*

$$\mathcal{F} = \{MON, FIN, COMB, NIL, DEF, ORD, NC, PS, SUF, COMM, CIRC, UF\}.$$

Proof. According to Theorem 2, it suffices to show for these incomparabilities that, for any natural number $n \geq 2$, there are a monoidal language, a finite language, and a combinational language which cannot be generated by a right-linear grammar with at most n rules and that there are a language which is not power-separating and a language which is not circular nor union-free but which can be generated by a right-linear grammar with at most two rules.

Let n be a natural number with $n \geq 2$ and let

$$V_n = \{a_1, a_2, \ldots, a_n\}$$

be an alphabet with n letters. The relations

$$\begin{aligned} V_n^* &\in MON \setminus RL_n^P, \\ V_{n+1} &\in FIN \setminus RL_n^P, \\ V_n^+ &\in COMB \setminus RL_n^P \end{aligned}$$

hold according to Lemma 29. Hence, there are a monoidal language, a finite language, and a combinational language which cannot be generated by a right-linear grammar with at most n rules.

From Lemma 21, we know that

$$\{aa\}^* \in RL_2^P \setminus PS.$$

From Lemma 30, we know that

$$\{ab, bb\} \in RL_2^P \setminus (CIRC \cup UF).$$

Hence, there are a language which is not power-separating and a language which is not circular nor union-free but which can be generated by a right-linear grammar with at most two rules.

These witness languages prove the incomparabilities. □

The results of this section are illustrated in Figure 3.

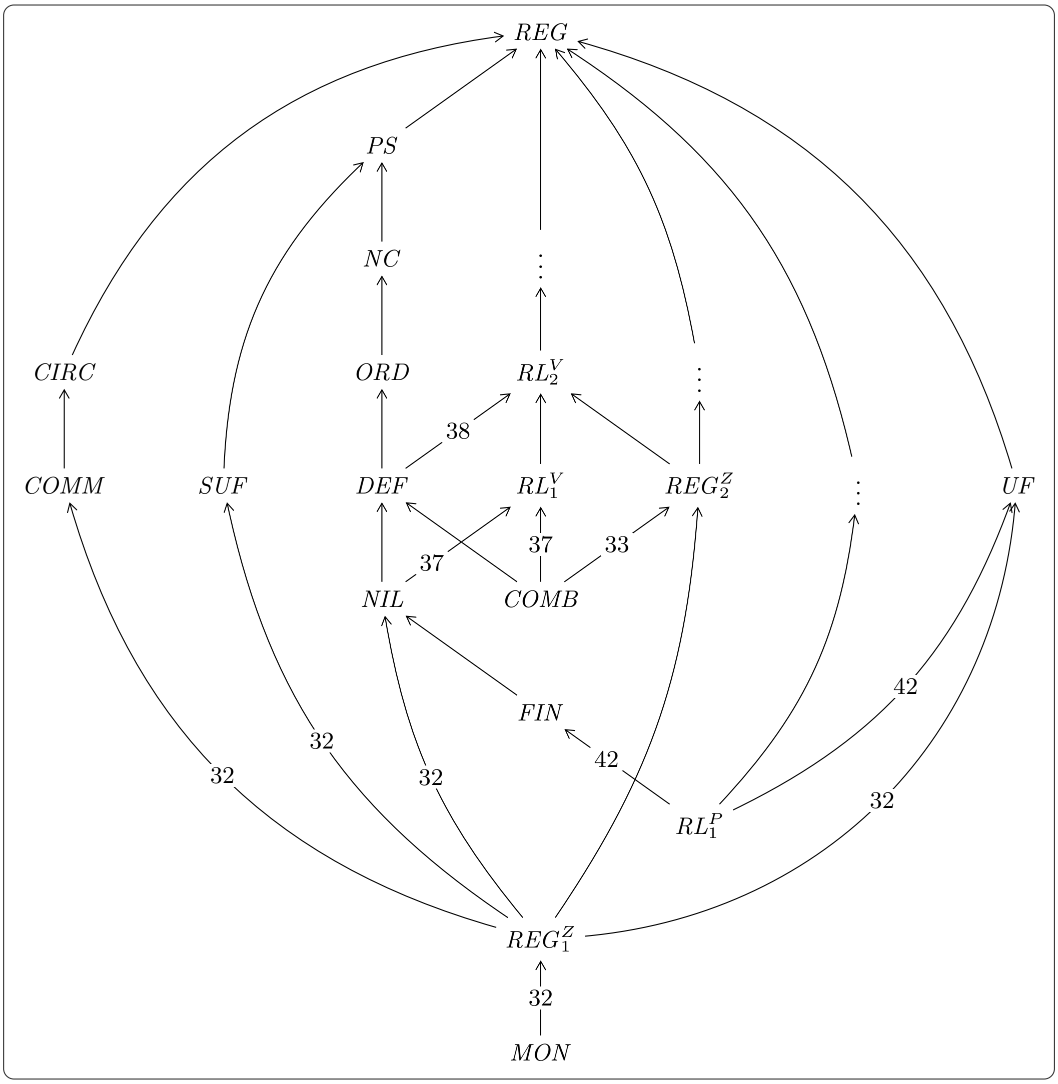


Figure 3: Hierarchy of subregular language families

Theorem 46. *The inclusion relations presented in Figure 3 hold. An arrow from an entry X to an entry Y depicts the proper inclusion $X \subset Y$; if two families are not connected by a directed path, then they are incomparable.*

Proof. The inclusions which correspond to arrows without a label were proven in the previous sections (Theorems 2 and 18). A label at an arrow in Figure 3 refers to the Lemma where the respective inclusion is proven. $\square$

References

1. J. Dassow, F. Manea, B. Truthe, Networks of Evolutionary Processors with Subregular Filters. In: A. H. Dediu, S. Inenaga, C. Martín-Vide (eds.), *Language and Automata Theory and Applications. Fifth International Conference, LATA 2011, Tarragona, Spain, May 26 – 31, 2011*. LNCS 6638, Springer-Verlag, 2011, 262–273.
2. J. Dassow, F. Manea, B. Truthe, On Contextual Grammars with Subregular Selection Languages. In: M. Holzer, M. Kutrib, G. Pighizzini (eds.), *Descriptional Complexity of Formal Systems – 13th International Workshop, DCFS 2011, Gießen/Limburg, Germany, July 25 – 27, 2011. Proceedings*. LNCS 6808, Springer-Verlag, 2011, 135–146.
3. J. Dassow, F. Manea, B. Truthe, On External Contextual Grammars with Subregular Selection Languages. *Theoretical Computer Science* **449** (2012) 1, 64–73.
4. J. Dassow, F. Manea, B. Truthe, On Subregular Selection Languages in Internal Contextual Grammars. *Journal of Automata, Languages, and Combinatorics* **17** (2012) 2–4, 145–164.
5. J. Dassow, F. Manea, B. Truthe, Networks of Evolutionary Processors: The Power of Subregular Filters. *Acta Informatica* **50** (2013) 1, 41–75.
http://link.springer.com/article/10.1007/s00236-012-0172-0
6. J. Dassow, F. Manea, B. Truthe, On the Power of Accepting Networks of Evolutionary Processors with Special Topologies and Random Context Filters. *Fundamenta Informaticae* **136** (2015) 1–2, 1–35.
http://content.iospress.com/articles/fundamenta-informaticae/fi136-1-2-02
7. J. Dassow, R. Stiebe, B. Truthe, Two Collapsing Hierarchies of Subregularly Tree Controlled Languages. *Theoretical Computer Science* **410** (2009) 35, 3261–3271.
http://dx.doi.org/10.1016/j.tcs.2009.03.005
8. J. Dassow, R. Stiebe, B. Truthe, Generative Capacity of Subregularly Tree Controlled Grammars. *International Journal of Foundations of Computer Science* **21** (2010), 723–740.
9. J. Dassow, B. Truthe, On Two Hierarchies of Subregularly Tree Controlled Languages. In: C. Câmpeanu, G. Pighizzini (eds.), *10th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2008, Charlottetown, Prince Edward Island, Canada, July 16–18, 2008, Proceedings*. University of Prince Edward Island, 2008, 145–156.
http://theo.cs.uni-magdeburg.de/pubs/papers/dastru-subreg-dcfs08.pdf
10. J. Dassow, B. Truthe, Subregularly Tree Controlled Grammars and Languages. In: E. Csuhaj-Varjú, Z. Ésik (eds.), *Automata and Formal Languages – 12th International Conference AFL 2008, Balatonfüred, Hungary, May 27–30, 2008, Proceedings*. Computer and Automation Research Institute of the Hungarian Academy of Sciences, 2008, 158–169.
11. J. Dassow, B. Truthe, On Networks of Evolutionary Processors with State Limited Filters. In: H. Bordihn, R. Freund, T. Hinze, M. Holzer, M. Kutrib, F. Otto (eds.), *Second Workshop on Non-Classical Models of Automata and Applications (NCMA), Jena, Germany, August 23–24, 2010, Proceedings*. books@ocg.at 263, Österreichische Computer Gesellschaft, Austria, 2010, 57–70.
12. J. Dassow, B. Truthe, On Networks of Evolutionary Processors with Filters Accepted by Two-State-Automata. *Fundamenta Informaticae* **112** (2011) 2–3, 157–170.
13. J. Dassow, B. Truthe, Extended Splicing Systems with Subregular Sets of Splicing Rules. In: R. Freund, M. Holzer, B. Truthe, U. Ultes-Nitsche (eds.), *Fourth Workshop on Non-Classical Models of Automata and Applications (NCMA), Fribourg, Switzerland, August 23–24, 2012, Proceedings*. books@ocg.at 290, Österreichische Computer Gesellschaft, Austria, 2012, 65–78.
14. I. M. Havel, The theory of regular events ii. *Kybernetika* **5** (1969) 6, 520–544.
15. M. Holzer, B. Truthe, On Relations Between Some Subregular Language Families. In: R. Freund, M. Holzer, N. Moreira, R. Reis (eds.), *Seventh Workshop on Non-Classical Models of Automata and Applications (NCMA), Porto, Portugal, August 31 – September 1, 2015, Proceedings*. books@ocg.at 318, Österreichische Computer Gesellschaft, Austria, 2015, 109–124.

16. J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, 1979.
17. R. Laing, J. Wright, *Commutative machines*. Technical Report 04422/0310525-T, University of Michigan, Ann Arbor, Michigan, 1962.
18. F. Manea, B. Truthe, Accepting Networks of Evolutionary Processors with Subregular Filters. In: P. Dömösi, S. Iván (eds.), *Automata and Formal Languages – 13th International Conference AFL 2011, Debrecen, Hungary, August 17–22, 2011, Proceedings*. College of Nyíregyháza, 2011, 300–314.
19. F. Manea, B. Truthe, On Internal Contextual Grammars with Subregular Selection Languages. In: M. Kutrib, N. Moreira, R. Reis (eds.), *Descriptional Complexity of Formal Systems – 14th International Workshop, DCFS 2012, Braga, Portugal, July 23 – 25, 2012. Proceedings*. LNCS 7386, Springer-Verlag, 2012, 222–235.
20. F. Manea, B. Truthe, Accepting Networks of Evolutionary Processors with Subregular Filters. *Theory of Computing Systems* **55** (2014) 1, 84–109.
http://alerts.springer.com/re?l=D0In5r6cdI6h4nom7Iy
21. R. McNaughton, S. Papert, *Counter-Free Automata*. The M.I.T. Press, Cambridge, Massachusetts, 1971.
22. B. Nagy, A Normal Form for Regular Expressions. In: C. S. Calude, E. Calude, M. J. Dinneen (eds.), *Supplemental Papers for DLT'04*. CDMTCS Research Report Series 252, University of Auckland, New Zealand, 2004, 53–62.
23. M. Peres, M. Rabin, E. Shamir, The theory of definite automata. *IEEE Transactions on Electronic Computers* **12** (1963) 3, 233–243.
24. H. Shyr, G. Thierrin, Ordered Automata and Associated Languages. *Tankang Journal of Mathematics* **5** (1974) 1, 9–20.
25. H. Shyr, G. Thierrin, Power-Separating Regular Languages. *Mathematical Systems Theory* **8** (1974) 1, 90–95.
26. B. Truthe, A Relation Between Definite and Ordered Finite Automata. In: S. Bensch, R. Freund, F. Otto (eds.), *Sixth Workshop on Non-Classical Models of Automata and Applications (NCMA), Kassel, Germany, July 28–29, 2014, Proceedings*. books@ocg.at 304, Österreichische Computer Gesellschaft, Austria, 2014, 235–247.
27. B. Wiedemann, *Vergleich der Leistungsfähigkeit endlicher determinierter Automaten*. Diplomarbeit, Universität Rostock, 1978.

Institut für Informatik

Justus-Liebig-Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany

Recent Reports

(Further reports are available at `www.informatik.uni-giessen.de`.)

M. Holzer, M. Hospodár, *On the Magic Number Problem of the Cut Operation*, Report 1703, October 2017.

M. Holzer, S. Jakobi, *A Note on the Computational Complexity of Some Problems for Self-Verifying Finite Automata*, Report 1702, April 2017.

S. Beier, M. Holzer, M. Kutrib, *On the Descriptional Complexity of Operations on Semilinear Sets*, Report 1701, April 2017.

M. Holzer, S. Jakobi, M. Wendlandt, *On the Computational Complexity of Partial Word Automata Problems*, Report 1404, May 2014.

H. Gruber, M. Holzer, *Regular Expressions From Deterministic Finite Automata, Revisited*, Report 1403, May 2014.

M. Kutrib, A. Malcher, M. Wendlandt, *Deterministic Set Automata*, Report 1402, April 2014.

M. Holzer, S. Jakobi, *Minimal and Hyper-Minimal Biautomata*, Report 1401, March 2014.

J. Kari, M. Kutrib, A. Malcher (Eds.), *19th International Workshop on Cellular Automata and Discrete Complex Systems AUTOMATA 2013 Exploratory Papers*, Report 1302, September 2013.

M. Holzer, S. Jakobi, *Minimization, Characterizations, and Nondeterminism for Biautomata*, Report 1301, April 2013.

A. Malcher, K. Meckel, C. Mereghetti, B. Palano, *Descriptional Complexity of Pushdown Store Languages*, Report 1203, May 2012.

M. Holzer, S. Jakobi, *On the Complexity of Rolling Block and Alice Mazes*, Report 1202, March 2012.

M. Holzer, S. Jakobi, *Grid Graphs with Diagonal Edges and the Complexity of Xmas Mazes*, Report 1201, January 2012.

H. Gruber, S. Gulan, *Simplifying Regular Expressions: A Quantitative Perspective*, Report 0904, August 2009.

M. Kutrib, A. Malcher, *Cellular Automata with Sparse Communication*, Report 0903, May 2009.

M. Holzer, A. Maletti, *An $n \log n$ Algorithm for Hyper-Minimizing States in a (Minimized) Deterministic Automaton*, Report 0902, April 2009.

H. Gruber, M. Holzer, *Tight Bounds on the Descriptional Complexity of Regular Expressions*, Report 0901, February 2009.

M. Holzer, M. Kutrib, and A. Malcher (Eds.), *18. Theorietag Automaten und Formale Sprachen*, Report 0801, September 2008.

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: Nondeterminism is Better than Determinism*, Report 0301, February 2003

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: $k+1$ Pushdown Reversals are Better Than k*, Report 0206, November 2002

M. Holzer, M. Kutrib, *Nondeterministic Descriptional Complexity of Regular Languages*, Report 0205, September 2002

H. Bordihn, M. Holzer, M. Kutrib, *Economy of Description for Basic Constructions on Rational Transductions*, Report 0204, July 2002

M. Kutrib, J.-T. Löwe, *String Transformation for n-dimensional Image Compression*, Report 0203, May 2002

A. Klein, M. Kutrib, *Grammars with Scattered Nonterminals*, Report 0202, February 2002

A. Klein, M. Kutrib, *Self-Assembling Finite Automata*, Report 0201, January 2002

M. Holzer, M. Kutrib, *Unary Language Operations and its Nondeterministic State Complexity*, Report 0107, November 2001

A. Klein, M. Kutrib, *Fast One-Way Cellular Automata*, Report 0106, September 2001

M. Holzer, M. Kutrib, *Improving Raster Image Run-Length Encoding Using Data Order*, Report 0105, July 2001

M. Kutrib, *Refining Nondeterminism Below Linear-Time*, Report 0104, June 2001

M. Holzer, M. Kutrib, *State Complexity of Basic Operations on Nondeterministic Finite Automata*, Report 0103, April 2001