Closure Properties and Descriptional Complexity of Deterministic Regular Expressions[☆]

Katja Losemann[a,1,*], Wim Martens[a], Matthias Niewerth[a,1]

[a] *Universität Bayreuth*

Abstract

We study the descriptional complexity of regular languages that are definable by deterministic regular expressions, i.e., we examine worst-case blow-ups in size when translating between different representations for such languages. As representations of languages, we consider regular expressions, deterministic regular expressions, and deterministic finite automata. Our results show that exponential blow-ups between these representations cannot be avoided. Furthermore, we study the descriptional complexity of these representations when applying boolean operations. Here, we start by investigating the closure properties of such languages under various language-theoretic operations such as union, intersection, concatenation, Kleene star, and reversal. Our results show that languages that are definable by deterministic regular expressions are not closed under any of these operations. Finally, we show that for all these operations except the Kleene star an exponential blow-up in the size of deterministic regular expressions cannot be avoided.

Keywords: deterministic regular expressions, one-unambiguous languages, boolean operations, automata theory, descriptional complexity

1. Introduction

Deterministic or *one-unambiguous* regular expressions (henceforth, DREs) have been a topic of research since they were formally defined by Brüggemann-Klein and Wood [2]. Their origins lie in the ISO standard for the Standard Generalized Markup Language (SGML) where they were introduced to ensure efficient parsing. Today, the prevalent schema languages for XML data, such as Document Type Definition (DTD) and XML Schema, require that the regular expressions in their specification are deterministic. From a more foundational point of view, one-unambiguity is a natural manner in which to define determinism in regular expressions. As such, several decision problems behave better

[☆] This article is the extended version of [25].

[*] Corresponding author

[1] Supported by grant number MA 4938/2–1 of the Deutsche Forschungsgemeinschaft (Emmy Noether Nachwuchsgruppe).

for deterministic regular expressions than for general ones. For example, language inclusion for regular expressions is PSPACE-complete but is tractable when the expressions are deterministic. Unfortunately, not every regular language can be expressed by a deterministic expression, i.e., not every regular language is DRE-definable. The canonical example for such a regular language is $L((a+b)^*a(a+b))$, see [2].

Although DRE-definable languages are rather widespread and have been around for quite some time, they are not yet well-understood. This motivates us to study their foundational properties. In particular, we investigate the differences in the descriptional complexity between regular expressions (REs), deterministic regular expressions (DREs), and deterministic finite automata (DFAs). Our initial motivation for this work was an unproved claim in [2] which states that, for expressions of the form Σ^*w, where w is a word over alphabet Σ, every equivalent DRE is at least exponentially larger than the length of w. We proved that this claim is indeed true in the conference version of this work [25], but the proof turned out to be rather non-trivial. The main challenge was that languages of the form Σ^*w have polynomial-size REs and DFAs, so one has to develop new techniques for proving lower bounds on the size of DREs. In this article (Section 3), we give two different proofs showing the unavoidable exponential blow-up when translating an RE to a DRE. The first one proves that it cannot be avoided even for finite languages. The latter uses a more general technique which gives more insights in the structure of DRE-definable languages and their DREs.

Another set of contributions in this paper is a study of the effect of language-theoretic operations on languages that are definable by a DRE. In particular, we consider union, intersection, difference, concatenation, Kleene star, and reversal, for unary and arbitrary alphabets. Several of these operations are relevant in XML schema management [9, 29]. We provide a complete overview of the closure properties of DRE-definable languages under these operations in Section 4. Afterwards, in Section 5, we briefly investigate the *state complexity* of minimal DFAs for DRE-definable languages. Here, *state complexity* refers to the number of states of the minimal DFA without the sink state. The main reason why we briefly consider state complexity is because we want to provide results that are directly comparable with the results on state complexity in [16, 30, 33]. That is, the first part of Section 5 lists the increase in state complexity when performing operations on DFAs for DRE-definable languages, if the result of the operation is also DRE-definable. In the second part of Section 5, we study a similar question for DREs. That is, what is the descriptional complexity of DREs that are obtained by performing the aforementioned operations on DREs? Here, we show that for all these operations except the Kleene star an exponential blow-up cannot be avoided when applying the operation on two DREs.

Related Work. Deterministic regular expressions have recently been investigated from several perspectives [6, 12, 26, 27]. Groz and Maneth proved that the membership problem (is a given word in the language of a given DRE?) can be solved in time $O(m + n \log\log m)$, where n is the size of the word and m the

size of the expression [12]. The *DRE-definability* problem asks whether a given regular expression or non-deterministic automaton defines a language that can be expressed by a DRE and was recently proved to be PSPACE-complete [6, 26].

Deterministic regular expressions *with counters* are also a topic of investigation [5, 11, 17, 20, 21], since these expressions are the ones used to define content models in XML Schema. In fact, determinism for regular expressions with counters can be defined in different ways (weak determinism and strong determinism) [11]. While the expressiveness of strongly deterministic expressions with counting is the same as DREs, the weakly deterministic expressions, which are the ones used in XML Schema, are more expressive [11]. However, weakly deterministic regular expressions with counting still cannot define all regular languages [11]. It was recently shown that it can be decided if the language of a given finite automaton is expressible by a weakly deterministic regular expression with counting [23].

In this article we focus on *descriptional complexity* of DREs. Research on descriptional complexity of regular languages focused mainly on REs and DFAs. It is well-known that an exponential blow-up cannot be avoided when translating an RE into a DFA [16]. Ehrenfeucht and Zeiger [7] proved that there also exist DFAs which are exponentially more succinct than each equivalent RE. Gruber and Holzer [13, 15] showed that there exist certain characteristics of automata which make equivalent regular expressions large. However, these characteristics cannot naïvely be transferred to DREs. For example, the languages used in the literature for proving lower bounds on the size of REs (e.g., [7, 13, 15]) are not definable by DREs.

The state complexity of boolean operations on DFAs is studied in [22, 28, 30, 32, 33], where in [30] the focus is on unary languages. In Section 5.1 we see that many results in [33] directly apply for DRE-definable languages since they concern finite languages and every finite language is DRE-definable [1].

Gelade and Neven [10] and Gruber and Holzer [14] independently examined the descriptional complexity of complementation and intersection for REs. They showed that the size of the smallest RE for the intersection of a fixed number of REs can be exponential; and that the size of the smallest RE for the complement of an RE can be double-exponential. Furthermore, these bounds are tight. Gelade and Neven also investigate these operations on DREs and proved that the exponential bound on intersection is also tight when the input is given as DREs instead of REs [10]. Moreover, they proved that the complement of a DRE can always be described by a polynomial-size RE. However, in their proofs, the languages of the resulting REs are not DRE-definable. Concatenation and reversal operations on regular languages are studied in [3, 18, 19, 31, 34], where in [34] also languages over unary alphabets are examined.

2. Definitions

By Σ we always denote a finite alphabet of symbols. A (Σ-)*word* w over alphabet Σ is a finite sequence of symbols $a_1 \cdots a_n$, where $a_i \in \Sigma$ for each $i = 1, \ldots, n$. The set of all Σ-words is denoted by Σ^*. The *length* of a word

$w = a_1 \cdots a_n$ is n and is denoted by $|w|$. The empty word is denoted by ε. A *(word) language* L is a set of words. For two languages L_1 and L_2, we define the concatenation $L_1 \cdot L_2$ as the set $\{vw \mid v \in L_1 \wedge w \in L_2\}$. By L^i with $i \in \mathbb{N}$ we denote the concatenation $L \cdots L$ of i-times the language L.

A *(deterministic, finite) automaton* (or *DFA*) A is a tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, the transition function $\delta : Q \times \Sigma \nrightarrow Q$ is a partial function, q_0 is the initial state, and $F \subseteq Q$ is the set of accepting states. We say that the aforementioned transition is q_1*-outgoing*, q_2*-incoming*, or a*-labeled*. The *run of* A *on word* $w = a_1 \cdots a_n$ is a sequence $q_0 \cdots q_n$ where, for each $i = 1, \ldots, n$, $\delta(q_{i-1}, a_i) = q_i$. The word w is *accepted* by A if the run is *accepting*, i.e., if $q_n \in F$. By $L(A)$ we denote the *language of* A, i.e., the set of words accepted by A. By δ^* we denote the extension of δ to words, i.e., $\delta^*(q, w)$ is the state which is reached from q by reading w. In this paper we assume that all states of an automaton are *useful*, that is, every state can appear in some accepting run. We define the size $|A|$ of a DFA A as $|\{(q, a) \mid \delta(q, a) \text{ is defined}\}|$.

The set of *regular expressions (RE)* over Σ is defined as follows: $\emptyset$, ε and every Σ-symbol is a regular expression; and whenever r and s are regular expressions then so are $(r \cdot s)$, $(r + s)$, and $(s)^*$. W.l.o.g. we can assume that $\emptyset$ does not occur as a (strict) sub-expression of a regular expression. We refer to Σ-symbols, ε, and $\emptyset$ as *atomic* expressions. For readability, we usually omit concatenation operators and parentheses in examples. For a regular expression r, the *language* $L(r)$ is inductively defined as follows: $L(\varepsilon) = \{\varepsilon\}$, $L(\emptyset) = \{\emptyset\}$, $L(\sigma) = \{\sigma\}$ for every $\sigma \in \Sigma$, $L(r \cdot s) = L(r) \cdot L(s)$, $L(r + s) = L(r) \cup L(s)$, and $L(r^*) = \{\varepsilon\} \cup (\cup_{i \in \mathbb{N}} L(r)^i)$.

Whenever we say that expressions or automata are *equivalent*, we mean that they define the same language. The *size* $|r|$ of r is defined to be the total number of occurrences of alphabet symbols, epsilons, and operators, i.e., the number of nodes in its parse tree (including the leaf nodes). A regular expression r is *minimal* if for every regular expression r' with $L(r') = L(r)$, we have $|r| \leq |r'|$.

In order to improve readability, we sometimes use an abbreviated notation for expressions. For $k, \ell \in \mathbb{N}$ we write $r^{k,\ell}$ to denote $rr \cdots r(r+\varepsilon)(r+\varepsilon) \cdots (r+\varepsilon)$, the concatenation of k times r with $\ell - k$ times $(r + \varepsilon)$. Since this is just an abbreviated notation to denote a larger expression consisting of ℓ occurrences of r, $\ell - 1$ additional concatenations, $(\ell - k)$ additional disjunctions, and $(\ell - k)$ additional occurrences of ε, the size of $r^{k,\ell}$ is therefore $\ell|r| + \ell - 1 + 2(\ell - k)$.

Let L be a language. By *first(L)* we denote the set of all symbols $a \in \Sigma$ for which there is a word $aw \in L$. For a regular expression r, we define first(r) as first$(L(r))$. Similarly to first(L), we define *last(L)* as the set of all symbols a, such that $wa \in L$. The set *followlast(L)* contains all symbols a, such that there exists words $v, w \in \Sigma^*$ with $v \in L$ and $vaw \in L$. We use the definition for regular expressions analogously. The *Brzozowski derivative* $w^{-1}L$ of a regular language L and a word $w \in \Sigma^*$ is defined as the language $\{v \in \Sigma^* \mid wv \in L\}$. For a regular language L and words $v, w \in \Sigma^*$, we say that v and w are in the same *Nerode equivalence class* C if and only if, for all words $z \in \Sigma^*$, it holds that

$$v \cdot z \in L \Leftrightarrow w \cdot z \in L.$$

Deterministic regular expressions are defined as follows. Let r be a regular expression over an alphabet Σ. The expression $\bar{r}$ over the alphabet $\bar{\Sigma} = \cup_{\sigma\in\Sigma}\{\sigma_1, \ldots, \sigma_{|r|}\}$ is obtained from r by replacing, for every i and a, the i-th occurrence of alphabet symbol a in r (counting from left to right) by a_i. For example, for $r = b^*a(b^*a)^*$ we have $\bar{r} = b_1^*a_1(b_2^*a_2)^*$. A regular expression r is *deterministic* (a *DRE* or *one-unambiguous* [2]) if there are no words wa_iv and wa_jv' in $L(\bar{r})$ such that $i \neq j$. The expression $(a + b)^*a$ is not deterministic since both words a_2 and a_1a_2 are in $L((a_1 + b_1)^*a_2)$. The equivalent expression $b^*a(b^*a)^*$ is deterministic. Brüggemann-Klein and Wood showed that not every regular expression is equivalent to a deterministic one [2]. We call a regular language *DRE-definable* if there exists a DRE that defines it. The canonical example for a language that is not DRE-definable is $L((a+b)^*a(a+b))$. We define *minimal DREs* similar as for REs. We note that minimal DREs are not unique (up to reordering of disjunctions). For example, the deterministic expressions $(a + \varepsilon)(c + d) + b(c + \varepsilon) + \varepsilon$ and $a(c + d) + (b + \varepsilon)(c + \varepsilon) + d$ are equivalent and both minimal.

Next, we briefly review a result from Brüggemann-Klein and Wood to characterize when a regular language is DRE-definable [2]. They designed a decision algorithm which outputs, given a minimal DFA A, whether $L(A)$ is DRE-definable. We review some of the results on which this algorithm is based and that are useful to us in the remainder of the article. The terminology comes from [2]. For a state q in an NFA A, the *orbit of* q, denoted $\mathcal{O}(q)$, is the (maximal) strongly connected component of A that contains q. We call q a *gate* of $\mathcal{O}(q)$ if q is accepting, or q has an outgoing transition that leaves $\mathcal{O}(q)$. The *orbit automaton of a state* q is the sub-automaton of A consisting of the orbit of q in which the initial state is q and the accepting states are the gates of $\mathcal{O}(q)$. We denote the orbit automaton of q by A_q. The *orbit language of* q is $L(A_q)$. The *orbit languages of* A are all orbit languages of q for all states q of A. Automaton A with transition function δ has the *orbit property* if, for every pair of gates q_1, q_2 in the same orbit, the following properties hold:

1. q_1 is accepting if and only if q_2 is accepting; and,
2. for all states q outside the orbit of q_1 and q_2, it holds $\delta(q_1, a) = q$ if and only if $\delta(q_2, a) = q$.

Then the following is a characterization of DRE-definable regular languages.

Theorem 1 (Brüggemann-Klein and Wood [2]). *Let A be a minimal DFA. Then, $L(A)$ is DRE-definable if and only if A has the orbit property and all orbit languages of A are DRE-definable.*

Furthermore, we need the notion of *A-consistent* symbols. A symbol $a \in \Sigma$ is *A-consistent* if there is a state $f(a)$, such that $\delta(q, a) = f(a)$ for every accepting state q of A. A set $S \subseteq \Sigma$ is *A-consistent* if every $a \in S$ is A-consistent. By A_S we denote the *S-cut* of A which is constructed from A by removing all transitions $\delta(q, a) = f(a)$ for every accepting state q and symbol $a \in S$.

Finite Languages					Infinite Languages				
RE	DRE	DFA	Case exists?	Ref	RE	DRE	DFA	Case exists?	Ref
$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	yes	Obs.3	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	yes	Obs.3
$\Theta(n)$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	yes	[2, 28]	$\Theta(n)$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	yes	Cor.9
$2^{\Omega(n)}$	$2^{\Omega(n)}$	$\Theta(n)$	no	[8]	$2^{\Omega(n)}$	$2^{\Omega(n)}$	$\Theta(n)$	?	
$\Theta(n)$	$2^{\Omega(n)}$	$\Theta(n)$	yes	Th.7	$\Theta(n)$	$2^{\Omega(n)}$	$\Theta(n)$	yes	Cor.9, The.17
$n^{\Theta(\log n)}$	$n^{\Theta(\log n)}$	$\Theta(n)$	yes	[15]					

Table 1: Descriptional complexity of DRE-definable languages.

Using the definition of S-cuts, Brüggemann-Klein and Wood provide another characterization of DRE-definable languages.

Theorem 2 (Brüggemann-Klein and Wood [2]). *Let A be a minimal DFA and S be a set of A-consistent symbols of A. Then, $L(A)$ is DRE-definable if and only if*

1. *A_S has the orbit property; and*
2. *all orbit languages of A_S are DRE-definable.*

3. Descriptional Complexity of DFAs, REs, and DREs

We consider the relative descriptional complexity of REs, DREs and DFAs. Here, the *descriptional complexity* of a language L w.r.t. a model $\mathcal{M}$ (where $\mathcal{M}$ is either the set of DFAs, the set of REs, or the set of DREs) is the smallest element e in $\mathcal{M}$ such that $L(e) = L$. An overview of our results is shown in Table 1. Since every DRE is an RE, we know that every minimal RE for a language L is smaller or equal to a minimal DRE for L. Furthermore, Brüggemann-Klein and Wood showed that, given a DRE r, one can construct a DFA A for $L(r)$ with size $O(|\Sigma||r|)$. Notice that, in general, it is possible that there is an exponential blow-up when translating an RE to a DFA and another exponential blow-up when translating the DFA into a DRE.

We start with a trivial observation that shows that there are languages that do not cause any significant blow-up between the different representations. For example, consider the language $L(a^n)$ and the infinite language $L((a^n)^*)$ for an arbitrary natural number n.

Observation 3. *There exists a class of finite languages $(L_n)_{n\in\mathbb{N}}$ and a class of infinite languages $(L'_n)_{n\in\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, the minimal DFAs, minimal REs, and minimal DREs for L_n and L'_n have size $\Theta(n)$.*

3.1. DREs for Finite Languages

We present an overview of what is known in the case of finite languages. First, notice that every finite language is DRE-definable (see, e.g., [1]). Mandl showed that, for each language $L((0+1)^{0,n}1(0+1)^n)$ with $n \in \mathbb{N}$, every DFA has size exponential in n. It was shown by Brüggemann-Klein and Wood that also every DRE for the language is of size exponential in n.

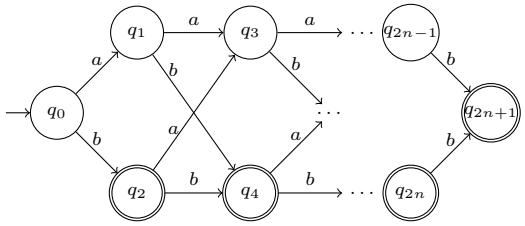


Figure 1: Minimal DFA for language L_n.

Theorem 4 ([2, 28]). *For each $n \in \mathbb{N}$, the minimal DFA (and, therefore, every minimal DRE) for the language $L((a+b)^{0,n}a(a+b)^n)$ has size $2^{\Omega(n)}$.*

Ellul et al. [8] showed that, for each DFA (or even non-deterministic automaton) A of size n that defines a finite language $L(A)$, there exists an RE for $L(A)$ of size $n^{O(\log n)}$. Gruber and Johannsen [15] proved that this bound is also tight.

Theorem 5 ([8]). *Let A be a DFA of size n and let $L(A)$ be finite. Then there exists an RE r for $L(A)$ such that $|r| \leq n^{O(\log n)}$.*

Theorem 6 ([15]). *There exists a family of finite languages $(L_n)_{n\in\mathbb{N}}$ such that the minimal DFA for L_n has $\Theta(n)$ states but every minimal RE for L_n has size $n^{\Theta(\log n)}$.*

In the following we prove that there exists a class of finite languages $(L_n)_{n\in\mathbb{N}}$ such that every minimal RE and the minimal DFA for L_n are exponentially more succinct than every minimal DRE for L_n.

Theorem 7. *There exists a family of finite languages $(L_n)_{n\in\mathbb{N}}$ such that every minimal RE for L_n has size $\Theta(n)$, the minimal DFA for L_n has size $\Theta(n)$, and every minimal DRE for L_n has size $2^{\Omega(n)}$.*

Proof. To prove the assumption we consider the family $(L_n)_{n\in\mathbb{N}}$ where

$$L_n = L((a+b)^{0,n} \cdot b), \qquad \text{for every } n \in \mathbb{N}.$$

For every n, the regular expression $(a+b)^{0,n} \cdot b$ is equivalent to the regular expression $(a+b+\varepsilon)\cdots(a+b+\varepsilon)\cdot b$ where the subexpression $(a+b+\varepsilon)$ appears n times. Observe that the latter expression has size $6n+1$. Moreover, every regular expression equivalent to L_n has to be at least of size $n+1$ since L_n is finite and contains a word of length $n+1$. The minimal DFA for L_n is shown in Figure 1 and has size $4n$.

Let r_n be a minimal DRE for L_n. We show by induction on n that r_n has at least size 2^n. For the base case, $n = 0$, the assumption holds because $L_0 = L(b)$ and $|r_0| \geq 1$.

For the induction case, assume that r_{n-1} has at least size 2^{n-1}. We will now prove that

$$r_n = a \cdot r_{n-1} + b \cdot (\varepsilon + r_{n-1}),$$

which implies that the size of r_n is at least twice the size of r_{n-1} and which would conclude our proof.

Towards a contradiction, assume that r_n has a concatenation operation as topmost operation in its syntax tree, i.e., $r_n = s_1 \cdot s_2$ for some DREs s_1 and s_2. Then, we distinguish two cases depending on whether $\varepsilon \in L(s_1)$ or not.

If $\varepsilon \notin L(s_1)$, then $\text{first}(s_1) = \{a, b\}$. Since $b \in L_n$, it follows that $\varepsilon \in L(s_2)$ and, thus, that every word in $L(s_1)$ ends with b by the structure of L_n. Moreover, we know that $s_2 \neq \varepsilon$ because r_n is minimal. Let ub be a longest word in $L(s_1)$ and vb be a longest word in $L(s_2)$ such that $ubvb \in L_n$. By the structure of L_n it follows that also $uavb \in L_n$. Since ua and vb are of maximal length for s_1 and s_2, respectively, we have that $ua \in L(s_1)$. It follows that also $ua \in L(r_n)$ (because $\varepsilon \in L(s_2)$), which contradicts the assumption that $L(r_n) = L_n$.

If $\varepsilon \in L(s_1)$, then it holds that $\varepsilon \notin L(s_2)$. Since $b \in L(r_n)$ and r_n is a DRE it follows that $\text{first}(s_1) = \{a\}$ and $\text{first}(s_2) = \{b\}$. (Indeed, $\text{first}(s_1) \cap \text{first}(s_2) = \emptyset$ because r_n is a DRE.) Let bw be a longest word in $L(r_n)$, then we know that $bw \in L(s_2)$. Since $s_1 \neq \varepsilon$ (because r_n is minimal) bw is not a longest word in L_n, which contradicts our assumption. This proves that r_n is not a concatenation.

Furthermore, r_n cannot be a Kleene star because $\varepsilon \notin L_n$. Therefore, we know that r_n has to be a disjunction.

Since r_n is a DRE and $\varepsilon \notin L_n$, the expression r_n has to be of the form $a \cdot s_1 + b \cdot s_2$ for some DREs s_1 and s_2. Moreover, we have that $a^{-1}L_n = L_{n-1}$, which means that $L(s_1) = L(r_{n-1})$. Since s_1 and r_{n-1} are defined as minimal DREs, we also know that $|s_1| = |r_{n-1}|$. Therefore, we assume that $s_1 = r_{n-1}$ for the sake of readability in the following. This proves that $r_n = a \cdot r_{n-1} + b \cdot s_2$.

It remains to show that every minimal DRE for the language $L(s_2) = b^{-1}L_n = L(r_{n-1} + \varepsilon) = L_{n-1} \cup \{\varepsilon\}$ is of the form $r^{\varepsilon}_{n-1} = r_{n-1} + \varepsilon$ with $L(r_{n-1}) = L_{n-1}$.

Let r^{ε}_m be a minimal DRE for $L^{\varepsilon}_m = L_{m-1} \cup \{\varepsilon\}$ and $m \in \mathbb{N}$. We show by induction on m that r^{ε}_m is of the form $r_m + \varepsilon$ where $L(r_m) = L_m$. For the base case, $m = 0$, the assumption holds because $L^{\varepsilon}_0 = L(b + \varepsilon)$, the minimal DRE for L^{ε}_0 is $b + \varepsilon$, and $L_0 = L(b)$.

For the induction case, assume that every minimal DRE for $L(r^{\varepsilon}_{m-1})$ is of the form $r_{m-1} + \varepsilon$ where $L(r_{m-1}) = L_{m-1}$.

The expression r^{ε}_m cannot have a Kleene star as topmost operation in its syntax tree because $L_m \cup \{\varepsilon\}$ is finite and $L_m \neq \{\varepsilon\}$.

Now, assume that r^{ε}_m has a concatenation operation as topmost operation in its syntax tree, i.e., $r^{\varepsilon}_m = s'_1 \cdot s'_2$ for some DREs s'_1 and s'_2. Then, we know that $\varepsilon \in L(s'_1)$ and $\varepsilon \in L(s'_2)$. By definition of the language, every word ($\neq \varepsilon$) in $L(s'_1)$ and every word ($\neq \varepsilon$) in $L(s'_2)$ has to end with b. Let ub be a longest word in $L(s'_1)$ and vb be a longest word in $L(s'_2)$ such that $ubvb \in L(r^{\varepsilon}_m)$. By definition of the language L^{ε}_m it follows that also $uavb \in L(r^{\varepsilon}_m)$ such that $ua \in L(s'_1)$ and $ua \in L(r^{\varepsilon}_m)$. This contradicts that $L(r^{\varepsilon}_m) = L_m \cup \{\varepsilon\}$ and, therefore, r^{ε}_m is not a concatenation.

Therefore, r^{ε}_m has to be a disjunction $s'_1 + s'_2$ for some DREs s'_1 and s'_2. It remains to prove that s'_1 or s'_2 equals ε. Without loss of generality, assume that $\text{first}(s'_1) = \{a\}$ and $\text{first}(s'_2) = \{b\}$. We know that $\varepsilon \in L(s'_1)$ or $\varepsilon \in L(s'_2)$.

Without loss of generality, assume that $\varepsilon \in L(s'_1)$. Since $\text{first}(s'_1) = \{a\}$ we have that $L(s'_1) \neq \{\varepsilon\}$ and, therefore, s'_1 cannot be a deterministic expression. Thus, it holds that $\text{first}(s'_1) = \{a, b\}$ and $\text{first}(s'_2) = \{\varepsilon\}$ without loss of generality which concludes the proof. □

3.2. DREs for Infinite Languages

In the case of infinite languages it is well known that it is possible to have an unavoidable exponential blow-up when translating between REs and DFAs.

Theorem 8 ([7, 16]).

- *The minimal DFA for the language $L((a+b)^*a(a+b)^n)$ has size $2^{\Theta(n)}$.*
- *There exists a family of infinite regular languages $(L_n)_{n\in\mathbb{N}}$ such that the minimal DFA for L_n has size $\Theta(n^2)$ and every minimal RE for L_n has size $2^{\Omega(n)}$.*

To the best of our knowledge, all languages that are used in the literature to prove such blow-ups are not DRE-definable. Here, we prove that such a blow-up also cannot be avoided for DRE-definable languages. To prove an exponential blow-up when translating an RE for a DRE-definable language to a DFA, we can easily extend the language of Theorem 4 to an infinite language. For the exponential blow-up when translating a DFA for an infinite language to a DRE we can extend Theorem 7 accordingly.

Corollary 9. *Let $\Sigma = \{a, b, \#\}$.*

- *For each $n \in \mathbb{N}$, the minimal DFA and every minimal DRE for the DRE-definable language $L((a+b)^{0,n}a(a+b)^n\#^*)$ have size $2^{\Omega(n)}$.*
- *Let $L_n = L((a+b)^{0,n}b\#^*)$ for some $n \geq 1$. Every minimal RE for L_n has size $\Theta(n)$, the minimal DFA for L_n has size $\Theta(n)$, and every minimal DRE for the language has size $2^{\Omega(n)}$.*

We now present another, more general technique to show lower bounds on the descriptional complexity of DREs. The main idea of the technique is to identify positions in the minimal DFA where a minimal DRE can have a concatenation. To this end, we search for *bottleneck states* that are states which every accepting run needs to visit. Using bottlenecks, we show an exponential blow-up when translating a DFA into a DRE by a different and much more complex technique than in Section 3.1. The reason why we show both techniques is that, in this way, we show that there exist two independent sources of exponential behaviour which we can identify by the different techniques (see, e.g., Theorem 7 and 17). Moreover, we show in both scenarios that there exist infinitely many languages for which an exponential blow-up cannot be avoided.

Definition 10. *Let $A = (Q, \Sigma, \delta, q_0, Q_f)$ be a DFA. A state $q \in Q \setminus \{q_0\}$ is a* bottleneck state *of A if,*

- *for every word* $w \in L(A)$*, there are words* $v, z \in \Sigma^*$ *such that* $w = v \cdot z$ *and* $\delta^*(q_0, v) = q$*, and*
- *if* $q \in Q_f$ *then* $Q_f = \{q\}$ *and there exist* $a \in \Sigma$*,* $p \in Q$ *such that* $\delta(q, a) = p$*.*

That is, bottleneck states are states that are visited by every word in the language and, if they are accepting, then they have an outgoing transition and the automaton does not have any other accepting states. Notice that we explicitly define initial states not to be bottleneck states.

Lemma 11. *Let* $A = (Q, \Sigma, \delta, q_0, Q_f)$ *be a DFA with a bottleneck state* q*. Then* A *has no equivalent DRE that is atomic or of the form* s^**.*

Proof. Let r be a DRE for $L(A)$. By the definition of a bottleneck state it holds that $q \neq q_0$ and therefore $\varepsilon \notin L(A)$. Thus, r cannot be of the form ε or s^*. Since q has to have at least one outgoing transition, r is neither an atomic expression a nor $\emptyset$. □

Next, we show that accepting bottleneck states in a DFA identify concatenations in at least one equivalent minimal DRE. Let $A = (Q, \Sigma, \delta, q_0, Q_f)$ be a DFA. We say that a DRE r is a *q-concatenation for A* if (1) $L(r) = L(A)$, (2) $r = r_1 \cdot r_2$, and (3) $\delta^*(q_0, v) = q$ in A for every $v \in L(r_1)$. We call r a *partial q-concatenation for A* if $L(r) \subsetneq L(A)$ and r fulfills conditions (2) and (3) of a q-concatenation.

Lemma 12. *Let* $A = (Q, \Sigma, \delta, q_0, \{q_f\})$ *be a DFA for a DRE-definable language* L *such that* q_f *is a bottleneck state of* A*. Then every minimal DRE* r *for* L *is a* q_f*-concatenation* $r_1 \cdot r_2$ *with* $\text{first}(r_2) = \{a \in \Sigma \mid \delta(q_f, a) \text{ is defined}\}$*.*

Proof. By Lemma 11, it holds that r is neither atomic nor of the form s^*. It remains to show that r is neither a disjunction nor a concatenation which is not a q_f-concatenation. To this end, we prove the following claim:

Claim 13. *Let* $A = (Q, \Sigma, \delta, q_0, \{q_f\})$ *be a DFA for a DRE-definable language* L *such that* q_f *is a bottleneck state of* A*. Let* $\emptyset \neq S \subseteq \text{first}(L)$ *and* $r = r_1 r_2 \cdots r_n$ *(with* $n > 1$*) be a minimal DRE for* $L \cap S\Sigma^*$*. Then there exists an* $i \in \{1, \ldots, n-1\}$ *such that,*

- *for every word* $w \in L(r_1 \cdots r_i)$*, it holds that* $\delta^*(q_0, w) = q_f$*, and*
- $\text{first}(r_{i+1} \cdots r_n) = \{a \in \Sigma \mid \delta(q_f, a) \text{ is defined}\}$*.*

In particular, this means that r *is a partial* q_f*-concatenation for* A*.*

Before proving the claim we show how we use Claim 13 to prove the lemma. From the discussion above, we know that r is either a disjunction or a concatenation.

If r is a disjunction $(s_1 + \cdots + s_k)$ (where the s_i themselves are not disjunctions) then we use Claim 13 to prove that r is not minimal, which is a contradiction. In particular, we do this by applying Claim 13 to every s_i. However,

first we have to show that we can apply Claim 13. We show that, for every i, (a) $L(s_i) = L \cap S_i\Sigma^*$ with $\emptyset \subsetneq S_i \subseteq \text{first}(L)$ and (b) s_i is a concatenation.

Since r is a DRE it holds that $\text{first}(s_i) \cap \text{first}(s_j) = \emptyset$ for all $i \neq j$. Moreover, we know that $\varepsilon \notin L$ and, thus, $\varepsilon \notin L(s_i)$ for every i. It follows that $L(s_i) = L \cap S_i\Sigma^*$ with $S_i = \text{first}(s_i) \subseteq \text{first}(r)$ for every i, which proves (a). Next we prove (b), i.e., every s_i is a concatenation. Towards a contradiction, assume that there exists an s_i that is not a concatenation. By the structure of $r = (s_1 + \cdots + s_k)$, we know that s_i is not a disjunction. Since $\varepsilon \notin L(s_i)$, expression s_i cannot be a Kleene star either. We now show that s_i is not atomic. Take an arbitrary $a \in S_i$. Then there exists a word $aw \in L(r)$ where $w \neq \varepsilon$ because q_f has at least one outgoing transition. Since r is a DRE we know that $aw \in L(s_i)$. As $|aw| > 1$, s_i cannot be atomic. The only remaining possibility is that s_i is a concatenation, which proves (b).

This means that we can apply Claim 13 to every s_i, which implies that we can write every s_i as $s_i' \cdot s_i''$ such that (i) $\delta(q_0, w) = q_f$ for every $w \in L(s_i')$ and (ii) $\text{first}(s_i'') = \{a \in \Sigma \mid \delta(q_f, a) \text{ is defined}\}$. Here, s_i' and s_i'' can also be concatenations themselves.

Let $A^{q_f} = (\Sigma, Q, \delta, q_f, \{q_f\})$ be the automaton A where we changed the initial state to q_f. From (i) and (ii), we can conclude that $L(s_i'') = L(A^{q_f})$ for every i. Thus, all expressions s_i'' are equivalent. Therefore, r can equivalently be written as $(s_1' + \cdots + s_k') \cdot s_1''$, which is strictly smaller than r. Since this contradicts the assumption that r is minimal, we know that r cannot be a disjunction.

Hence, r is a concatenation and applying Claim 13 directly on r implies the lemma statement. Notice that Claim 13 can indeed be applied because if $S = \text{first}(r)$, then $L \cap S\Sigma^* = L$. The latter equality holds because $\varepsilon \notin L$ (by definition of bottleneck states). So it remains to prove Claim 13.

Proof of Claim 13.

In the following, let $A = (Q, \Sigma, \delta, q_0, \{q_f\})$ be a DFA for a DRE-definable language L such that q_f is a bottleneck state of A. Furthermore, for a set $S \neq \emptyset$ and $S \subseteq \text{first}(L)$, let r be a minimal DRE for $L \cap S\Sigma^*$.

The proof is by induction on $m = |r|$. We first argue that $|r| \geq 4$. By assumption, r is a concatenation which describes an infinite language. Therefore, r is a concatenation of at least two alphabet symbols and it also contains a Kleene star operation. This means that $|r| \geq 4$ and we use $|r| = 4$ as the induction base case.

If $|r| = 4$ then, using the same arguments as above, r can only be of the form $a^* \cdot b$ or $a \cdot b^*$ for some $a, b \in \Sigma$. If $r = a^* \cdot b$ then we know that $a \neq b$ because r is deterministic. It follows that $ab \in L(r)$ but $ab \cdot z \notin L(r)$ for every word $z \neq \varepsilon$. Since this contradicts that q_f has at least one outgoing transition, r has to be of the form $a \cdot b^*$. This implies that $\delta(q_0, a) = q_f$ in A, which makes r a partial q_f-concatenation for A. Since $L(r) = L \cap S\Sigma^*$ and A has only one accepting state, it holds that $\text{followlast}(L(r)) = \text{followlast}(L) = \{b\}$, which proves that Claim 13 holds for $|r| = 4$.

For the induction step, we assume that Claim 13 holds for all minimal DREs

$s = s_1 \cdots s_\ell$ for a language $L \cap S\Sigma^*$ with $\ell \geq 2, S \subseteq \text{first}(L)$, and $|s| < m$. We now prove that Claim 13 also holds for $r = r_1 \cdots r_n$ with $|r| = m$. To this end, let i be maximal such that $\varepsilon \notin L(r_i \cdots r_n)$. Since $\varepsilon \notin L(r)$, i is well-defined. Let $T := \{a \mid \delta(q_f, a) = q\} \setminus \text{first}(r_{i+1} \cdots r_n)$. We note that, in the case $i = n$, T equals $\{a \mid \delta(q_f, a) = q\}$.

We first look at the case $i < n$ and $T = \emptyset$. In this case, $\varepsilon \in L(r_{i+1} \cdots r_n)$ and, therefore, for every word $w \in L(r_1 \cdots r_i)$, we also have that $w \in L$. Thus, $\delta^*(q_0, w) = q_f$, which implies that r is a partial q_f-concatenation for A. Furthermore, $T = \emptyset$ implies that $\text{first}(r_{i+1} \cdots r_n) = \{a \in \Sigma \mid \delta(q_f, a) = q\}$, which implies that Claim 13 holds in this case.

The remaining case is that either $i = n$ or $i < n$ and $T \neq \emptyset$. We show that this case contradicts the assumption that r is minimal.

Due to the maximality of i, we know that $\varepsilon \notin L(r_i)$. It follows that r_i cannot be of the form s^* or ε. Let u, v, and w be words such that $\delta^*(q_0, uv) = q_f$, $\delta^*(q_f, w) = q_f$, $\text{first}(w) \in T$, $u \in L(r_1 \cdots r_{i-1})$, and $v \in L(r_i)$. By assumption, u, v, and w exist. It easy to see that $L(vw^*) \subseteq L(r_i)$ by definition of T and the determinism of r. This implies that $L(r_i)$ is infinite and, therefore, r_i cannot be an atomic expression.

The only remaining possibility is that r_i is of the form $(s_1 + \cdots + s_k)$ with $k \geq 2$. Moreover, using the same arguments than above, none of the s_j's are atomic expressions or of the form s^*. Finally, it follows that all s_j have to be concatenations and, since r_i is a DRE it holds that, for every s_j,

$$L(s_j) = L(r_i) \cap S_j\Sigma^*, \text{ with } S_j \subseteq \text{first}(L(r_i)) \text{ for all } j \in \{1, \ldots, k\}.$$

To apply the induction hypothesis on r_i, we first prove that the minimal DFA A' for $L(r_i)$ fulfils the following conditions:

(a) A' has exactly one accepting state p_f;

(b) A' has a p_f-outgoing transition; and

(c) p_f is not initial in A'.

Notice that conditions (a) to (c) imply that the minimal DFA A' for $L(r_i)$ has an accepting bottleneck state.

Condition (c) obviously holds because $\varepsilon \notin L(r_i)$. It remains to show (a) and (b). Remember that $L(r) = L(A) \cap S\Sigma^*$ and A has only one accepting state.

Let u be some word from $L(r_1 \cdots r_{i-1})$. As r is a DRE, we can conclude that $L(r_i \cdots r_n) = u^{-1}L(r) \cap \text{first}(L(r_i))\Sigma^*$. It follows that, for all words $v, w \in L(r_i)$,

$$v^{-1}L(r_i \cdots r_n) = w^{-1}L(r_i \cdots r_n) = v^{-1}u^{-1}L(A).$$

As r is a DRE, we can furthermore conclude that

$$v^{-1}L(r_i) = w^{-1}L(r_i) \text{ for all words } v, w \in L(r_i),$$

because, for each word z in $L(r)$, the decomposition into $z_1 \in L(r_1 \cdots r_i)$ and $z_2 \in L(r_{i+1})$ is unique. (Notice that if $i = n$, then $z_2 = \varepsilon$.)

Thus, the minimal DFA A' for $L(r_i)$ has only one accepting state p_f. Furthermore, p_f has an outgoing transition, since q_f has an outgoing transition labeled with a letter that is not in T. This shows (a) and (b).

We can now apply the induction hypothesis to every expression s_j. We obtain that every s_j is a partial p_f-concatenation for A' of the form $s_j^1 \cdot s_j^2$ where $\text{first}(s_j^2) = T$. Hence, all s_j^2 are equivalent and $(s_1^1 + \cdots + s_k^1) \cdot s_1^2$ is a DRE for $L(r_i)$ which is strictly smaller than r_i. This contradicts that r is minimal. □

Notice that, if $A = (Q, \Sigma, \delta, q_0, \{q_f\})$ is a DFA with a bottleneck state q_f then $L(A)$ is infinite. In this case, Lemma 12 gives us a rather precise structure of a minimal DRE of the form $r_1 \cdot r_2$. By the following lemma we can now clarify the languages $L(r_1)$ and $L(r_2)$.

Lemma 14. *For a DFA $A = (Q, \Sigma, \delta, q_0, \{q_f\})$ with a bottleneck state q_f, let the q_f-concatenation $r_1 \cdot r_2$ be an equivalent minimal DRE with* $\text{first}(r_2) = \{a \in \Sigma \mid \delta(q_f, a) \text{ is defined}\}$. *Then*

(1) $L(r_1) = L(A_S)$ where $S = \{a \in \Sigma \mid (\delta(q_f, a) = q) \wedge q \in Q\}$; and

(2) $L(r_2)$ is infinite where $L(r_2) = L(A^{q_f})$ with $A^{q_f} = (Q, \Sigma, \delta, q_f, \{q_f\})$.

Proof. (1) We prove $L(A_S) \subseteq L(r_1)$ first. Let $w = a_1 \cdots a_n$ be a word in $L(A_S)$. By definition of A_S, there is an accepting run $q_1 \cdots q_n$ of A_S on w such that the smallest i with $q_i = q_f$ is n. Since $w \in L(A) = L(r_1 \cdot r_2)$, we have that $w = w_1 \cdot w_2$ with $w_1 \in L(r_1)$ and $w_2 \in L(r_2)$. However, since $r_1 \cdot r_2$ is a q_f-concatenation, we have that $\delta^*(q_0, w_1) = q_f$. It follows that $w_1 = w$ and, therefore, $w \in L(r_1)$.

Next, we prove that $L(r_1) \subseteq L(A_S)$. Towards a contradiction, assume that w is a word in $L(r_1)$ such that $w \notin L(A_S)$. Since $r_1 \cdot r_2$ is a q_f-concatenation, we have that $\delta^*(q_0, w) = q_f$ and $w \in L(A)$. Let $w = a_1 \cdots a_n$ and $q_1 \cdots q_n$ be the accepting run of w in A. Since $w \notin A_S$ by assumption, we have that $q_1 \cdots q_n$ is not an accepting run of A_S on w. By definition of A_S, this means that there is an $i < n$ such that $q_i = q_f$ and that $\delta(q_f, a_{i+1}) = q_{i+1}$ in A. Take the minimal such i. We now have that $a_1 \cdots a_i \in L(A_S)$. Since we already proved that $L(A_S) \subseteq L(r_1)$, we also have that $a_1 \cdots a_i \in L(r_1)$. Moreover, we have that $a_{i+1} \in \text{first}(r_2)$ by the lemma statement. But this contradicts that $w \in L(r_1)$, since $r_1 \cdot r_2$ is a DRE. Therefore, it holds that $L(r_1) \subseteq L(A_S)$.

(2) By definition of a q_f-concatenation, we know that $\delta^*(q_f, w) = q_f$, for every $w \in L(r_2)$ in A. This directly implies that $L(r_2)$ is infinite and $L(r_2) \subseteq L(A^{q_f})$. It remains to prove that $L(A^{q_f}) \subseteq L(r_2)$. Let w be a word in $L(A^{q_f})$. If $w = \varepsilon$, then $w \in L(r_2)$. Now, assume that $w = aw'$ with $a \in \Sigma$. Then, we know that $\delta^*(q_f, aw') = q_f$ in A^{q_f} and in A. It follows, for every word $v \in L(A_S)$, that $\delta^*(q_0, vaw') = q_f$ in A and $vaw' \in L(r_1 \cdot r_2)$. By the lemma statement, $a \in \text{first}(r_2)$ and $r_1 \cdot r_2$ is a DRE such that $aw' = w \in L(r_2)$. □

Algorithm 1 Delete-Epsilon(r)

Require: DRE $r = r_1 + \cdots + r_k$ with $k \geq 1$ and $r \neq \varepsilon$
Ensure: DRE r^- with $L((r^-)^*) = L(r^*)$ and $\varepsilon \notin L(r^-)$

```
r⁻ := r
while ε ∈ L(r⁻) do
    for all r_i = s_1 ··· s_ℓ with ε ∈ L(r_i) do
        r_i := s_1 + ··· + s_ℓ
    for all r_i = s* do
        r_i := s
    r⁻ := Σ_{r_i ≠ ε} r_i
return r⁻
```

Before we can finally apply bottleneck states to prove the exponential blow-up from DFAs to DREs, we need a minor general result on minimal DREs. This result is a very straightforward property of a state and a concatenation in a DRE. As is well-known, we say that a regular language L is *prefix-free* if and only if, for every word $v \in L$, there exists no $z \in \Sigma^+$ such that $v \cdot z \in L$.

Lemma 15. *Let $L_a = L \cdot \{a\}$ be a prefix-free DRE-definable language. Then there exists a minimal DRE for L_a which is either a or of the form $r \cdot a$.*

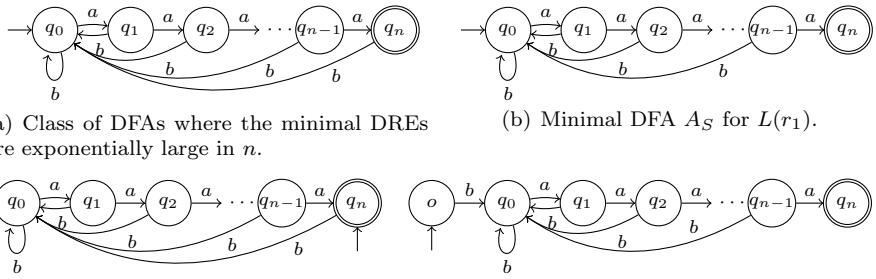
Proof. The proof is by structural induction on a minimal DRE r for L_a. For the induction base case, $r = a$, the assumption holds. (Notice that L_a cannot be $L(\emptyset)$ or $L(\varepsilon)$ by definition.)

For the induction case, assume that r has r_1 and r_2 as immediate subexpressions. Notice that r cannot be a Kleene star expression due to the fact that L_a never contains ε. Furthermore, let the assumption hold for DREs r_1 and r_2.

Now, assume that r is a disjunction, i.e., $r = r_1 + r_2$. Then we have that $L(r_1) = L_1 \cdot \{a\}$ and $L(r_2) = L_2 \cdot \{a\}$ for some DRE-definable languages L_1 and L_2. By the induction hypothesis, it follows that, for every $i = \{1, 2\}$, there exists a minimal DRE for r_i of the form a or $s_i \cdot a$. This implies that r is of the form $(a + a)$, $(s_1 \cdot a + s_2 \cdot a)$, or $(s_i \cdot a + a)$ for some $i \in \{1, 2\}$. In each case, there exists an expression for $L(r)$ that is of the same size or smaller, namely a, $(s_i + \varepsilon) \cdot a$, or $(s_1 + s_2) \cdot a$, respectively.

Now, let r be a concatenation of the form $r = r_1 \cdot r_2$. Because L_a is prefix-free it follows that $\varepsilon \notin L(r_2)$. Thus, $L(r_2)$ is of the form $L' \cdot \{a\}$ for some DRE-definable language L'. By induction hypothesis, there is a minimal DRE for $L(r_2)$ which is of the form a or $r_3 \cdot a$. Since r is a DRE, we know that $\text{followlast}(L(r_1)) \cap \text{first}(L(r_2)) = \emptyset$. Thus, there exists a minimal DRE for L_a which is of the form $r_1 a$ or $r_1 r_3 a$. □

Finally, we are ready to prove an exponential blow-up when translating DFAs to DREs using bottleneck states. In particular, we prove that every minimal DRE for the DFA in Figure 2(a) is exponential in n. We denote the language of this DFA with $L^{[n]}$.



(a) Class of DFAs where the minimal DREs are exponentially large in n.

(b) Minimal DFA A_S for $L(r_1)$.

(c) Minimal DFA A^{q_n} for $L(r_2)$.

(d) Minimal DFA for $L(s_2)$.

Figure 2: Minimal DFAs for subexpressions from the proof of Lemma 16.

Lemma 16. *For every $n > 0$, there exists a minimal DRE for the language $L^{[n]}$ that contains at least 2^n concatenations.*

Proof. Let A be the minimal DFA for $L^{[n]}$ (see Figure 2(a)). The proof is by induction on n. For the induction base, $n = 1$, we observe that A has an accepting bottleneck state. By Lemma 12, we know that r is a concatenation $r_1 \cdot r_2$ with $\text{first}(r_2) = \{b\}$. By Lemma 14, it follows that $L(r_1) = L(b^*a)$. Thus, there is a minimal DRE for $L^{[1]} = L(b^* \cdot a \cdot r_2)$ with at least two concatenations.

For the induction step, assume that there exists a minimal DRE for $L^{[n-1]}$ containing at least 2^{n-1} concatenations.
Let r_n be a minimal DRE for $L^{[n]}$. By Lemma 12, r_n is a q_n-concatenation $r_1 \cdot r_2$ with $\text{first}(r_2) = \{b\}$. Lemma 14 implies that the automaton in Figure 2(b) is a DFA for $L(r_1)$ and the automaton in Figure 2(c) is a DFA for $L(r_2)$.

Next, we show that r_1 and r_2 each contain a subexpression for the language $L^{[n-1]}$. For r_1, observe that $L(r_1)$ (see Figure 2(b)) is prefix-free and its language is of the form $L' \cdot \{a\}$. By Lemma 15, there exists a minimal DRE r_1 of the form $s_1 \cdot a$ such that $L(s_1)$ is defined by the DFA in Figure 2(b) without the transition $\delta(q_{n-1}, a) = q_n$ and with q_{n-1} as accepting state. Hence, $L(s_1) = L^{[n-1]}$ such that, by applying the induction hypothesis, there exists a DRE $r_1 = s_1 \cdot a$ containing at least 2^{n-1} concatenations.

For r_2 observe that $L(r_2)$ is infinite (see A^{q_n} in Figure 2(c)), which implies that r_2 is not an atomic expression. Moreover, it holds that $|\text{first}(r_2)| = 1$ and $\varepsilon \in L(r_2)$, which means that r_2 cannot be a concatenation and deterministic. Next, we show by contradiction that r_2 cannot be a disjunction. Since $\text{first}(r_2) = \{b\}$, the only possible disjunction for the DRE r_2 is of the form $r_2 = b \cdot r_3 + \varepsilon$ for some DRE r_3. As $\delta(q_n, b) = q_0$ in A^{q_n}, it follows that $L(r_3) = L^{[n]}$, which directly contradicts that r is a minimal DRE for $L^{[n]}$.

Hence, r_2 has to be an expression of the form s_2^*. We investigate the structure of a DFA for $L(s_2)$ in the following. For every word $v \in L(s_2)$, it holds that $\delta^*(q_n, v) = q_n$ in A^{q_n}. Since $r_2 = s_2^*$ is a DRE and $\text{first}(r_2) = \{b\}$, we have that $L(s_2)$ cannot contain a word v such that $v = wz$ with $w, z \neq \varepsilon$ and $\delta^*(q_n, w) = q_n$. These properties uniquely characterize $L(s_2)$, for which the

minimal DFA is shown in Figure 2(d). Because the DFA has a bottleneck state, s_2 cannot be atomic or an expression of the form t^* by Lemma 11. The expression s_2 is not a disjunction because $|\operatorname{first}(s_2)| = 1$, $\varepsilon \notin L(s_2)$, and s_2 is a DRE. Thus, s_2 is a concatenation $b \cdot t$, where $L(t)$ is defined by the DFA from Figure 2(d) without the transition $\delta(o, b) = q_0$ and with q_0 as initial state. By Lemma 15, it follows that $s_2 = b \cdot t \cdot a$, where $L(t) = L^{[n-1]}$. Thus, by the induction hypothesis, there exists a minimal DRE for r_2 containing at least 2^{n-1} concatenations. This concludes the proof. □

Since we can describe each language $L^{[n]}$ with $n \in \mathbb{N}$ using the regular expression

$$(b + ab + \cdots + a^n b)^* a^n = (b(a + b(\cdots(ab + b)\cdots)))^* a^n,$$

we obtain the following theorem.

Theorem 17. *For each $n \in \mathbb{N}$, every minimal RE for $L^{[n]}$ has size $\Theta(n)$, the minimal DFA for $L^{[n]}$ has size $\Theta(n)$, and every minimal DRE has size $2^{\Omega(n)}$.*

To demonstrate the utility of the technique, we give the proof for an unproved claim in [2] using bottlenecks. Brüggemann-Klein and Wood claimed that every minimal DRE for languages $L(\Sigma^* a_1 \cdots a_n)$ where $a_1 \cdots a_n$ is a fixed Σ-word is exponential in n [2]. However, to the best of our knowledge, no proof for this result exists in the literature. We can now prove this claim by using bottleneck states. Therefore, we will generalize the special structure of the automata of languages $L^{[n]}$ (see Figure 2(a)) to provide a formal proof.

Definition 18. *Let $A = (Q, \Sigma, \delta, o, \{q_n\})$ be a DFA, $\{a_1, \ldots, a_n\} \subseteq \Sigma$, and $\{q_0, \ldots, q_n\} \subseteq Q$. Then A contains a* bottleneck tail of length n *if A fulfills the following properties:*

1. *q_i is a bottleneck state for every $i \in \{0, \ldots, n\}$;*
2. *$\delta(q_{i-1}, a_i) = q_i$ for every $i \in \{1, \ldots, n\}$;*
3. *for all $i \in \{0, \ldots, n\}$, it holds that $\delta(q_i, a) = o$ for some $a \in \Sigma$; and*
4. *for all $i \in \{1, \ldots, n\}$, if $\delta(q, a) = q_i$ then $q = q_{i-1}$ and $a = a_i$.*

For instance, the DFA in Figure 2(a) and the minimal DFA for $L(\Sigma^* a_1 \cdots a_n)$ both contain a bottleneck tail of length $n - 1$. In the following, we prove that a bottleneck tail of length n causes a blow-up in an equivalent DRE that is exponential in n.

Theorem 19. *Let $A = (Q, \Sigma, \delta, o, \{q_n\})$ be a DFA for a DRE-definable language L with a bottleneck tail of length n. Then there exists a minimal DRE r for L which contains at least 2^n concatenations.*

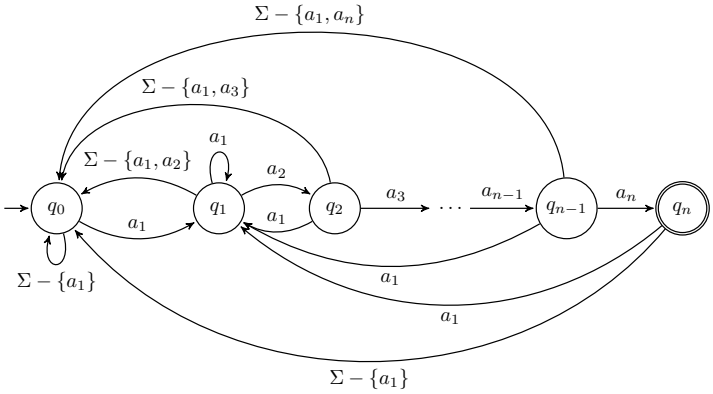


Figure 3: A minimal DFA A for $L(\Sigma^* a_1 \cdots a_n)$.

Proof. The proof is by induction on the length n of the bottleneck tail.

For the induction base case, let $n = 0$. By definition, A has at least one accepting bottleneck state q_0. (Recall that q_0 is not the initial state here.) By Lemma 12, we have that r is a q_0-concatenation $r_1 \cdot r_2$, which proves the assumption.

As induction hypothesis, we assume that every minimal DRE r_{n-1} for a DFA with a bottleneck tail of length $n-1$ contains at least 2^{n-1} concatenations. Moreover, let S be the set $\{a \in \Sigma \mid \delta(q_n, a) = q\}$ in the following.

Now, let A be a DFA for a DRE-definable language with a bottleneck tail of length n. We know that r is a q_n-concatenation of the form $r_1 \cdot r_2$ with $\text{first}(r_2) = S$ by Lemma 12. We prove that r_1 and r_2 each contain a subexpression r_{n-1}.

For r_1 we have that $L(r_1) = L(A_S)$ by Lemma 14. By definition of bottleneck tails, q_n has only one incoming transition labeled a_n. Therefore, $L(r_1) = L' \cdot \{a_n\}$. Since $L(r_1)$ is prefix-free we know by Lemma 15 that there is a minimal DRE for r_1 which is of the form $r'_1 \cdot a_n$. Moreover,

$$L(r'_1) = L(A'_S), \text{ where } A'_S = (Q \backslash \{q_n\}, \Sigma, \delta', o, \{q_{n-1}\}),$$

where δ' is the transition function of A_S without the transition $\delta(q_{n-1}, a_n) = q_n$. Observe that A'_S is a DFA with a bottleneck tail of length $n-1$. Applying the induction hypothesis, r'_1 contains at least 2^{n-1} concatenations.

For r_2 we know by Lemma 14 that A^{q_n} is a DFA for $L(r_2)$. By the structure of A^{q_n}, we know that r_2 is not atomic or ε. However, r_2 can be of the form $s_1 \cdots s_k$, $(s_1 + \cdots + s_k)$, or s^*. In the following, we prove that in each case we can find a subexpression s of r_2 such that there is a DFA for $L(s)$ which has a bottleneck tail of length $n-1$.

For the remainder of the proof, we denote by *last positions* of an expression r all symbols b of r such that b can be matched to a last symbol in a word of $L(r)$. Let b be the rightmost last position of r_2, i.e., the rightmost leaf of the parse

tree of r_2. Let $w \in L(r_2)$ be such that the last symbol of w is matched at b. (Observe that w is well-defined.) Moreover, we know that $wv \in L(r_2)$ for every word $v \in L(r_2)$, due to the structure of A^{q_n}. Hence, b has to be the last position of a subexpression s^* in r_2. Fix the minimal subexpression s^* of r that contains b. (In the parse tree of r_2, the expression s^* would correspond to the closest ancestor of b bearing the label $*$.) Notice that s^* could be included in some other starred subexpressions, i.e., there is a subexpression $(s_1(s_2 \dots (s_k \cdot s^*)^* \dots)^*)^*$ in r. However, by definition of b, there are no more alphabet symbols occurring to the right of s^* in r_2. Because $w^{-1}L(r_2) = L(r_2)$ we have that

$$\text{first}(s_1) \uplus \text{first}(s_2) \uplus \cdots \uplus \text{first}(s_k) \uplus \text{first}(s) = \text{first}(r_2).$$

By the structure of A^{q_n}, we know that $\text{followlast}(s) \subseteq \text{first}(r_2)$ and, since r_2 is deterministic, we know that

$$\text{followlast}(s) \cap (\text{first}(s_1) \cup \cdots \cup \text{first}(s_k) \cup \text{first}(s)) = \emptyset.$$

But then also $\text{followlast}(s) \cap \text{first}(r_2) = \emptyset$ and, therefore, $\text{followlast}(s) = \emptyset$. Hence, no reachable accepting state of any DFA for $L(s)$ has an outgoing transition. We show that s defines the following language in particular:

$$L(s) = L(A') \text{ where } A' = (Q \uplus \{o^{new}\}, \Sigma, \delta', o^{new}, \{q_n\}) \text{ and}$$
$$\delta' \text{ is the transition function of } A_S \text{ with the additional transition}$$
$$\delta'(o^{new}, a) = q \text{ where } a \in \text{first}(s) \text{ and } \delta(q_n, a) = q.$$

We prove $L(A') \subseteq L(s)$ first. Let $z \in L(A')$, then we have that $z \neq \varepsilon$, $\text{first}(z) \in \text{first}(s)$, and $z \in L(A^{q_n}) = L(r_2)$, by definition. Let $w \in L(r_2)$ be such that, when reading w in r_2, the last symbol of w is matched to the last symbol in s. By definition of s and due to the structure of r_2, we have that $wz \in L(r_2)$. But this implies that z should be matched by s, i.e., $z \in L(s)$.

Next, we prove $L(s) \subseteq L(A')$. Let $z \in L(s)$ and take $w \in L(r_2)$ such that, when reading w in r_2, the last symbol in w is matched to the last symbol in s. By definition of s and due to the structure of r_2, we have that $wz \in L(r_2)$. But this implies that $z \in L(A^{q_n})$ and, therefore, $L(s) \subseteq L(A^{q_n})$. Moreover, we already proved that $\text{followlast}(s) = \emptyset$ such that $L(s)$ has to be prefix-free. It follows that $L(s)$ contains exactly the words from $L(A^{q_n})$ which start with a symbol from $\text{first}(s)$ and do not have a non-empty prefix in $L(A^{q_n})$. Since this is exactly the language accepted by A', we have $L(s) \subseteq L(A')$.

Now we know that $L(s) = L(A')$, it remains to show that A' contains a bottleneck tail of length $n-1$. Notice that the states $q_0, \dots, q_{n-1}$ remain unchanged in A'. By Definition 18 and Lemma 15, we get that s is of the form $s' \cdot a_n$. Moreover, by deleting the state q_n and making q_{n-1} accepting in A' we get a DFA for s' that contains a bottleneck tail of length $n-1$. By applying the induction hypothesis, we get that r_2 contains at least 2^{n-1} concatenations.

Finally we have shown that r_1 and r_2 contain 2^{n-1} concatenations each. It follows that r has at least 2^n concatenations which concludes the proof. $\square$

Op.	$\lvert\Sigma\rvert = 1$	$\lvert\Sigma\rvert \geq 1$	Op.	$\lvert\Sigma\rvert = 1$	$\lvert\Sigma\rvert \geq 1$	Op.	$\lvert\Sigma\rvert = 1$	$\lvert\Sigma\rvert \geq 1$
$\setminus$	no	no	$\cup$	no	no	$\cdot$	no	no
Rev	yes	no	$\cap$	yes	no	*	yes	no

Table 2: Closure Properties of DRE-definable languages.

Theorem 20. *Every minimal DRE for $L(\Sigma^* a_1 \cdots a_n)$ has size $2^{\Omega(n)}$.*

Proof. The minimal DFA A for $L(\Sigma^* a_1 \cdots a_n)$ is shown in Figure 3. As we can see, A contains a bottleneck tail of length $n-1$. By Theorem 19, we know that there is a minimal DRE for $L(A)$ which contains $2^{\Omega(n)}$ concatenations. Thus, every minimal DRE for $L(A)$ has at least size $2^{\Omega(n)}$. □

4. Closure Properties of DRE-Definable Languages

To investigate the descriptional complexity of several language-theoretic operations on DREs and their DFAs in Section 5, we present an overview of the closure properties of DRE-definable languages first.

It has been observed that DRE-definable languages are not closed under union [2], intersection [4, 24] or complement [10]. DRE-definable languages are also not closed under concatenation [2], reversal[2] (take $L((a+b)^* a(a+b))$) or Kleene star [2]. These results hold for alphabets with at least two symbols. For unary alphabets, the same results hold, except for reversal, intersection and Kleene star. In these three cases, we prove that DRE-definable languages are closed. All results are summarized in Table 2. It is easy to see that DRE-definable languages over unary alphabets are closed under reversal, since for unary alphabets the language and its reversal are equal. In the following we show the remaining two cases.

DFAs over a unary alphabet have a very restricted form. The following notions come from, e.g., Shallit [30], but we repeat them here for completeness. (Notice that, Shallit used *tail* to refer to what we call a *chain*.) A DFA with initial state q_0 and state set $Q = \{q_0, \ldots, q_{n+m}\}$ is a *chain followed by a cycle* if its transition function is of the form $\delta(q_0, a) = q_1$, $\ldots$, $\delta(q_{n-1}, a) = q_n$, $\delta(q_n, a) = q_{n+1}$, $\ldots$, $\delta(q_{n+m-1}, a) = q_{n+m}$, $\delta(q_{n+m}, a) = q_n$, where $q_i \neq q_j$ if $i \neq j$. Furthermore, we have that at least one of the states in $\{q_n, \ldots, q_{n+m}\}$ is an accepting state. We refer to the states $q_0, \ldots, q_{n-1}$ as *chain states* and to $q_n, \ldots, q_{n+m}$ as the *cycle states* of this DFA.

Lemma 21 ([30]). *Every minimal DFA for an infinite regular language over a unary alphabet is a chain followed by a cycle.*

Then, the next result follows directly from the characterization of DRE-definable languages in [2] (see, e.g., Theorem 1).

[2]The reversal of a language L is the set of words $\{a_n \cdots a_1 \mid a_1 \cdots a_n \in L\}$.

Corollary 22. *An infinite regular language L over a unary alphabet is DRE-definable if and only if it has exactly one accepting cycle state.*

Notice that, the languages L from the corollary above may have additional accepting chain states. Furthermore, we say that a regular language L over a unary alphabet $\{a\}$ is (n_0, n_1, x)-*periodic* if

(i) $L \subseteq L(a^{n_1}(a^x)^*)$, and

(ii) for every $n \in \mathbb{N}$ such that $nx \geq n_0$, L contains the word $a^{n_1}a^{nx}$, i.e., the word of a's of length $nx + n_1$.

We say that L is *ultimately periodic* if it is (n_0, n_1, x)-periodic for some (n_0, n_1, x). Notice that, these properties imply that L is infinite. In an ultimately periodic language all *sufficiently long* words must have the same length y (modulo x), for a fixed y. This length modulo x can be different from 0.

It follows that a language L over a unary alphabet is ultimately periodic if and only if the minimal DFA for L has exactly one accepting cycle state; hence it holds the following.

Corollary 23. *An infinite regular language L over a unary alphabet is DRE-definable if and only if it is ultimately periodic.*

We show that DRE-definable languages over unary alphabets are closed under Kleene star by proving the assumption for ultimately periodic languages. Therefore, we use the following proposition by Bézout.

Proposition 24 (Bézout's Identity). *For any numbers $k_1, \ldots, k_n \in \mathbb{N}$ there exist integers $x_1, \ldots, x_n \in \mathbb{Z}$ such that*

$$k_1x_1 + \cdots k_nx_n = \gcd(k_1, \ldots, k_n)$$

Lemma 25. *Let L be any language over a unary alphabet. Then L^* is ultimately periodic.*

Proof. Let $\mathcal{K} = \{k | a^k \in L\}$ be the lengths of the words in L. Let $d = \gcd(\mathcal{K})$. Obviously, we have that $L^* \subseteq (a^d)^*$. We prove that there exists an n_0 such that, for every natural number $x \geq \frac{n_0}{d}$, we have that $a^{dx} \in L^*$, thereby obtaining that L^* is ultimately periodic.

Let $\{k_1, \ldots, k_n\} \subseteq \mathcal{K}$, be a finite subset of $\mathcal{K}$, such that $\gcd(k_1, \ldots, k_n) = d$. Such a set exists, as the gcd decreases by adding more numbers and cannot be smaller than 1.

According to Bézout's Identity, there exist integers $x_1, \ldots, x_k$ such that $k_1x_1 + \cdots + k_nx_n = d$. Thus any multiple of d can be written as a linear combination of $k_1, \ldots, k_n$. It remains to show that there exists an n_0, such that the all multiples of d, which are greater than n_0 can be written as a positive linear combination of $k_1, \ldots, k_n$.

Let be $X = \max\{|x_1|, \ldots, |x_n|\}$ and $K = \max\{k_1, \ldots, k_n\}$ and take $n_0 = n^2XK(k_1 \cdot k_2 \cdots k_n)$.

Let $x \in \mathbb{N}$ be arbitrary such that $dx \geq n_0$. We choose $y, z \in \mathbb{N}_0$, such that $dx = n_0 + zn(k_1 \cdots k_n) + dy$ and $y < n \cdot k_1 \cdots k_n$.

Now we can write dx as follows:

$$\begin{aligned} dx &= n_0 + zn(k_1 \cdots k_n) + dy \\ &= n^2XK(k_1 \cdots k_n) + zn(k_1 \cdots k_n) + dy \\ &= \sum_{i=1}^{n} k_i((nXK + z) \cdot \tfrac{k_1 \cdots k_n}{k_i}) + \sum_{i=1}^{n} k_i x_i y \\ &= \sum_{i=1}^{n} k_i((nXK + z) \cdot \tfrac{k_1 \cdots k_n}{k_i} + x_i y) \end{aligned}$$

Note that every coefficient $c_i = (nXK + z) \cdot \frac{k_1 \cdots k_n}{k_i} + x_i y$ is positive, as $|x_i| < X$ and $y < nK \cdot \frac{k_1 \cdots k_n}{k_i}$. Note that $k_i \leq K$. Thus a^{dx} can be written as $(a^{k_1})^{c_1} \cdots (a^{k_n})^{c_n}$. □

We are now able to obtain the following.

Theorem 26. *DRE-definable regular languages over a unary alphabet are closed under intersection and Kleene star.*

Proof. Closure under Kleene star is immediate from Lemma 25. It remains to prove that DRE-definable languages over a unary alphabet are closed under intersection.

Since every finite language is DRE-definable (see, e.g., [1]), the intersection of two languages in which one is finite is always DRE-definable.

It remains to consider intersections of two infinite DRE-definable regular languages over an alphabet $\{a\}$. The result is obtained by Lemma 21, Corollary 22 and by observing that the minimal DFA for the intersection of two DFAs in which one cycle state is accepting, also has exactly one accepting cycle state. This proves that DRE-definable languages are closed under intersection. □

5. Descriptional Complexity of Operations on DRE-Definable Languages

In Section 5.1 we give a short overview of the state complexity of boolean operations on DFAs for DRE-definable languages. Afterwards, we investigate the descriptional complexity of boolean operations on DREs in Section 5.2. In both cases, we take a look on unary and arbitrary alphabets as well as finite and infinite languages separately.

5.1. Boolean Operations on DFAs

We summarize results on the state complexity of minimal DFAs for DRE-definable languages in Tables 3 and 4. In each case we consider a single use of a boolean operation and a k-times application. Notice that, we study DFAs without a sink state here. However, in most of the related work on the state complexity of minimal DFAs the authors considered complete DFAs. We chose to study DFAs without a sink state here to avoid confusion with the definition of

	$\lvert\Sigma\rvert = 1$		$\lvert\Sigma\rvert \geq 1$	
	1	k	1	k
$\setminus$	$\Theta(m)$ [16]	—	$\Theta(m)$ [16]	—
$\cap$	$\Theta(\min\{m_1, m_2\})$ [33]	$\Theta(\min\{m_1, ..., m_k\})$ [33]	$\Theta(m_1 m_2)$ [33]	$2^{\Omega(k)}$ (Cor. 27)
$\cup$	$\Theta(\max\{m_1, m_2\})$ [33]	$\Theta(\max\{m_1, ..., m_k\})$ [33]	$\Theta(m_1 m_2)$ [33]	$2^{\Omega(k)}$ (Cor. 27)

Table 3: State complexity of minimal DFAs for finite languages.

	$\lvert\Sigma\rvert = 1$		$\lvert\Sigma\rvert \geq 1$	
	1	k	1	k
$\setminus$	$\Theta(m)$ [16]	—	$\Theta(m)$ [16]	—
$\cap$	$\Theta(m_1 m_2)$ (Th. 28)	$k^{\Omega(k)}$ (Th. 28)	$\Theta(m_1 m_2)$ (Th. 28)	$k^{\Omega(k)}$ (Th. 28)
$\cup$	$\Theta(\max\{m_1 m_2\})$ (Th. 29)	$\Theta(\max\{m_1, ..., m_k\})$(Th. 29)	$\Theta(m_1 m_2)$ (Cor. 27)	$2^{\Omega(k)}$ (Cor. 27)

Table 4: State complexity of minimal DFAs for infinite DRE-definable languages

DRE-definable languages. Since the results on the state complexity of minimal complete DFAs always differs only by a constant from our results (that is, the missing sink state), we can still compare the results.

In general, we can transfer all existing results on state complexity of DFAs for finite languages to our setting. (Every finite language is DRE-definable.) As far as we know, there does not exist any previous work on the state complexity of DFAs for infinite DRE-definable languages.

It is well-known that for the complement on DFAs (for every regular language) there is no blow-up [16]. Since all finite languages are DRE-definable, we provide the known results of Yu [33] on DFAs for finite languages in Table 3. However, regarding these results notice the following. For the union and intersection of two finite languages, Yu proved an $m_1 m_2$ upper and lower bound. Nevertheless, they only stated the result for the upper bound in the paper since they were searching for the exact state complexity. Concerning this matter, it is easy to see that using the product construction the resulting automaton can never have exactly $m_1 m_2$ states. For example, the state (s, q) where s is the initial state of the first automaton and q is a non-initial state of the second automaton can neither be part of an automaton for the union nor for the intersection of two finite languages. As far as we know, this question regarding the exact state complexity of the union or intersection of two finite languages is still open.

From results in [16, 30, 33] we get that the descriptional complexity of union or intersection on k finite languages over an arbitrary alphabet is exponential in the worst case. For the union operation the result can be transferred to infinite DRE-definable languages over arbitrary alphabets.

Corollary 27 ([16, 30, 33]).

- *For every $k \in \mathbb{N}$, there exist finite languages $L_1, \ldots, L_k$ such that the minimal DFA for every L_i has $\Theta(k)$ states and the minimal DFA for $L_1 \cap \cdots \cap L_k$ or $L_1 \cup \cdots \cup L_k$ has at least $2^{\Omega(k)}$ states.*
- *For each $k \in \mathbb{N}$, there exist infinite DRE-definable languages $L_1, \ldots, L_k$ such that, for every $i \in \{1, \ldots, k\}$, the minimal DFA for L_i has k states,*

the language $L_\cup = L_1 \cup \cdots \cup L_k$ is DRE-definable, and the minimal DFA for $L_\cup$ has size $2^{\Omega(k)}$.

These results can be obtained when computing the intersection or union of the k distinct languages $I_{(\ell,k)} = \{x_1 \cdots x_k y_k \cdots y_1 | x_i, y_i \in \Sigma \wedge x_\ell = y_\ell\}$ where $\ell \in \{1, \ldots, k\}$ for example. To prove the result for the union of infinite DRE-definable languages over arbitrary alphabets we extend the above languages to languages of the form $I^{\inf}_{(\ell,k)} = \{x_1 \cdots x_k y_k \cdots y_1 \#^* | x_i, y_i \in \Sigma \backslash \{\#\} \wedge x_\ell = y_\ell\}$ where $\#$ is a new alphabet symbol.

For the intersection of infinite DRE-definable languages (over unary alphabets) we can obtain the worst case complexity by using k languages $L_i = L((a^{m_i})^*)$ with $1 \leq i \leq k$ and k different m_i such that $\gcd(m_i, m_j) = 1$ for each pair (m_i, m_j).

Theorem 28.

- *There exist infinitely many infinite DRE-definable languages L_1 and L_2 such that the minimal DFAs for L_1 and L_2 have m_1 and m_2 states, respectively, the language $L_1 \cap L_2$ is DRE-definable, and the minimal DFA for $L_1 \cap L_2$ has at least $\Theta(m_1 m_2)$ states.*
- *For each $k \in \mathbb{N}$, there exist infinite DRE-definable languages $L_1, \ldots, L_k$ such that, for every $i \in \{1, \ldots, k\}$, the minimal DFA for L_i has $O(k \log k)$ states, the language $L_\cap = L_1 \cap \cdots \cap L_k$ is DRE-definable, and the DFA for $L_\cap$ has $k^{\Omega(k)}$ states.*

Both results hold even when the alphabet is unary.

Finally, we prove that for the union of DFAs for DRE-definable languages over unary alphabets the descriptional complexity is linear; hence, strictly lower than for arbitrary regular languages.

Theorem 29. *For each $k \in \mathbb{N}$, let $L_1, \ldots, L_k$ be infinite DRE-definable languages over a unary alphabet such that, for every $i \in \{1, \ldots, k\}$, the minimal DFA for L_i has m_i states and the language $L_\cup = L_1 \cup \cdots \cup L_k$ is DRE-definable. Then the minimal DFA for $L_\cup$ has $\Theta(\max\{m_1, \ldots, m_k\})$ states.*

Proof. We prove the assumption for the union of two languages which directly implies the assumption for the union of k languages.

By Lemma 21 and Corollary 22, we know that DFAs for DRE-definable languages over a unary alphabet consist of a chain and a cycle where exactly one cycle state is accepting. Let A_1 and A_2 be the minimal DFAs for L_1 and L_2 and A_3 be the minimal DFA for $L_1 \cup L_2$. Let for the DFA A_i with $i \in \{1, \ldots, 3\}$ be Chain_i and Cycle_i the number of states in the chain and cycle of A_i. To prove the assumption it is sufficient to show that the following holds

1) $\text{Chain}_3 = \max\{\text{Chain}_1, \text{Chain}_2\}$, and

2) $\text{Cycle}_1 = \text{Cycle}_2$.

	$\setminus$	$\cap$	$\cup$	Rev	$\cdot$
two DREs of size $\Theta(n)$	$2^{\Omega(n)}$ (Th. 31)	$2^{\Omega(n)}$ (Th. 32)	$2^{\Omega(n)}$ (Th. 33)	$2^{\Omega(n)}$ (Th. 34)	$2^{\Omega(n)}$ (Th. 35)

Table 5: Descriptional complexity of boolean operations on DREs over arbitrary alphabets.

Note that 2) implies that $\text{Cycle}_3 = \text{Cycle}_1$. As 1) was proven in [30] it remains to show 2). Towards contradiction, assume that $\text{Cycle}_1 \neq \text{Cycle}_2$. Then it directly follows that $\text{Cycle}_3 > \text{Cycle}_1$. This implies, that the cycle of A_3 must have more than one accepting state which directly contradicts that $L_1 \cup L_2$ is DRE-definable. □

5.2. Boolean Operations on DREs

In this section we investigate the descriptional complexity of DREs that are itself the result of applying a boolean operation on some DREs. For regular languages, almost every operation causes an unavoidable exponential blow-up when representing the languages as regular expressions. Since DRE-definable languages are a strict subclass of all regular languages one could hope for a better complexity for the class of DRE-definable languages. In this section we show that this is not the case. Furthermore, remember that DRE-definable languages are not closed under any boolean operation which is summarized in Section 4. An overview of the results for DREs over arbitrary alphabets is shown in Table 5. However, for languages over a unary alphabet, one can always find a small DRE compared to the minimal DFA for the language which we prove in the following.

5.2.1. Boolean Operations on DREs over unary alphabets

For unary alphabets, the descriptional complexity of DREs is the same as for their DFAs (see, e.g., Tables 3 and 4). In more detail, we observe that, for DRE-definable languages over unary alphabets, minimal DREs are only linearly larger than the equivalent minimal DFA for the language.

Observation 30. *Let L be a DRE-definable language over a unary alphabet and A be a minimal DFA for L with m states. Then, there exists a minimal DRE r for L such that r is of size $O(m)$.*

5.2.2. Boolean Operations on DREs for arbitrary alphabets

We show first that complementing a DRE can cause an unavoidable exponential blow-up when representing the complement language as a DRE.

Theorem 31. *There exist DRE-definable languages $(L_n)_{n\in\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, a minimal DRE for L_n has size $\Theta(n)$ and a minimal DRE for $\overline{L_n} = \Sigma^* \setminus L_n$ has size $2^{\Omega(n)}$.*

Proof. We prove the assumption by showing that the language L_n^C for every $n \geq 1$ (see Figure 4) has a minimal DRE of size $\Theta(n)$ and that every minimal DRE for the language $\overline{L_n^C} = \Sigma^* \setminus L_n^C$ is at least exponential in n.

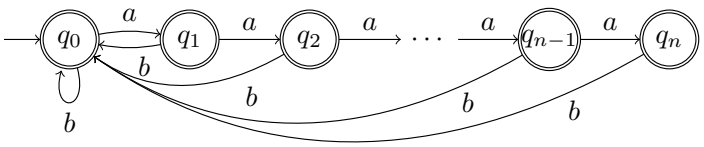


Figure 4: Minimal DFAs for languages L_n^C from Theorem 31.

Intuitively, the language L_n^C contains all words over the alphabet $\Sigma = \{a, b\}$ that do not contain the subword a^n. To see that L_n^C is DRE-definable, observe that

$$r_n = \underbrace{(\varepsilon + a(\varepsilon + a(\cdots)))}_{n \text{ times}} \cdot (b \cdot \underbrace{(\varepsilon + a(\varepsilon + a(\cdots)))}_{n \text{ times}})^*$$

is a DRE for L_n^C of size $\Theta(n)$.

It remains to show that, for the language $\overline{L_n^C} = \Sigma^* \backslash L_n^C$, every minimal DRE has at least size $2^{\Omega(n)}$. Therefore, observe that $\overline{L_n^C} = L^{[n]} \cdot L(a \cdot (a+b)^*)$. In the following we show that every minimal DRE for $\overline{L_n^C}$ is of the form $\overline{r_n} \cdot a(a+b)^*$ where $L(\overline{r_n}) = L^{[n]}$ (see Lemma 16). Then, the assumption directly holds by applying Lemma 16.

Let $\overline{r}$ be a minimal DRE for $\overline{L_n^C}$. We show first that $\overline{r}$ is a concatenation. Notice that $\overline{r}$ cannot be atomic, $\emptyset$ or a star expression. (For the latter, observe that $\varepsilon \notin \overline{L_n^C}$.) Towards contradiction, assume that $\overline{r}$ is a disjunction $s_1 + s_2$. Because $\varepsilon \notin \overline{L_n^C}$, it holds that $\text{first}(s_1) = \{a\}$ and $\text{first}(s_2) = \{b\}$ w.l.o.g.. Notice that s_1 and s_2 cannot be atomic, $\emptyset$, star expressions, or disjunctions. Therefore, they have to be concatenations. Moreover, s_1 and s_2 end with a star expression because every word in $L(\overline{r})$ is the prefix of another word in $L(\overline{r})$. Hence, they are of the form $s_1' \cdot (s_1'')^*$ and $s_2' \cdot (s_2'')^*$, respectively. Towards contradiction, assume w.l.o.g. that $(s_1'')^*$ is not equal to $(a+b)^*$. Then we can choose words $w \in L(s_1)$ and $v \notin L((s_1'')^*)$. Observe that $wv \in \overline{L_n^C}$ but $wv \notin L(\overline{r})$ because $\overline{r}$ is deterministic. This contradicts the assumption that $\overline{r}$ is a DRE for $\overline{L_n^C}$.

Therefore, $\overline{r} = s_1' \cdot (a+b)^* + s_2' \cdot (a+b)^*$, which directly contradicts that $\overline{r}$ is minimal. We proved that $\overline{r}$ is a concatenation.

Finally, it remains to prove that one cannot write $\overline{r}$ more succinctly than in $\overline{r_n} \cdot a(a+b)^*$. Analogously as above, we get that every minimal DRE for $\overline{r}$ is of the form $r' \cdot (a+b)^*$. It holds that $L(r') = L(\overline{r_n} \cdot a) = L^{[n]} \cdot \{a\}$. Thus, $L(r')$ contains all words over the alphabet $\{a, b\}$ that end with a^{n+1} but do not have other occurrences of the subword a^{n+1} (except that the suffix). Therefore, $L(r')$ is prefix-free such that we can apply Lemma 15 on the language. By Lemma 15, we get that every minimal DRE is of the form $\overline{r_n} \cdot a(a+b)^*$, which concludes the proof. □

Next, we prove an exponential blow-up for intersection and union.

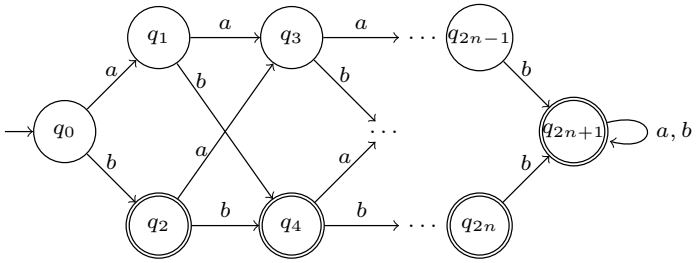


Figure 5: DFA for the language $L_n^{\inf}$

Theorem 32. *There exist DRE-definable languages $(L_n^1)_{n\in\mathbb{N}}$ and $(L_n^2)_{n\in\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, minimal DREs for L_n^1 and L_n^2 have size $\Theta(n)$ and a minimal DRE for $L_n^1 \cap L_n^2$ has size $2^{\Omega(n)}$.*

Proof. We prove the assumption by showing that the languages

$$L_n = L((a+b)^{0,n}b) \text{ for } n \in \mathbb{N},$$

can be written as the intersection of two DRE-definable languages with small DREs of size n. By Theorem 7, we know that a minimal DRE for L_n is exponentially large in n. Now, take the languages

$$L_n^1 = L((a^*b)^*) \text{ and } L_n^2 = L((a+b)^{1,n+1}).$$

It easy to see that L_n^1 is a DRE-definable language with a small DRE. For L_n^2, notice that the language is finite and, therefore, DRE-definable. Furthermore, L_n^2 has a minimal DRE of size $\Theta(n)$. Since $L_n = L_n^1 \cap L_n^2$ this concludes the proof. □

Theorem 33. *There exist DRE-definable languages $(L_n^1)_{n\in\mathbb{N}}$ and $(L_n^2)_{n\in\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, minimal DREs for L_n^1 and L_n^2 have size $\Theta(n)$ and a minimal DRE for $L_n^1 \cup L_n^2$ has size $2^{\Omega(n)}$.*

Proof. We prove the assumption by showing that the language $L_n^{\inf}$ from Figure 5 can be written as the union of two DRE-definable languages with a small DRE of size n and that every minimal DRE for $L_n^{\inf}$ is of size $2^{\Omega(n)}$.

First, notice that, $L_n^{\inf}$ can be written as the union of the languages

$$L_n^1 = L((a^*b)^*) \text{ and } L_n^2 = L((a+b)^{n+2}(a+b)^*).$$

It easy to see that L_n^1 is a DRE-definable language with a small DRE. Since L_n^2 is finite it is DRE-definable. Furthermore, L_n^2 has a minimal DRE of size $\Theta(n)$ and $L_n^{\inf} = L_n^1 \cup L_n^2$.

It remains to prove that every minimal DRE for the language $L_n^{\inf}$ is at least of size $2^{\Omega(n)}$. Therefore, we show that every minimal DRE for $L_n^{\inf}$ is of the form $r \cdot (a+b)^*$ where $L(r) = L_n$ (see Theorem 7). Then, the assumption holds by

applying Theorem 7. The proof to show that every minimal DRE for $L_n^{\inf}$ is of the form $r \cdot (a+b)^*$ follows the same lines as the proof of Theorem 31. □

Additionally, we get the following result for the reversal operation by taking the language $L((a+b)^{0,n}a(a+b)^n)$ with $n \in \mathbb{N}$ (see e.g. Theorem 4).

Theorem 34. *There exist DRE-definable languages $(L_n)_{n\in\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, the minimal DREs for L_n have size $\Theta(n)$, whereas the minimal DREs for the reversal of L_n have size $2^{\Omega(n)}$.*

For the concatenation operation, one cannot avoid an exponential blow-up either. To obtain the following theorem take the languages $L_n^1 = L((a+b)^{0,n})$ and $L_n^2 = L(a(a+b)^n)$ with $n \in \mathbb{N}$ (see, e.g., Theorem 4).

Theorem 35. *There exist DRE-definable languages $(L_n^1)_{n\in\mathbb{N}}$ and $(L_n^2)_{n\in\mathbb{N}}$ such that, for each $n \in \mathbb{N}$, the minimal DREs for L_n^1 and L_n^2 have size $\Theta(n)$ and the minimal DREs for $L_n^1 \cdot L_n^2$ have size $2^{\Omega(n)}$.*

6. Conclusions

We were motivated by the aim to come to a better understanding of DRE-definable languages. To this end, we investigated the descriptional complexity of representations for DRE-definable languages and proved that in the most cases the complexity is not better than for general regular expressions. In this paper we summarized old and new results on the descriptional complexity of DRE-definable languages in general and of boolean operations on DRE-definable languages.

We now know that, when translating an RE into a DFA and when translating a DFA into a DRE, an exponential blow-up cannot be avoided even for finite languages. For infinite languages, we developed a new technique to prove lower bounds on the size of DREs by using bottleneck states and tails in a DFA. It remains open whether there is a DRE-definable language that has an exponentially larger RE than its DFA and whether there is a DRE-definable languages for which a translation from an RE to a DRE causes a double exponential blow-up.

Moreover, we examined several operations on DRE-definable languages. We obtained an overview of the closure properties of these languages and showed that they are not closed under several language-theoretic operations. Since most of these operations are also relevant in XML schema management, this diminishes hope to have easy algorithms when processing DREs in schemas.

We continued examining the descriptional complexity of DFAs and DREs for DRE-definable languages that are itself the result of applying one of the considered operations on two DRE-definable languages. Since DFAs for DREs are of a very restricted form, one could hope that representations for such languages may be more succinct in general such that algorithms processing boolean operations on DREs could be simplified. Unfortunately, when applying any of the

considered operations only once on two DREs an exponential blow-up cannot be avoided in general.

Acknowledgments We thank the anonymous reviewers of the Theoretical Computer Science Journal for their constructive and insightful comments that helped to improve the presentation of the paper.

References

[1] G. J. Bex, W. Gelade, W. Martens, and F. Neven. Simplifying XML Schema: effortless handling of nondeterministic regular expressions. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 731–744. ACM, 2009.

[2] A. Brüggemann-Klein and D. Wood. One-unambiguous regular languages. *Information and Computation*, 142(2):182–206, 1998.

[3] C. Câmpeanu, K. Culik, K. Salomaa, and S. Yu. State complexity of basic operations on finite languages. In *Proceedings of the International Workshop on Implementing Automata (WIA)*, pages 60–70. Springer, 2001.

[4] P. Caron, Y. Han, and L. Mignot. Generalized one-unambiguity. In *Proceedings of the International Conference on Developments in Language Theory (DLT)*, pages 129–140. Springer, 2011.

[5] H. Chen and P. Lu. Checking determinism of regular expressions with counting. *Information and Computation*, 241: 302–320, 2015.

[6] W. Czerwiński, C. David, K. Losemann, and W. Martens. Deciding Definability by Deterministic Regular Expressions. In *Proceedings of the International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, pages 289–304, 2013.

[7] A. Ehrenfeucht and H. Zeiger. Complexity measures for regular expressions. *Journal of Computer and System Sciences (JCSS)*, 12(2):134–146, 1976.

[8] K. Ellul, B. Krawetz, J. Shallit, and M. Wang. Regular expressions: new results and open problems. *Journal of Automata, Languages and Combinatorics (JALC)*, 9(2-3):233–256, 2004.

[9] W. Gelade, T. Idziaszek, W. Martens, F. Neven, and J. Paredaens. Simplifying XML Schema: Single-type approximations of regular tree languages. *Journal of Computer and System Sciences (JCSS)*, 79(6):910–936, 2013.

[10] W. Gelade and F. Neven. Succinctness of the complement and intersection of regular expressions. *ACM Transactions on Computational Logic (TOCL)*, 13(1):4, 2012.

[11] W. Gelade, M. Gyssens, and W. Martens. Regular Expressions with Counting: Weak versus Strong Determinism. *SIAM Journal on Computing (SICOMP)*, 41(1): 160–190, 2012.

[12] B. Groz, S. Maneth, and S. Staworko. Deterministic regular expressions in linear time. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*, pages 49–60, 2012.

[13] H. Gruber and M. Holzer. Finite automata, digraph connectivity, and regular expression size. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, pages 39–50. Springer, 2008.

[14] H. Gruber and M. Holzer. Tight bounds on the descriptional complexity of regular expressions. In *Proceedings of the International Conference on Developments in Language Theory (DLT)*, pages 276–287. Springer, 2009.

[15] H. Gruber and J. Johannsen. Optimal lower bounds on regular expression size using communication complexity. In *Proceedings of the International Conference on Foundations of Software Science and Computational Structures (FOSSACS)*, pages 273–286. Springer, 2008.

[16] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Pearson Education, 2007.

[17] D. Hovland. Regular Expressions with Numerical Constraints and Automata with Counters. In *International Colloquium on Theoretical Aspects of Computing (ICTAC)*, pages 231–245. Springer, 2009.

[18] J. Jirásek, G. Jirásková, and A. Szabari. State complexity of concatenation and complementation of regular languages. In *Proceedings of the International Conference on Implementation and Application of Automata (CIAA)*. 2004.

[19] G. Jirásková. On the state complexity of complements, stars, and reversals of regular languages. In *Proceedings of the International Conference on Developments in Language Theory (DLT)*, pages 431–442. Springer, 2008.

[20] P. Kilpeläinen. Checking determinism of XML Schema content models in optimal time. *Information Systems*, 36(3): 596–617, 2011.

[21] P. Kilpeläinen and R. Tuhkanen. One-unambiguity of regular expressions with numeric occurrence indicators. *Information and Computation*, 205(6): 890-916, 2007.

[22] C. Kintala and D. Wotschke. Amounts of nondeterminism in finite automata. *Acta Informatica*, 13:199–204, 1980.

[23] M. Latte and M. Niewerth. Definability by Weakly Deterministic Regular Expressions with Counters is Decidable. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 369–381, 2015.

[24] K. Losemann. Boolesche Operationen auf deterministischen regulären Ausdrücken. Master's thesis, TU Dortmund, October 2010.

[25] K. Losemann, W. Martens, and M. Niewerth. Descriptional complexity of deterministic regular expressions. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 643–654. Springer, 2012.

[26] P. Lu, J. Bremer and H. Chen. Deciding Determinism of Regular Languages. *Theory of Computing Systems*, 57(1): 97–139, 2015.

[27] P. Lu, F. Peng, H. Chen, L. Zheng. Deciding determinism of unary languages. *Information and Computation*, 245: 181–196, 2015.

[28] R. Mandl. Precise bounds associated with the subset construction on various classes of nondeterminism finite automata. In *Proceedings of the Annual Princeton Conference on Information Science and Systems*, pages 263–267. Princeton University Press, 1973.

[29] W. Martens, M. Niewerth, and T. Schwentick. Schema design for XML repositories: Complexity and tractability. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*, ACM, 2010.

[30] G. Pighizzini and J. Shallit. Unary language operations, state complexity and Jacobsthal's function. *International Journal of Foundations of Computer Science*, 13(1):145–159, 2002.

[31] A. Salomaa, D. Wood, and S. Yu. On the state complexity of reversals of regular languages. *Theoretical Computer Science (TCS)*, 320(2-3):315–329, 2004.

[32] K. Salomaa and S. Yu. NFA to DFA transformation for finite languages over arbitrary alphabets. *Journal of Automata, Languages and Combinatorics (JALC)*, 2(3):177–186, 1997.

[33] S. Yu. State complexity of regular languages. *Journal of Automata, Languages and Combinatorics (JALC)*, 6(2):221, 2001.

[34] S. Yu, Q. Zhuang, and K. Salomaa. The state complexities of some basic operations on regular languages. *Theoretical Computer Science (TCS)*, 125(2):315 – 328, 1994.