Definability by Weakly Deterministic Regular Expressions with Counters is Decidable

Markus Latte* and Matthias Niewerth*

Universität Bayreuth

Abstract. We show that weakly deterministic regular expressions with counters (WDREs) —as they are used in XML Schema— are at most exponentially larger than equivalent DFAs. As a consequence, the problem, whether a given DFA is equivalent to any WDRE, is decidable in EXPSPACE.

1 Introduction

Deterministic or one-unambiguous regular expressions have been a topic of research since they were formally defined by Brüggemann-Klein and Wood in order to investigate a requirement in the ISO standard for the Standard Generalized Markup Language (SGML), where they were introduced to ensure efficient parsing. XML Schema, the current industry standard schema language for XML, also requires that the regular expressions defining its content models are deterministic. More precisely, an XML Schema is essentially a regular tree grammar in which right-hand sides of rules are *weakly deterministic regular expressions with counting (WDREs)* [6]. In the light that XML Schema is so wide-spread, it is surprising that WDREs are not well-understood. It is known that WDREs cannot define all regular word languages [6] but it is not yet known if it can be decided whether a given regular word language can be defined by a WDRE. In this paper, we prove that the latter problem is decidable in EXPSPACE.

Related Work Brüggemann-Klein and Wood [3] first described an algorithm to decide whether there exist a deterministic regular expression — without counters but with Kleene stars — (DRE) for a given regular language. There has been further research on this BKW algorithm in [5,14]. Gelade et al. analysed weakly and strongly deterministic regular expressions [6]. Bex et al. investigated algorithms for approximating regular languages by DREs [1]. Kilpeläinen and Tuhkanen provide PTIME algorithms for the membership problem of WDREs [10] and the problem checking whether a given regular expression with counters is weakly deterministic [11]. The latter complexity has been improved to linear time in [9]. Chen and Lu provide efficient algorithms checking whether an expression with counting is strongly deterministic [4]. There has been previous research

* Supported by grant number MA 4938/21 of the Deutsche Forschungsgemeinschaft (Emmy Noether Nachwuchsgruppe).

on descriptional complexity of DREs [12]. Dag Hovland investigated efficient matching algorithms for (deterministic) regular expressions with counters using automata with counters [7,8].

Structure of the Proof We limit the size of WDREs in several stages.

- Define some normal form that normalizes the structure of immediately nested counters. (§ 3)
- Partition all nodes in the parse tree of an expression in looping subexpressions and non-looping subexpressions. (§ 4)
- Show that a leaf-directed path in the parse tree consisting only of looping subexpression has at most polynomial length in the alphabet size when counting each block of immediately nested counters as length one. (§ 4)
- Show that each leaf-directed path in the parse tree has at most polynomially many non-looping subexpressions in the size of a minimal DFA. (§ 5)
- Define iterators to recover in a DFA the effective counters of a looping subexpression. (§ 6)
- Limit the number of immediately nested counters logarithmically and the upper bounds of each counter linearly in the size of a minimal DFA. (§ 7)

In § 8, we connect all partial results to show that a minimal WDRE is at most exponentially larger than an equivalent DFA and in § 9, we show a corresponding exponential lower bound for the size of WDREs.

2 Preliminaries

A *language* is a (possibly infinite) set of strings over a finite alphabet Σ. For any language L, we define $\mathsf{first}(L) = \{a \in \Sigma \mid \exists w \in \Sigma^*.\ aw \in L\}$ and $\mathsf{followlast}(L) = \{a \in \Sigma \mid \exists v \in L, w \in \Sigma^*.\ vaw \in L\}$. The *left quotient* of a language L by a word u is $u^{-1}L := \{w \mid uw \in L\}$ and by a language U is $U^{-1}L := \bigcup_{u\in U} u^{-1}L$. For all $N \subseteq \mathbb{N}$, let L^N abbreviate $\bigcup_{n\in N} L^n$ where L^n is the n-fold concatenation.

A *(deterministic, finite) automaton* (or *DFA*) $\mathcal{A}$ is a tuple (Q, δ, q_0, F), where Q is a set of states, $\delta : Q \times \Sigma \rightharpoonup Q$ is a partial transition function, q_0 is the initial state, and F is the set of final states. By δ^* we denote the extension of δ to strings (and languages), i.e., $\delta^*(q, w)$ is the state that can be reached from q by reading w. The *language* of $\mathcal{A}$ is $L(\mathcal{A}) := \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$. We define the *size* of an automaton to be the number of its states.

We let $\mathbb{N}$ and $\mathbb{N}_+$ denote the natural numbers with and without zero. Addition and multiplication are one-sidedly expanded on subsets of $\mathbb{N}$ and understood pointwise. We set $I \otimes J := \left\{ \sum_{k=1}^{j} i_k \mid j \in J \text{ and } i_k \in I \text{ for } 1 \le k \le j \right\}$ as the *product* of two subsets I, J of $\mathbb{N}$. Because $(2^{\mathbb{N}}, \otimes, \{1\})$ is a monoid, the product is canonically extended to lists, written as $\bigotimes$. The product $\otimes$ is not commutative, as $[a \dots b] \otimes \{c\} = [ac \dots bc] \neq c[a \dots b] = \{c\} \otimes [a \dots b]$ if $a < b$ and $c > 1$.

We define $\mathbb{C}_{<\omega} := \left\{ [c^- \dots c^+] \mid (c^-, c^+) \in \mathbb{N}^2 \setminus \{(0,0)\} \text{ and } c^- \le c^+ \right\}$, $\mathbb{C}_\omega := \{\mathbb{N}, \mathbb{N}_+\}$, and $\mathbb{C} := \mathbb{C}_{<\omega} \cup \mathbb{C}_\omega$, as the sets of finite, infinite and all *counters*. Each counter is just a subset of $\mathbb{N}$. Let $C \in \mathbb{C}$ be a counter. Then we use C^-

and C^+ to denote $\min C$ and $\sup C$, respectively and $C^\ominus$ to denote $\max(1, C^-)$. If $C^- = C^+$, we may denote the counter by the singleton set $\{C^-\}$. Lists of (finite) counters live in $[\mathbb{C}]$ and $[\mathbb{C}_{<\omega}]$, respectively. The concatenation symbol for counters is "$\cdot$" or is omitted.

The *regular expressions* over Σ are defined as follows: ε, $\emptyset$ and every Σ-symbol is a regular expression; and whenever r and s are regular expressions, then so are (rs), $(r + s)$, and $(r)^C$, where C is a counter. For readability, we usually omit parentheses. Sometimes we write $r \cdot s$ instead of rs to emphasize that two expressions are concatenated. As syntactic sugar, $r^{\vec{C}}$ denotes the expression $(\ldots((r)^{C_1})^{C_2}\ldots)^{C_N}$ for any regular expression r and list $\vec{C} = C_1 \cdot \ldots \cdot C_N \in [\mathbb{C}]$ of counters with $C_i \in \mathbb{C}$ for each i. The language of a regular expression r, denoted by $L(r)$, is defined as usual. For instance, $L(r^C) := L(r)^C$ for $C \in \mathbb{C}$. Thus, the usual Kleene star is a synonym for the counter $\mathbb{N}$ in our setting. Although we do not explicitly consider the left-bounded interval $[c\ldots]$ as a counter for $c \geq 2$, the expression $r^{[c\ldots]}$ can be emulated by $(r^{\mathbb{N}+})^{[c\ldots c]}$. For each regular expression r, we let $\mathsf{first}(r) = \mathsf{first}(L(r))$ and $\mathsf{followlast}(r) = \mathsf{followlast}(L(r))$.

Intuitively, a regular expression is weakly deterministic if the following holds. When reading the input string from left to right, the expression always allows to match each symbol of that string uniquely against a position in the expression, without looking ahead. However, which counter is incremented might be ambiguous — in contrast to strong determinism [6].

Formally, let $\bar{r}$ be the regular expression obtained from r by annotating every alphabet symbol with its position in the expression. For example, for $r = b^*a(b^*a)^*$ we have $\bar{r} = b_1^*a_2(b_3^*a_4)^*$. A regular expression r is *weakly deterministic* if for all $a \in \Sigma$ and all wa_iv, $wa_jv' \in L(\bar{r})$ the annotations i and j are equal. We denote the class of weakly deterministic regular expressions with counters by WDRE.

The expression $(a + b)^*a$ is not deterministic as already the first symbol in the string aaa could be matched by either the first or the second a in the expression. The equivalent expression $b^*a(b^*a)^*$, on the other hand, is deterministic. Brüggemann-Klein and Wood showed that not every (non-deterministic) regular expression is equivalent to a deterministic one [3]. Thus, semantically, not every regular language can be defined with a deterministic regular expression.

A WDRE r is *reentrant* iff whenever an alphabet symbol occurs in $\mathsf{first}(r)$ and in $\mathsf{followlast}(r)$ then both occurrences are justified by the same position in r. Intuitively, a reentrant WDRE is allowed to occur under a counter. It is easy to see that a WDRE r is reentrant iff r^* is a WDRE. We denote the set of all reentrant WDREs by WDRE^Ω. The *size* of a WDRE r denoted by $|r|$ is the number of nodes of its parse tree plus the sum of the logarithms of the upper bounds of all its finite counters.

3 Normal Form

In this section, we will give a normal form for WDREs based on the following rewrite rules.

Lemma 1. *Let $a, b \in \mathbb{N}$ such that $a \leq b$, let $c, d \in \mathbb{N}_+$, $C_\omega \in \mathbb{C}_\omega$, and $C \in \mathbb{C}$.*

$$\begin{array}{lcllcl} [1\dots c] \otimes [0\dots d] & = & [0\dots cd] & [0\dots c] \otimes [a\dots b] & = & [0\dots bc] \\ [1\dots c] \otimes [1\dots d] & = & [1\dots cd] & [0\dots c] \otimes C_\omega & = & \mathbb{N} \\ [1\dots c] \otimes C_\omega & = & C_\omega & \mathbb{N} \otimes C & = & \mathbb{N} \\ \mathbb{N}_+ \otimes C & = & [C^- \dots 2C^\ominus - 1] \otimes \mathbb{N}_+ & & & \end{array} \tag{1}$$

A challenge to our goal is that the operation $\otimes$ does not preserve intervals in general as $[5\dots6] \otimes [3\dots4] = \{15, \dots, 18,\ \ 20, \dots, 24\}$.

Lemma 2. *Let $r \in \mathsf{WDRE}$ and $\vec{C}, \vec{D} \in [\mathbb{C}]$. $L(r^{\vec{C}}) = L(r^{\vec{D}})$, if $\bigotimes \vec{C} = \bigotimes \vec{D}$.*

Definition 3. *Let r be a WDRE. The expression r is in normal form, iff the following conditions are true for every subexpression $s^{\vec{C}}$ with $\vec{C} = C_1 \cdot \ldots \cdot C_N$ and every $i \leq N$:*

$$\begin{array}{cc} \varepsilon \in L(s) \to 0 \in C_1 & 0 \in C_i \to i = N \\ 1 \in C_i \ \&\ i < N \to 1 \notin C_{i+1} & C_i \in \mathbb{C}_\omega \to i = N \end{array}$$

Furthermore, $\emptyset$ occurs in a WDRE r, iff $r = \emptyset$, ε occurs in a WDRE r, iff $r = \varepsilon$ and the counter $\{1\}$ does not occur.

As for the first condition, the empty words allows to lower C_1^-. The remaining conditions can be achieved by the rewriting rules from Lem. 1 via Lem. 2. Furthermore, it is well known that we need $\emptyset$ only to represent the empty language. We can get rid of ε by replacing ε^C with ε, $\varepsilon \cdot s$ and $s \cdot \varepsilon$ with s, and $\varepsilon + s$ and $s + \varepsilon$ with $s^{[0\dots1]}$. We note that expressions in normal form can be slightly larger than minimal expressions, as the application of (1) can add at most one to the size for each counter above some $\mathbb{N}_+$ counter.

From now on, all considered WDREs are implicitly assumed to be in normal form. If we require some WDRE to be minimal, we mean a minimal WDRE among all WDREs in normal form.

4 Looping Subexpression

In this section, we define looping subexpressions and limit how many looping subexpressions can be nested into each other.

Definition 4. *The relation $\_ \circlearrowleft^- \_$ is inductively defined as a subset of $\mathsf{WDRE}^\Omega \times [\mathbb{C}] \times \mathsf{WDRE}$.*

$$\mathsf{Loop}_0\ \frac{}{r \circlearrowleft^\varepsilon r} \qquad \mathsf{Loop}_+\ \frac{s \circlearrowleft^{\vec{C}} r_i \qquad i \in \{0,1\}}{s \circlearrowleft^{\vec{C}} r_0 + r_1}$$

$$\mathsf{Loop}_{\mathsf{ctr}}\ \frac{s \circlearrowleft^{\vec{C}} r}{s \circlearrowleft^{\vec{C} \cdot C} r^C} \qquad \mathsf{Loop}_\bullet\ \frac{s \circlearrowleft^{\vec{C}} r_i \qquad \varepsilon \in L(r_{1-i}) \qquad i \in \{0,1\}}{s \circlearrowleft^{\vec{C}} r_0 r_1}$$

Informally, $s \circlearrowleft^{\vec{C}} r$ states that s occurs under the counters $\vec{C}$ in r and that it is possible to reenter s silently, i.e., without parsing the hypothetical input word for r any further. A subexpression s is looping *(in r), denoted by $s \circlearrowleft r$, if $s \circlearrowleft^{\vec{C}} r$ for some $\vec{C} \in [\mathbb{C}]$.*

We emphasize that $s \circlearrowleft^{\vec{C}} r$ does not imply that $s^{\vec{C}}$ is a subexpression of r, as the notation explicitly ignores some side branches in the parse tree. The negation of $\circlearrowleft$ is denoted by $\not\circlearrowleft$.

Example 5. *In $(a^{[2\dots3]}b^{[0\dots1]})^*$, the subexpressions a, $a^{[2\dots3]}$, and $a^{[2\dots3]}b^{[0\dots1]}$ are looping while $b^{[0\dots1]}$ is not. Moreover, b is looping in $b^{[0\dots1]}$ although the counter exposes every reenter as pointless.*

To restrict the maximal length of paths containing only looping subexpressions, we use the measure $\mu : 2^{\Sigma^*} \to 2^{\Sigma} \times 2^{\{\varepsilon\}} \times 2^{\Sigma}$ defined as follows.

$$L \quad \mapsto \quad \big(\, \mathsf{first}(L),\ L \cap \{\varepsilon\},\ \mathsf{followlast}(L) \,\big)$$

The implicit order is the lexicographic order over the inclusion where the left position is the most significant one. The measure is extended to regular expressions by considering their languages. For example, $a \cdot b^*$ is smaller than a^* although their $\mathsf{followlast}$-sets are incomparable.

Lemma 6. *Concerning the right argument of $\_ \circlearrowleft^- \_$, the rule $\mathsf{Loop}_{\mathsf{ctr}}$ decreases the measure weakly, while the rules Loop_+ and $\mathsf{Loop}_\bullet$ decrease the measure strictly. The rules are read upwards.*

Proof. We use the notation as stated in the rules. With $\dot\cup$ we denote the union of two disjoint sets.

Rule $\mathsf{Loop}_{\mathsf{ctr}}$: We have $\mathsf{first}(r^C) = \mathsf{first}(r)$, $\mathsf{followlast}(r^C) \supseteq \mathsf{followlast}(r)$ and that $\varepsilon \in L(r)$ implies $\varepsilon \in L(r^C)$.

Rule Loop_+: Because $r_0{+}r_1 \in \mathsf{WDRE}$, $\mathsf{first}(r_0{+}r_1) = \mathsf{first}(r_0)\dot\cup\mathsf{first}(r_1)$. Assume that $\mathsf{first}(r_i) = \mathsf{first}(r_0 + r_1)$. Then, $\mathsf{first}(r_{1-i}) = \emptyset$, and thus $L(r_{1-i}) \subseteq \{\varepsilon\}$. However, the normal form excludes this situation.

Rule $\mathsf{Loop}_\bullet$: The side condition "$\varepsilon \in L(r_{1-i})$" entails that $\varepsilon \in L(r_0r_1)$ iff $\varepsilon \in L(r_i)$. Moreover, the weak determinism of r_0r_1 entails that

$$\mathsf{first}(r_0r_1) \quad = \quad \mathsf{first}(r_0) \quad \underbrace{\dot\cup \quad \mathsf{first}(r_1)}_{\text{iff } \varepsilon \in L(r_0)}\,, \qquad \text{and}$$

$$\mathsf{followlast}(r_0r_1) \quad = \quad \underbrace{\Big(\mathsf{first}(r_1) \quad \dot\cup \quad \mathsf{followlast}(r_0)\Big) \ \cup}_{\text{iff } \varepsilon \in L(r_1)} \quad \mathsf{followlast}(r_1)\,,$$

and is also is responsible for the disjoint unions. So far, the measure is weakly decreasing. Because of the mentioned side condition and because r is in normal form, $L(r_{1-i})$ contains the empty word and a further word. Thus, $\mathsf{first}(r_{1-i}) \neq \emptyset$. Therefore, if $i = 0$ then $\mathsf{followlast}(r_0r_1) \supsetneq \mathsf{followlast}(r_0)$, and otherwise $\mathsf{first}(r_0r_1) \supsetneq \mathsf{first}(r_1)$. □

Theorem 7. *In any expression, the length of any path of therein looping subexpressions is bounded by $2|\Sigma|^2$ if every maximal group of immediately nested counters is counted as one.*

Proof. By Lem. 6 and the definition of μ. □

5 Non-Looping Subexpression

The following lemma will allow us to characterize languages of non-looping subexpressions by means of (simple) DFA operations that do not increase the size of an equivalent DFA.

Lemma 8. *If $s \not\circlearrowleft r$ and u is a word that leads parsing in r to s, then*

$$u^{-1} R \quad \cap \quad \big(\mathsf{first}(SZ)\Sigma^* \cup (SZ \cap \{\varepsilon\})\big) \quad = \quad S \cdot Z \tag{2}$$

where R stands for $L(r)$, S for $L(s)$ and $Z := \big(S^{-1} u^{-1} R\big) \setminus \big(\mathsf{followlast}(S)\Sigma^*\big)$.

Proof. Let Y denote the language on the left side of (2).

Direction $\subseteq$. We silently use that r and s are weakly deterministic. Let $x \in Y$. Since $ux \in R$, the parsing of $\mathsf{first}(x)$ is handled by s or $x = \varepsilon \in S$. Thus, $x = yz$ for some $y \in S$ and some $z \notin \mathsf{followlast}(S)\Sigma^*$. Therefore, $x = y \cdot y^{-1} u^{-1} (ux) \in S \cdot \big((S^{-1} u^{-1} R) \setminus (\mathsf{followlast}(S)\Sigma^*)\big) = S \cdot Z$.

Direction $\supseteq$. To show $u^{-1}R \supseteq SZ$, let $s_0, s_1 \in S$ and $z \in Z$ such that $us_1z \in R$. The parsing of the factor s_1 in us_1z does not require the support of any rootward counter, because s is not looping in r. Therefore, s_1 can be replaced by any word in S, for instance by s_0. In other words, $s_0z \in u^{-1} R$. □

Theorem 9. *Let r be a minimal WDRE, and let $\mathcal{A}$ be an equivalent DFA with n states. Every leaf-directed path p in r hosts at most $n^3(|\Sigma|+1)^2$ non-looping subexpressions. For any non-looping subexpression s on p, there exists a DFA with at most $n+1$ states.*

Proof sketch. The length of paths that only have disjunctions and concatenations and take the right branch of each concatenation is limited by $n(|\Sigma|+1)$. The basic idea is, that these paths can contain only n concatenations, as the language of the right side of a concatenation can be constructed in the automaton by just choosing a different initial state. Disjunctions restrict the set of first symbols, i.e., after each concatenation, there can be at most $|\Sigma|$ consecutive disjunctions.

Finally, whenever a path leading to a non-looping subexpression has some counter or uses the left branch of a concatenation, then Lem. 8 entails that the corresponding DFA essentially has less transitions than the DFA for the subexpression at the beginning of the path. Indeed, the left quotient and the intersection in (2) can be read as local modifications of R's DFA. To unravel S from the right-hand side of (2), we remove those transitions, that leave some state reached after reading some word from S and using a symbol not in $\mathsf{followlast}(S)$. As Z is nonempty in this case, such transitions have to exist.

Each automaton construction does not change the set of states. The only exception is caused by the reduction of the first-set in the case of disjunctions and of Lem. 8: the initial state is duplicated but without its incoming transitions. □

6 Iterators

In this section we will introduce iterators to connect the size of finite automata equivalent to some WDRE r such that $s \circlearrowleft^{\vec{C}} r$ with the size of automata accepting a unary representation of $\bigotimes \vec{C}$. This allows us in the next section to analyse the counters in $\vec{C}$ without looking at the concrete language accepted by s.

Definition 10. *An* iterator for $r \in \mathsf{WDRE}^{\Omega}$ *is a pair* (x_0, x_1) *of words such that*

$$(x_0x_1)^{\lfloor k/2 \rfloor}\, x_0^{k \bmod 2} \in L(r)^{\ell} \qquad \textit{iff} \qquad k = \ell \quad \textit{or} \quad \varepsilon \in L(r) \textit{ and } k \leq \ell$$

for all $k, \ell \in \mathbb{N}$.

Intuitively, reading both words of an iterator alternatively requires a hypothetical counter at the top of r, as the parsing cannot be continued within r.

Lemma 11. *Let* $r \in \mathsf{WDRE}^{\Omega}$ *such that the topmost operator of* r *is not a counter. Then there is an iterator for* r.

Proof sketch. If r is a letter, then we use (r, r) as iterator. If $r = r_0 + r_1$, we use (v_0, v_1) as iterator. And if $r = r_0 r_1$, an iterator is $(v_0 v_1, v_0 v_1)$. In the last two cases, v_i is a shortest word in $L(r_i) \setminus \{\varepsilon\}$. □

The next technical lemma will be used to lift the iterator property from s to $r^{\vec{C}}$ whenever $s \circlearrowleft^{\vec{C}} r$. In the following theorem, we use the lemma to limit the size of DFAs accepting exactly the words of lengths from $\bigotimes \vec{C}$.

Lemma 12. *If* $s \circlearrowleft^{\vec{C}} r$ *then* $L(r) \cap L(s^*) \subseteq L(s^{\vec{C}})$.

Proof. The more general statement "$L(r^{\vec{D}}) \cap L(s^*) \subseteq L(s^{\vec{C}\vec{D}})$ for all non-empty $\vec{D} \in [\mathbb{C}]$ such that $r^{\vec{D}}$ is in normal form" implies the claim with $[1 \dots 1]$ or $[0 \dots 1]$ as D depending on whether $\varepsilon \in L(r)$. For Loop_0, the list $\vec{C}$ is empty and $s = r$, yielding the statement. For the other rules, we prefer the notation of Def. 4.

Rule $\mathsf{Loop}_{\mathsf{ctr}}$: The induction hypothesis is instantiated with $C \cdot \vec{D}$ as $\vec{D}$.

Rule $\mathsf{Loop}_{\bullet}$: Let $v \in L\big((r_0 r_1)^{\vec{D}}\big) \cap L(s^*)$. Because $v \in L(s^*)$ and because $(r_0 r_1)^{\vec{D}}$ is deterministic, the matching of v against $(r_0 r_1)^{\vec{D}}$ considers no leafs outside the parse tree of s, i.e., r_{1-i} is always matched against the empty word. Therefore, $L\big((r_0 r_1)^{\vec{D}}\big) \cap L(s^*) \subseteq L(r_i^{\vec{D}}) \cap L(s^*)$. The induction hypothesis yields the statement.

Rule Loop_{+}: Let $v \in L\big((r_0 + r_1)^{\vec{D}}\big) \cap L(s^*)$. As with $\mathsf{Loop}_{\bullet}$, the matching of v against $(r_0 + r_1)^{\vec{D}}$ considers no leafs outside the parse tree of s. Thus, the side branch r_{1-i} contributes empty words at the most. Because $\vec{D}$ is not empty and since $(r_0 + r_1)^{\vec{D}}$ is in normal form, the existence of ε-contributions entails that $\bigotimes \vec{D}$ is downward closed. Hence, the omitted ε-contributions can be simulated by a smaller instance in $\bigotimes \vec{D}$. Thus, $v \in L\big(r_i^{\vec{D}}\big) \cap L(s^*)$. The induction hypothesis yields $v \in L(s^{\vec{C}\vec{D}})$. □

Let $\mathbb{1}$ be some fixed letter. For each $n \in \mathbb{N}$, $\langle n \rangle$ stands for $\mathbb{1}^n$. The operation is extended to sets. A DFA *expresses* a subset N of $\mathbb{N}$ iff its language is $\langle N \rangle$.

Theorem 13. *Let s, $\vec{C}$ and r be such that $s \leftarrow\!\!\circlearrowleft^{\vec{C}} r$ and s does not have a counter as topmost operation. If there is a DFA with n states for the language $L(r)$, then there is a DFA with $2n+1$ states for $\langle \bigotimes \vec{C} \rangle$.*

Proof. The statement is trivial for $\vec{C} = \varepsilon$, therefore we assume $\vec{C} \neq \varepsilon$. Due to Lem. 11, the expression s has an iterator (x_0, x_1). Let $g\colon \{\mathbb{1}\}^* \to \Sigma^*$ be the function $\mathbb{1}^k \mapsto (x_0\ x_1)^{\lfloor k/2 \rfloor}\ x_0^{k \bmod 2}$. If $\varepsilon \in L(s)$, then the normal form entails that each counter in $\vec{C}$ starts with 0 and thus $N := \bigotimes \vec{C}$ is downward closed. Independently of whether $\varepsilon \in L(s)$, Def. 10 therefore comes down to the statement: $g(\mathbb{1}^k) \in L(s)^N$ iff $k \in N$, for all $k \in \mathbb{N}$. A simple induction on $s \leftarrow\!\!\circlearrowleft^{\vec{C}} r$ yields $L(s^{\vec{C}}) \subseteq L(r)$, because the additional side branches do not harm. Since $g(\mathbb{1}^k) \in L(s)^k \subseteq L(s^*)$ due to Def. 10, we obtain from Lem. 12 that $g(\mathbb{1}^k) \in L(r)$ implies $g(\mathbb{1}^k) \in L(s^{\vec{C}})$ for $k \in \mathbb{N}$. All together, $g^{-1}\big(L(r)\big) = g^{-1}\big(L(s)^{\vec{C}}\big) = \langle \bigotimes \vec{C} \rangle = \langle N \rangle$, where g^{-1} denotes the pre-image under g. The language $g^{-1}(L(r))$ is expressible by a $2n$-state DFA, which can be shown by some kind of product construction of the DFA for $L(r)$ with the two-state DFA keeping track whether the next string should be x_0 or x_1. □

7 Upper Bounds for Counters

In this section we give an upper bound on the number and values of counters of a minimal WDRE r based on the size n of an equivalent minimal DFA. We show that the upper bound of each finite counter is bounded linearly in n and that the number of immediately nested counters is bounded logarithmically in n. The former is established by showing, that each "large" upper bound can be replaced by a smaller one without changing the language.

We define $\hbar : [\mathbb{C}] \to \mathbb{N}$ as $\hbar(C_1 \cdot \ldots \cdot C_N) := \prod_{i \leq N} C_i^{\ominus}$. We show in a series of technical lemmas, that if $s^C \leftarrow\!\!\circlearrowleft^{\vec{C}} r$ then $\hbar(\vec{C})$ iterations of C can be absorbed by the counters in $\vec{C}$. This will allow us to bound C linearly in $\hbar(\vec{C})$, while we show that $\hbar(\vec{C})$ is itself bounded linearly in the minimal DFA equivalent to r.

Lemma 14. *Let $C \in \mathbb{C}_{<\omega}$, and let $c^+, h \in \mathbb{N}_+$ such that $c^+ \geq C^-(1+h)$, then $C \subseteq [C^- \ldots c^+] \otimes \big(1 + h\mathbb{N}\big)$.*

Proof. Let $n \in \mathbb{N}$, then $c^+(1+hn) \geq C^-(1+h)(1+hn) \geq C^-\big(1 + h(n+1)\big)$, and thus the $(1+hn)$th and the $(1+h(n+1))$th incarnation of $[C^- \ldots c^+]$ are overlapping. Because $c^+ > 0$, the maximal number representable by the jth incarnation of $[C^- \ldots c^+]$ is strictly growing with j. Therefore, the set $[C^- \ldots c^+] \otimes \big(1 + h\mathbb{N}\big)$ covers each number from C^- onwards. □

Lemma 15. *Let $C \in \mathbb{C}$ and $h \in \mathbb{N}$. Then, $\big(1 + C^{\ominus} h\mathbb{N}\big) \otimes C \ \subseteq\ C \otimes \big(1 + h\mathbb{N}\big)$.*

Proof. Let $c \in C$ and $n_1, \ldots, n_c, h \in \mathbb{N}$.

$$\sum_{i \le c} 1 + C^{\ominus} h n_i \;=\; c + C^{\ominus} h \sum_{i \le c} n_i \;\in\; C \otimes \Big\{1 + h \sum_{i \le c} n_i\Big\} \;\subseteq\; C \otimes (1 + h\mathbb{N})$$

where "$\in$" holds because $\{c, C^{\ominus}\} \subseteq C$, and "$\subseteq$" because of $\otimes$'s monotonicity. □

The subsequent statements until Lem. 18 assume that each expression determines its position within its hosting expression. In this context, $r[s \leftarrow s']$ denotes the substitution of (the position of) s with s' in an expression r.

Lemma 16. *Let $r, s, s' \in$ WDRE and $\vec{C}, \vec{D} \in [\mathbb{C}]$ such that $s \circlearrowleft^{\vec{C}} r$. Then $L(s) \subseteq L(s')^{1+\hbar(\vec{C}\vec{D})\mathbb{N}}$ implies $L(r) \subseteq L(r[s \leftarrow s'])^{1+\hbar(\vec{D})\mathbb{N}}$.*

Proof. Induction on $s \circlearrowleft^{\vec{C}} r$ where $\vec{D}$ is quantified internally. The list $\vec{D}$ names hypothetical counter at the top of r. In the case of Loop_0, the list $\vec{C}$ is empty, and $s = r$. For the remaining rules, we prefer the notation of Def. 4.

Rule Loop_+:

$$\begin{array}{lcll} L(r) & \subseteq & L(r_i + r_{1-i}) & \\ & \subseteq & L\big(r_i[s \leftarrow s']\big)^{1+\hbar(\vec{D})\mathbb{N}} \;\cup\; L(r_{1-i}) & \text{(sem. of } + \text{ and IH)} \\ & \subseteq & \big(L(r_i[s \leftarrow s']) \;\cup\; L(r_{1-i})\big)^{1+\hbar(\vec{D})\mathbb{N}} & \text{(monotonicity of } \_^{1+\_}\text{)} \\ & \subseteq & L(r[s \leftarrow s'])^{1+\hbar(\vec{D})\mathbb{N}} & \text{(sem. of } + \text{ and } \_[\_ \leftarrow \_]\text{)} \end{array}$$

Rule $\mathsf{Loop}_\bullet$: The argument is analogous to Loop_+'s case but with additional use of $\varepsilon \in L(r_{1-i})$.

Rule $\mathsf{Loop}_{\mathsf{ctr}}$: As the rule addresses the counters $\vec{C}C$ instead of $\vec{C}$, the aimed implication is adjusted accordingly. The induction hypothesis for $C\vec{D}$ as $\vec{D}$ entails that $L(r) \subseteq L(r[s \leftarrow s'])^{1+\hbar(C\cdot\vec{D})\mathbb{N}}$. With help of Lem. 15 and the definition of $\hbar$, we get $(1 + \hbar(C \cdot \vec{D})\mathbb{N}) \otimes C \subseteq C \otimes (1 + \hbar(\vec{D})\mathbb{N})$. Finally, the substitution $\_[s \leftarrow s']$ commutes with $\_^C$. □

Lemma 17. *Let $r, s, s' \in$ WDRE, let $\vec{C} \in [\mathbb{C}]$, let and $C_\omega \in \mathbb{C}_\omega$ such that $s \circlearrowleft^{\vec{C}\cdot C_\omega} r$. Then $L(s) \subseteq L(s')^{1+\hbar(\vec{C})\mathbb{N}}$ implies $L(r) \subseteq L(r[s \leftarrow s'])$.*

Proof. By induction on "$s \circlearrowleft^{\vec{C}\cdot C_\omega} r$". Eventually the rule $\mathsf{Loop}_{\mathsf{ctr}}$ justifies the statement $s \circlearrowleft^{\vec{C}\cdot C_\omega} r_0^{C_\omega}$ with $s \circlearrowleft^{\vec{C}} r_0$ for some r_0. For an empty list $\vec{D}$, Lem. 16 entails that $L(r_0) \subseteq L(r_0[s \leftarrow s'])^{1+\mathbb{N}}$. Because $(1 + \mathbb{N}) \otimes C_\omega \subseteq \mathbb{N}_+ \otimes C_\omega \subseteq C_\omega$ and because $\_^{C_\omega}$ commutes with the substitution, $L(r_0^{C_\omega}) \subseteq L(r_0^{C_\omega}[s \leftarrow s'])$. □

Lemma 18. *Let $C_0 \in \mathbb{C}_{<\omega}$, $\vec{C} \in [\mathbb{C}]$, and $C_\omega \in \mathbb{C}_\omega$. If $s^{C_0} \circlearrowleft^{\vec{C}\cdot C_\omega} r$ for some minimal WDRE r, then $C_0^+ \;\le\; C_0^{\ominus} \cdot \big(1 + \hbar(\vec{C})\big)$.*

Proof. For the sake of contradiction, assume that $C_0^+ > c^+ := C_0^{\ominus} \cdot \big(1 + \hbar(\vec{C})\big)$. Set $s^- := s^{[C_0^- \ldots c^+]}$ and $r^- := r[s^{C_0} \leftarrow s^-]$. These expressions are weakly deterministic, because $[C_0^- \ldots c^+] \in \mathbb{C}$. Due to monotonicity, $L(r^-) \subseteq L(r)$. For the other direction, Lem. 14 entails that $L(s^{C_0}) \subseteq L(s^-)^{1+\hbar(\vec{C})\mathbb{N}}$. Thus, Lem. 17 yields $L(r) \subseteq L(r^-)$. Therefore, $L(r^-) = L(r)$ although $|r^-| < |r|$. □

The previous restriction of counters is related with the transformation into the star normal form [2]. There, $(s_0 s_1 + s_2)^*$ is rewritten as $(s_0 + s_1 + s_2)^*$ if $\varepsilon \in L(s_0 s_1)$, for instance. One incarnation of $s_0 s_1$ is replaced with two of $s_0 + s_1$ while the Kleene star can absorb arbitrarily many incarnations. Because $\mathbb{C}$ also admits finite intervals, the absorption quantum here depends on the stack of counters under which the expression appears effectively.

Definition 19. *A list $C_1 \cdot \ldots \cdot C_N \in [\mathbb{C}]$ of counters* propagates 0 *iff $0 \in C_i$ together with $i \leq j$ implies $0 \in C_j$ for all $i, j \leq N$.*

Lemma 20. *Let $\vec{C} \cdot \vec{D} \in [\mathbb{C}]$ such that $\vec{C}$ propagates 0. If $\bigotimes(\vec{C} \cdot \vec{D})$ is expressible by an n-state DFA, then $\hbar(\vec{C}) < n$.*

Proof. For every $C \in \mathbb{C}$ and $N \subseteq \mathbb{N}$, $\min\big((C \otimes N) \setminus \{0\}\big) = C^{\ominus} \cdot \max\big(1, \min(N)\big)$. A simple induction using the 0-propagation entails that

$$\min\Big(\bigotimes(\vec{C} \cdot \vec{D}) \setminus \{0\}\Big) \quad = \quad \hbar(\vec{C}) \cdot \min\Big(\bigotimes \vec{D} \setminus \{0\}\Big) \quad \geq \quad \hbar(\vec{C})\,.$$

The number on the left is strictly bounded by $n-1$ because the smallest non-empty word is reachable in the DFA without any loop. □

Lemma 21. *Let $N \in \mathbb{N}$, let $C_1, \ldots, C_N \in \mathbb{C}_{<\omega}$. If $\bigotimes_{i=1}^{N} C_i$ is expressible by an n-state DFA, then $\prod_{i=1}^{N} C_i^+ < n$.*

Proof. Because $\bigotimes_{i=1}^{N} C_i$ is finite, the DFA lacks in loops. □

Theorem 22. *Let $s, t \in$ WDRE, $C \in \mathbb{C}_{<\omega}$, and $\vec{C} \in [\mathbb{C}]$ such that $t^C \leftarrow\!\!\circlearrowleft^{\vec{C}} s$. If s is minimal and $L(s)$ is expressible by an n-state DFA, then $C^+ \leq 4n$.*

Proof. We remove all outermost counters of t. The normal form guarantees that these removed counters are finite. Thus, there is a list $\vec{C}_0 \in [\mathbb{C}_{<\omega}]$ such that $u \leftarrow\!\!\circlearrowleft^{\vec{C}_0} t$ where u denoted the obtained pruned subexpression. By transitivity, $u \leftarrow\!\!\circlearrowleft^{\vec{C}_0 \cdot C \cdot \vec{C}} s$. Because u does not have a counter as topmost operation, Thm. 13 entails that $\bigotimes\big(\vec{C}_0 \cdot C \cdot \vec{C}\big)$ is expressible by a $(2n+1)$-state DFA.

If $\vec{C}$ consists only of finite counters, then Lem. 21 bounds C^+. Otherwise, $\vec{C}$ can be written as $\vec{C}_{<\omega} \cdot C_\omega \cdot \vec{C}_{\omega+1}$ such that $\vec{C}_{<\omega} \in [\mathbb{C}_{<\omega}]$, $C_\omega \in \mathbb{C}_\omega$, and $\vec{C}_{\omega+1} \in [\mathbb{C}]$. The normal form entails that $C\vec{C}_{<\omega}$ propagates 0. By Lem. 18 and Lem. 20, we have $C^+ \leq C^{\ominus} \cdot \big(1 + \hbar(\vec{C}_{<\omega})\big) \leq 2\hbar(C\vec{C}_{<\omega}) \leq 4n$. □

Theorem 23. *Let s, t, $\vec{C}_0$, $\vec{C}_1$ be such that $t^{\vec{C}_0} \leftarrow\!\!\circlearrowleft^{\vec{C}_1} s$ and t does not have a counter as topmost operation. If $L(s)$ is expressible by an n-state DFA, then $|\vec{C}_0| \leq 2\lg(n) + 4$.*

Proof. The normal form ensures that at most the last counter of $\vec{C}_0$ belongs to $\mathbb{C}_\omega$. If so the last counter can be attributed to $\vec{C}_1$. Formally, there are $\vec{C}_2 \in [\mathbb{C}_{<\omega}]$ and $\vec{C}_3 \in [\mathbb{C}]$ such that $\vec{C}_0\vec{C}_1 = \vec{C}_2\vec{C}_3$ and $|\vec{C}_2| \geq |\vec{C}_0| - 1$. The normal form also entails that $\vec{C}_2$ propagates 0. The set $\bigotimes \vec{C}_2 \otimes \bigotimes \vec{C}_3$ is expressible by an

$(2n+1)$-state DFA due to Thm. 13. Thanks to Lem. 20, $\hbar(\vec{C}_2) \leq 2n$. Therefore, the number of counters which deny 0 or 1 is bounded by $\lg(2n)$. In front of, between, and after those counters, at most one other counter appears. Thus, $|\vec{C}_0| \leq |\vec{C}_2| + 1 \leq \big(2\lg(2n)+1\big) + 1 \leq 2\lg(n) + 4$. □

8 Upper Bound

Theorem 24. *Let $\mathcal{A}$ be a DFA with n states such that $L(\mathcal{A})$ is expressible by a WDRE. Then a minimal WDRE for $L(\mathcal{A})$ is of size at most $2^{\mathcal{O}(|\Sigma|^4 n^3 \lg^2(n))}$ and all of its finite counters are bounded by $\mathcal{O}(n)$.*

Proof. We show that the parse tree of a minimal WDRE has depth at most $\mathcal{O}\big(|\Sigma|^4 n^3 \lg(n)\big)$ and the upper bounds of (finite) counters are all bounded by $\mathcal{O}(n)$. As the parse tree is binary, this directly yields the claimed size bound.

Let r be a minimal WDRE in normal form equivalent to $\mathcal{A}$. By Thm. 9, any leaf-directed path in the parse tree of r can host at most $n^3(|\Sigma|+1)^2$ non-looping subexpressions. Furthermore, each non-looping subexpression has an automaton with at most $n+1$ states. By Thm. 23, each block of immediately nested counters has at most $2\lg(n+1)+4$ counters. Combining this with Thm. 7 yields that any leaf-directed path consisting only of looping sub-expressions has length at most $2|\Sigma|^2(2\lg(n+1)+4)$. Altogether, we get that the depth of the parse tree is bounded by $\mathcal{O}\big(|\Sigma|^4 n^3 \lg(n)\big)$. Finally, we can apply Thm. 22 to bound the values of counters. Let C be a finite counter of r, such that $t^C \circlearrowleft^{\vec{C}} s$, where t is the subexpression below C and s is the lowest non-looping subexpression of r above C. Applying Thm. 22 yields that $C^+ \leq 4(n+1)$. □

Corollary 25. *Let L be a regular language. If L is given by a DFA (a regular expression without counters, a regular expression with counters), it can be decided in EXPSPACE (2-EXPSPACE, 3-EXPSPACE), whether there is some WDRE for r.*

Proof. Compute a DFA $\mathcal{A}$ for L with linearly (exponentially, double exponentially) many states. Enumerate [11] each WDRE up to the size bound of Thm. 24 and test whether its language is L: (i) Unravel all counters [6] while ignoring weak determinism. (ii) Test the obtained general regular expression against $\mathcal{A}$. □

9 Lower Bound

We adapt an existing proof showing that WDREs without counters are exponentially larger than minimal DFAs from [13].

Theorem 26. *There exists a family of languages $(L_n)_{n\in\mathbb{N}}$ such that the minimal DFA for L_n has size $\Theta(n)$, and every minimal WDRE for L_n has size $2^{\Omega(n)}$.*

Proof sketch. As [13], we consider the finite languages $L_n = L\big((a+b)^{[0\dots n]} \cdot b\big)$ for every $n \in \mathbb{N}$. The minimal DFA for L_n has $2n+2$ states. By an inductive proof, it can be shown, that the minimal WDRE r_n for the language L_n is of the form $a \cdot r_{n-1} + b \cdot r_{n-1}^{[0\dots 1]}$ which directly proves the assumption. □

The main difference to the proof in [13] is that we have to consider counters in the inductive step. We note that Thm. 26 is independent of our normal form.

10 Conclusion

We have shown both an exponential upper and an exponential lower bound for the size of WDREs in terms of minimal DFA size. This easily gives an EXPSPACE upper bound for the decision problem, given a DFA does there exist an equivalent WDRE, solving an open problem from [6]. However, the complexity of this decision problem is still open, as we only have an NL lower bound that carries over from the problem for expressions without counters [14]. Especially, it is unclear, whether there is an adaption of the BKW algorithm presented in [3] that includes counters.

In [12,13], the descriptional complexity of DREs has been analysed. We believe, that the lower bounds for expression size can be transferred to WDREs, as the language families used in the proofs should not benefit from the use of counters.

References

1. G. J. Bex, W. Gelade, W. Martens, and F. Neven. Simplifying XML Schema: effortless handling of nondeterministic regular expressions. In *ACM SIGMOD*, pages 731–744. ACM, 2009.
2. A. Brüggemann-Klein. Regular expressions into finite automata. *TCS*, 120(2):197–213, 1993.
3. A. Brüggemann-Klein and D. Wood. One-unambiguous regular languages. *Inf. and Comput.*, 142(2):182–206, 1998.
4. H. Chen and P. Lu. Checking determinism of regular expressions with counting. In *DLT*, pages 332–343, 2012.
5. W. Czerwiński, C. David, K. Losemann, and W. Martens. Deciding definability by deterministic regular expressions. In *FOSSACS*, pages 289–304, 2013.
6. W. Gelade, M. Gyssens, and W. Martens. Regular expressions with counting: Weak versus strong determinism. *SIAM J. Comp.*, 41(1):160–190, 2012.
7. D. Hovland. Regular expressions with numerical constraints and automata with counters. In *ICTAC*, pages 231–245. Springer, 2009.
8. D. Hovland. The membership problem for regular expressions with unordered concatenation and numerical constraints. In *LATA*, pages 313–324, 2012.
9. P. Kilpeläinen. Checking determinism of XML schema content models in optimal time. *Inf. Systems*, 36(3):596–617, 2011.
10. P. Kilpeläinen and R. Tuhkanen. Towards efficient implementation of XML schema content models. In *DocEng*, pages 239–241. ACM, 2004.
11. P. Kilpeläinen and R. Tuhkanen. One-unambiguity of regular expressions with numeric occurrence indicators. *Inf. and Comput.*, 205(6):890–916, 2007.
12. K. Losemann, W. Martens, and M. Niewerth. Descriptional complexity of deterministic regular expressions. In *MFCS*, pages 643–654, 2012.
13. K. Losemann, W. Martens, and M. Niewerth. Closure properties and descriptional complexity of deterministic regular expressions. Submitted, 2015.
14. P. Lu, J. Bremer, and H. Chen. Deciding determinism of regular languages. *TOCS*, pages 1–43, 2014.

A Proof of Thm. 9

For two words u and v, $u \sqsubseteq v$ denotes that u is a prefix of v, and $v^{\sqsubseteq} := \{u \mid u \sqsubseteq v\}$. Moreover, $u \sqsubsetneq v$ states that u is a strict prefix of v, that is $u \sqsubseteq v \neq u$.

Lemma 27. *Let r be a minimal WDRE and p be a leaf-directed path in its parse tree consisting only of disjunctions and concatenations, such that p takes the right branch of each concatenation and the language of the left branch of each concatenation contains ε. The length of p is at most $|\Sigma|$.*

Proof. Along the path from r to s the respective set $\mathsf{first}(\_)$ is strictly decreasing w.r.t. inclusion, because the normal forms prevent any children from having the language $\{\varepsilon\}$. □

Lemma 28. *Let r be a minimal WDRE and p be a leaf-directed path starting at the root of its parse tree, such that p consists only of disjunctions and concatenations and p uses the right branch of any concatenation. The length of p is at most $n(|\Sigma| + 1)$, where n is the number of states of the minimal DFA equivalent to r.*

Proof. Let r be a WDRE and s be a subexpression of r such that

- the parent t of s is a concatenation, such that the language of the left branch of t does not contain ε; and
- the path from r to s only contains disjunctions and concatenations and uses the right branch of each concatenation.

Let $\mathcal{A}$ be a minimal DFA equivalent to r and u be a string that leads parsing in r to s. We construct a DFA for s by changing the initial state of $\mathcal{A}$ to the state reached after reading u. The final states remain because the path neither turn left at a concatenation nor pass a counter.

As there are only n states in $\mathcal{A}$, there are at most n different automata, which can be created by only changing the initial state. As the length of paths that have only disjunctions and concatenations where the left branch contains ε are bounded by Lem. 27 to $|\Sigma|$, we get the lemma statement. □

In the following lemma, $\mathrm{size}_{\frown}(\mathcal{A})$ denotes the number of those transitions of $\mathcal{A}$ which leave some state with at least one incoming transition, i.e., transitions leaving the initial state are not counted, if the initial state has no incoming transition.

Lemma 29. *If s is a non-looping subexpression in a minimal WDRE r and $\mathcal{A}$ is a DFA for $L(r)$, then there exists a DFA $\mathcal{B}$ equivalent to s such that $\mathrm{size}_{\frown}(\mathcal{B}) \leq \mathrm{size}_{\frown}(\mathcal{A})$. Let p be the path from r to s. If p contains at least one counter, or p takes the left outgoing edge of some concatenation, then $\mathrm{size}_{\frown}(\mathcal{B}) < \mathrm{size}_{\frown}(\mathcal{A})$.*

Proof. Basically, $\mathcal{A}$ is a blueprint for the claimed DFA $\mathcal{B}$. Let u and Z be as in Lem. 8 and let Y denote the language on the left side of (2). Let q_0 be the initial state of $\mathcal{A}$. We construct an intermediate automaton $\mathcal{A}_Y$ by replacing the initial state q_0 by a copy q_0' of the state q_u reached after reading u from q_0. We keep

only those outgoing transitions of q_0' that are labelled by a symbol from $\mathsf{first}(SZ)$. The state q_0' is final iff $\varepsilon \in SZ$. We note, that $\mathcal{A}_Y$ accepts the language $Y = S \cdot Z$ from Lem. 8. We now construct $\mathcal{B}$ from $\mathcal{A}_Y$ by marking exactly those states as final, that can be reached from q_0' by reading some string of s. From the new final states, we keep only those outgoing transitions, that are labelled by some symbol from $\mathsf{followlast}(s)$.

It is easy to see that $\mathrm{size}_\frown(\mathcal{B}) \leq \mathrm{size}_\frown(\mathcal{A})$. If p contains a counter (minimality forbids the counter $[1 \dots 1]$) or if p takes the left edge in some concatenation, it holds that $Z \neq \{\varepsilon\}$. In this case at least one transition is removed from every final state of $\mathcal{B}$ and therefore $\mathrm{size}_\frown(\mathcal{A}) < \mathrm{size}_\frown(\mathcal{A})$.

It remains to show that $\mathcal{B}$ accepts the correct language, especially that $L(\mathcal{B}) \subseteq L(s)$, as the other direction is trivial by construction of $\mathcal{B}$. Let therefore v be a string from $L(\mathcal{B})$ and let $w \in Z$ be a string such that $vw \in L(r) = L(\mathcal{A}_Y)$. We conclude that such a string w exists as follows: (i) The final state of $\mathcal{B}$ reached by reading v can be reached using some string of $L(s)$ (by definition of $\mathcal{B}$). (ii) Lem. 8 assures that after reading any string from $L(s)$, any string from Z leads to an accepting state of $\mathcal{A}_Y$, as $L(\mathcal{A}_Y) = S \cdot Z$.

Let $v' \sqsubseteq vw$ be a longest prefix such that $v' \in L(s)$ and let $w' = {v'}^{-1} vw$. Towards a contradiction, assume that $v' \sqsubsetneq v$. We let a be the symbol immediately following v' in v. By definition of Z, we have that $a \in \mathsf{first}(Z)$. However $a \in \mathsf{followlast}(s)$ holds by definition of $\mathcal{B}$, as else the state reached after reading v' would have no outgoing a-transition. This is a contradiction to r being a WDRE. Assume now that $v \sqsubsetneq v'$. We let a be the first symbol of w. By definition of $\mathcal{B}$, we can again conclude that $a \in \mathsf{followlast}(s)$. However $a \in \mathsf{first}(Z)$ holds as $w \in Z$. Again, this is a contradiction to r being a WDRE. The only remaining option is that $v' = v$ as desired. We can conclude that $L(\mathcal{B}) = L(s)$. □

Statement of Thm. 9. *Let r be a minimal WDRE, and let $\mathcal{A}$ be an equivalent DFA with n states. Every leaf-directed path p in r hosts at most $n^3(|\Sigma|+1)^2$ non-looping subexpressions. For any non-looping subexpression s on p, there exists a DFA with at most $n+1$ states.*

Proof. Lem. 28 limits the length of each path that only consists of disjunctions and concatenations and uses the right branch of each concatenation by $n(|\Sigma|+1)$. And Lem. 29 limits the number of non-looping subexpressions that are not connected by such a path by $n^2|\Sigma|$. □

B Proof of Lem. 11

For any non-empty word w, we denote the first letter by $\mathsf{first}(w)$. WDREs in Lem. 30 and Lem. 31 are understood as marked while their language refers to the unmarked counterpart.

Lemma 30 (Iterating words I)**.** *Let $r \in \mathit{WDRE}^{\bigcirc}$ such that r is a letter, a concatenation with $\varepsilon \notin L(r)$, or a disjunction. Then there are non-empty words*

x_0 and x_1 such that

$$x_i \quad \in \quad \widetilde{L} \tag{3}$$

$$x_i \in \widetilde{L}a\Sigma^* \quad \textit{implies} \quad a\Sigma^* \cap \widetilde{L} = \emptyset, \quad \textit{for all } a \in \Sigma \tag{4}$$

$$\widetilde{L} \cap x_i\mathsf{first}(x_{1-i})\Sigma^* \quad = \quad \emptyset \tag{5}$$

for each $i \in \{0,1\}$ where $\widetilde{L} := L(r) \setminus \{\varepsilon\}$.

Informally, (5) states maximality with respect to prefixes of $(x_i x_{1-i})^\omega$. On the other side, the premise of (4) requires that a strict prefix of x_i belongs to $L(r)$. If we can render the conclusion as false, (4) is minimality statement.

Proof. If r is a letter then take it as x_0 and x_1.

Suppose that $r = r_0 + r_1$. As r is assumed to be in normal form, $L(r_i) \setminus \{\varepsilon\} \neq \emptyset$. Choose $x_i \in \min\big(L(r_i) \setminus \{\varepsilon\}\big)$. Since the first letter of x_i determines the subexpression, $x_i \in \min \widetilde{L}$. Therefore, the premise of (4) is false. As for (5), assume that there was a word $z \in \widetilde{L} \cap x_i\mathsf{first}(x_{1-i})\Sigma^*$. Since r is weakly deterministic, $z \in L(r_i)$, as $\mathsf{first}(z)$ cannot be in $\mathsf{first}(r_{1-i})$. By definition of z, $\mathsf{first}(x_{1-i}) = p$ for some $p \in \mathsf{followlast}(r_i)$. Because r is reentrant and $p \in \mathsf{followlast}(r)$, we know that $p \in \mathsf{first}(r_i)$. This is a contradiction to r being weakly deterministic, as $p \in \mathsf{first}(r_i) \cap \mathsf{first}(r_{1-i})$.

Suppose that $r = r_0 r_1$. As r is assumed to be in normal form, $L(r_i) \setminus \{\varepsilon\} \neq \emptyset$. Choose $y_i \in \min(L(r_i) \setminus \{\varepsilon\})$ and set $x_0 := x_1 := y_0 y_1$. Clearly, $y_0 y_1 \in L(r_0 r_1)$. As for the minimality (4), let v be a strict non-empty prefix of $y_0 y_1$ such that $v \in L(r)$. Because r is weakly deterministic, the first letter of v is parsed in r_0. Since y_0 and y_1 are minimal, $v = y_0$ and $\varepsilon \in L(r_1)$. Hence, $a = \mathsf{first}(y_1)$. For the sake of contradiction, assume the conclusion of (4) was false and let $z \in L(r)$ such that $\mathsf{first}(z) = a$. Thus, the first letter of z can be handled by r_1 due to construction and by r due to assumption. This situation requires that $\varepsilon \in L(r_0)$ because otherwise r wouldn't be weakly deterministic. Therefore, $\varepsilon \in L(r_0 r_1)$ in contradiction to the assumption on r. As for the maximality (5), suppose that $y_0 y_1 y_2 \in L(r)$ for some $y_2 \in \mathsf{first}(y_0)\Sigma^*$. Since r is weakly deterministic and the parsing of $y_0 y_1$ ends in r_1, there is a $p \in \mathsf{followlast}(r_1)$ such that $p = \mathsf{first}(y_2) = \mathsf{first}(y_0)$. Because $\mathsf{followlast}(r_1) \subseteq \mathsf{followlast}(r)$, $\mathsf{first}(y_0) \in \mathsf{first}(r)$, and r is reentrant, we can conclude that $p \in \mathsf{first}(r)$. However, then both $\mathsf{first}(r_0)$ and $\mathsf{first}(r_1)$ contain a position which is labelled with $\mathsf{first}(y_0)$. Hence r would not be weakly deterministic. □

Lemma 31 (Iterating words II)**.** *Let $r \in \mathsf{WDRE}^{\circlearrowright}$ such that r is a concatenation with $\varepsilon \in L(r)$. Then there are non-empty words y_0 and y_1 such that*

$$y_1^{\sqsubseteq} \quad \cap \quad L(r) \quad = \quad \{\varepsilon,\ y_1\}\,, \tag{6}$$

$$(y_0 y_1)^{\sqsubseteq} \quad \cap \quad L(r) \quad = \quad \{\varepsilon,\ y_0,\ y_0 y_1\}\,, \tag{7}$$

$$L(r) \quad \cap \quad y_0 y_1 \mathsf{first}(y_0)\Sigma^* \quad = \quad \emptyset\,, \textit{ and} \tag{8}$$

$$L(r) \quad \cap \quad y_1 \mathsf{first}(y_0)\Sigma^* \quad = \quad \emptyset\,. \tag{9}$$

Proof. Let r be $r_0 r_1$. By assumption, $\varepsilon \in L(r_i)$. As we assume that r is in normal form, the languages $L(r_0)$ and $L(r_1)$ contain a non-empty word. Choose $y_i \in \min(L(r_i) \setminus \{\varepsilon\})$. Because r is weakly deterministic, every non-empty prefix of y_i is handled by r_i, and for every non-empty prefix z of y_1, the word z is handled by r_1 if $y_0 z \in L(r)$. Thus, (6) and (7) hold. As for (8), assume a word $z \in L(r) \cap y_0 y_1 \mathsf{first}(y_0)\Sigma^*$. Since r is weakly deterministic, $\mathsf{first}(y_0) \in \mathsf{first}(r_0) \cap \mathsf{followlast}(r_1)$. This is a contradiction to r being reentrant. The argument for (9) is analogous. □

Statement of Lem. 11. *Let $r \in \mathit{WDRE}^{\bigcirc}$ such that the topmost operator of r is not a counter. Then there is an iterator for r.*

Proof. We distinguish the following cases.

Case: r is a letter, a concatenation with $\varepsilon \notin L(r)$, or a disjunction.
We let x_0 and x_1 be the words given by Lem. 30. The "if"-direction is an immediate consequence of (3). For the other direction, let $\widetilde{L} = L(r) \setminus \{\varepsilon\}$. We strengthen the claim for an induction on k to

$$(x_i x_{1-i})^{\lfloor k/2 \rfloor} x_i^{k \bmod 2} \cap \widetilde{L}^{\ell} \neq \emptyset \quad \text{implies} \quad k = \ell$$

for all $\ell \in \mathbb{N}$ and $i \in \{0, 1\}$. For $k = 0$, the statement is vacuously true, as $\varepsilon \notin \widetilde{L}$. For the induction step, let $k > 0$ be given. As $x_0 \neq \varepsilon \neq x_1$, we have that $\ell > 0$ and we may assume words u_0 and u_1 such that

$$(x_i x_{1-i})^{\lfloor k/2 \rfloor} x_i^{k \bmod 2} \quad = \quad \underbrace{u_0}_{\in \widetilde{L}} \underbrace{u_1}_{\in \widetilde{L}^{\ell-1}} .$$

By induction hypothesis for $k-1$, it suffices to show that $u_0 = x_i$. Clearly, one is the prefix of the other. First, assume that $u_0 \sqsubsetneq x_i$. If $\ell = 1$ then $u_1 = \varepsilon$ and thus $x_i = u_0$, which is a contradiction. Otherwise, $\mathsf{first}(u_1)\Sigma^* \cap \widetilde{L} \neq \emptyset$. However, this situation contradicts (4). Second, assume that $x_i \sqsubsetneq u_0$. Thus, $u_0 \in \widetilde{L} \cap x_i \mathsf{first}(x_{1-i})\Sigma^*$, which is a contradiction to (5).

Case: r is a concatenation with $\varepsilon \in L(r)$.
We let $x_0 = x_1 = y_0 y_1$, where y_0 and y_1 are as in Lem. 31. Because the "if"-direction follows from (6) and (7), we continue with the other direction. It suffices to show for all $k, \ell \in \mathbb{N}$ that

$$k > 0 \text{ and } y_1 (y_0 y_1)^{k-1} \cap L(r)^{\ell} \neq \emptyset \quad \text{implies} \quad k \leq \ell \tag{10}$$

$$(y_0 y_1)^{k} \cap L(r)^{\ell} \neq \emptyset \quad \text{implies} \quad k \leq \ell \tag{11}$$

because the second line is just the claim. We will prove both statement together by an induction over ℓ. For $\ell = 0$, both lines are true, as neither y_0 nor y_1 is empty. For the induction step, let $\ell > 0$ be given. We may assume words u_0, u_1, v_0, v_1 such that

$$(y_0 y_1)^{k} \quad = \quad \underbrace{u_0}_{\in L(r) \setminus \{\varepsilon\}} \underbrace{u_1}_{\in L(r)^{\ell'}} \quad \text{and} \quad y_1 (y_0 y_1)^{k-1} \quad = \quad \underbrace{v_0}_{\in L(r) \setminus \{\varepsilon\}} \underbrace{v_1}_{\in L(r)^{\ell'}}$$

for some $\ell' < \ell$ each. If $u_0 \in \{y_0, y_0y_1\}$ and $v_0 = y_1$, then the induction hypothesis for the respective ℓ' completes the argument. All involved words are already prefix ordered. By (8) and (9), we know that y_0y_1 cannot be a strict prefix of u_0 and that y_1 cannot be a strict prefix of v_0. By (6), we know that v_0 cannot be a strict prefix of y_1. And by (7), $u_0 \in \{y_0, y_0y_1\}$. □

C Proof of Thm. 26

Statement of Thm. 26. *There exists a family of languages $(L_n)_{n\in\mathbb{N}}$ such that the minimal DFA for L_n has size $\Theta(n)$, and every minimal WDRE for L_n has size $2^{\Omega(n)}$.*

Proof. We consider the following languages for every $n \in \mathbb{N}$:

$$L_n = L\big((a+b)^{[0\dots n]} \cdot b\big).$$

The minimal DFA for L_n has $2n+2$ states. It remains to show that a minimal WDRE r_n for a language L_n is at least of size 2^n. In this proof, we use as lower approximation of the size of a WDRE the number of nodes in its parse tree.

The proof is by induction on n. For the induction base case, $n = 0$, the assumption holds because b is a WDRE (of size 1 for L_0).

Now assume that a minimal WDRE r_{n-1} for L_{n-1} is at least of size 2^{n-1} and let r_n be a minimal WDRE for L_n. We show that r_n is of the form $a\cdot r_{n-1} + b\cdot r_{n-1}^{[0\dots 1]}$ which directly proves the assumption.

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

If $\varepsilon \notin L(s_1)$ then we know that $\mathsf{first}(s_1) = \{a, b\}$. Since $b \in L_n$, we have that $\varepsilon \in L(s_2)$ and that every word in $L(s_1)$ ends with b. Let ub be one of the longest words in $L(s_1)$. Since r_n is minimal, we know that $s_2 \neq \varepsilon$. Moreover, the longest word of L_n has size $n+1$ such that $|ub| < n+1$. Then, the longest word vb of $L(s_2)$ is of length $n+1-|ub|$ and $ubvb \in L_n$. Furthermore, we know that $uavb \in L_n$ by the structure of L_n. Because ua and vb are of maximal length for s_1 and s_2 respectively, it follows that $ua \in L(s_1)$. Since we have that $\varepsilon \in L(s_2)$, it holds that $ua \in L(r_n)$ which contradicts the assumption that $L(r_n) = L_n$.

If $\varepsilon \in L(s_1)$ then we know that $\varepsilon \notin L(s_2)$ because $\varepsilon \notin L_n$. Since $b \in L(r_n)$ and r_n is a WDRE, we have that $\mathsf{first}(s_1) = \{a\}$ and $\mathsf{first}(s_2) = \{b\}$. Then, by the definition of L_n, there exists a longest word $bw \in L_n$ with $|bw| = n+1$. Furthermore, it holds that $bw \in L(s_2)$. Because r_n is minimal by assumption it follows that $s_1 \neq \varepsilon$ which directly contradicts that bw is the longest word in L_n. Altogether, this proves that r_n is not a concatenation.

We now assume that $r_n = s^C$, i.e., the topmost operation is a counter $C \in \mathbb{C}_{<\omega}$. We note that as L_n is finite, it is not possible that the topmost operation is a counter from $\mathbb{C}_\omega$. We know that $C^- = 1$, as the shortest word in L_n has length one. Furthermore $C^+ > 1$, as the counter $\{1\}$ cannot appear in a minimal WDRE. As any string in L_n ends with b, any string in $L(s)$ has to end with b. As the

longest string in L_n has size $n+1$, the longest string in $L(s)$ has size at most $\lfloor \frac{n+1}{2} \rfloor$. We can conclude that every string of size more than $\lfloor \frac{n+1}{2} \rfloor$ has at least two b in it, which is a contradiction to the definition of L_n.

The only remaining case is that r_n is a disjunction $s_1 + s_2$. As r_n is a WDRE and $\varepsilon \notin L_n$, the first-set is distributed over the disjuncts s_1 and s_2, say $\mathsf{first}(s_1) = \{a\}$ and $\mathsf{first}(s_2) = \{b\}$. For each $i \in \{1, 2\}$, we follow the left-most branch in s_i to the leaf. Because $L(s_i)$ captures all words in L_n which start $\mathsf{first}(s_i)$, we can exclude concatenations and counters with the same argument as for r_n. Indeed, the contradiction there faces either the length or some not head-placed letter. Moreover, if a disjunction appeared on the way to the leaf, one disjunct would be ε because the first-set is a singleton. However, this disjunction with ε can be shortened with the $\_^{[0\ldots1]}$ instead. Therefore, r_n is of the form $a \cdot s_a + b \cdot s_b$ for some WDREs s_a and s_b. Moreover, it holds that $a^{-1} L_n = L_{n-1}$ such that it already holds that $s_a = r_{n-1}$. On the other hand, we have that $b^{-1} L_n = L_{n-1} \cup \{\varepsilon\}$. Therefore, it remains to show that a minimal WDRE t for the language $L_n \cup \{\varepsilon\}$ is of the form $r_n^{[0\ldots1]}$, where $L(r_n) = L_n$.

Again, we first show that t is not a concatenation. Towards contradiction assume that $t = t_1 \cdot t_2$. Then, we know that $\varepsilon \in L(t_1 t_2)$ and that every word in $L(t_1)$ and $L(t_2)$ has to end with b. Let ub be a longest word in $L(t_1)$ and vb be a longest word in $L(t_2)$. Then, $ubvb \in L(t)$ and $|ubvb| = n + 1$. Similarly as before, we get that the word $uavb$ belongs to $L(t)$ and, therefore, that the word ua belongs to $L(t_1)$. Since $\varepsilon \in L(t_2)$, the word ua is also $L(t)$. However, this situation contradicts the assumption that $L(t) = L_n \cup \{\varepsilon\}$. Thus, every minimal WDRE for the language $L_n \cup \{\varepsilon\}$ has a counter or a disjunction as topmost operation.

We now assume that t is a disjunction. W.l.o.g. we assume that $t = t_1 + t_2$ where $\varepsilon \in L(t_1)$ and where $|\mathsf{first}(t_1)| = 1 = |\mathsf{first}(t_2)|$ because of minimality. Remember, we now have $L(t) = L_n \cup \{\varepsilon\}$ and that the partitioning $t = t_1 + \varepsilon$ would be larger than $t_0^{[0\ldots1]}$. As $\varepsilon \in L(t_1)$ and $|\mathsf{first}(t_1)| = 1$, t_1 cannot be a concatenation, because one of the subexpression would have to be ε, which is a contradiction to minimality. We can also exclude that t_1 is a counter by a similar reasoning as above. The remaining case is that t_1 is a disjunction, which gives a contradiction, as $|\mathsf{first}(t_1)| = 1$ implies that one of the disjuncts is ε, contradicting minimality.

The only remaining case is that $t = t_0^C$. We can assume that $C^- = 0$, as $\varepsilon \in L(t)$. If we assume $C^+ > 1$, we get the same contradiction as above, i.e., every long word has at least two b. Therefore $C^+ = 1$, which gives the desired result, as $\varepsilon \in L(t_0)$ contradicts minimality and $\varepsilon \notin L(t_0)$ implies $L(t_0) = L_n$. $\square$