A Rewriting System for the Construction of Small Finite Automata from Regular Expressions

Stefan Gulan[a,1,]

[a]*Dept. IV, Computer Science*
Trier University
D-54286 Trier

Abstract

We present a linear-time construction of normalized nondeterministic finite automata with ε-transitions from regular expressions. This is realized by rewriting digraphs that are labeled with regular expressions. The size of the constructed automata is shown to be within $\frac{22}{15}$ times the size of the input expressions. A family of expression that reaches this value is inferred constructively, and the automata provided by our method for such expressions are shown to be minimal. This provides a tight nonasymptotic upper and lower bound for the size-ratio of corresponding regular expressions and finite automata.

Keywords: Regular Expressions, Finite Automata, Descriptional Complexity

1. Introduction

A fundamental result in formal language theory is the equivalent descriptive power of regular expressions and finite automata, first shown by Kleene [1]. While regular expressions, being merely terms, come naturally to humans as a means of denoting regular languages, finite automata suggest themselves as a representation of such languages on the machine level. This makes the construction of finite automata from regular expressions a highly important task in basic human-computer interaction; a survey of various algorithms addressing this problem is given by Watson [2]. The standard situation requiring this conversion is pattern-search in text, where different classes of automata are used: according to Friedl [3], most versions of `awk` and `egrep` construct deterministic automata, while `emacs`, `less`, and `sed` employ nondeterministic ones. In this work we consider the latter kind of automata, enriched with ε-transitions.

The efficiency of an expression-to-automaton conversion is generally measured by comparing the size of the input expression to that of the output automaton. Several definitions of the sizes of expressions and automata have been proposed: the number of states and / or the number of transitions for automata and any reasonable combination of the number of products, sums, stars, letters and parentheses for regular expressions. Linear dependencies between most such measures are known [4, 5]; here, we settle for the number of states and transitions as automaton-, and the number of literals and

Email address: `gulan@uni-trier.de` (Stefan Gulan)

operators as expression size. It was shown by Ilie & Yu [6] that the ratio of automaton-to-expression sizes is bounded by 1.33 from below and by 1.5 from above. These bounds were tightened by Gulan & Fernau [7] to the common value of $\frac{22}{15}$, or approximately 1.47. Although this value is correct, the line of argumentation in [7] is flawed; the results will be reinstated in this work using a slightly different construction and a considerably different analysis.

The article is organized as follows: First we introduce a rewriting system that realizes the conversion from regular expression to finite automata. We slightly constrain the rewriting rules in order to guarantee unique outputs. Next, we show that our construction satisfies a property already known for the construction of *position-automata* [8, 9]; namely, that the output is invariant under taking the star normal form of the input-expression [10]. The main effort of this work is to bound the size of a constructed automata relative to the size of the input. This is done by constructing a class of expressions that maximize this value; we further prove that no smaller automata can be constructed for such expressions by any algorithm. Finally, we propose a modified algorithm that improves on a naive implementation in several ways.

2. Preliminaries

Braces for singleton sets may be omitted. The size of a finite set S is denoted $|S|$. Binary relations are written in infix notation, and the reflexive and transitive closure of a binary relation R is denoted $R^\star$.

A *multidigraph*, or just *graph*, is a pair $D = (V, A)$ where V is the set of vertices and A is the multiset of *arcs* over $V \times V$. Both V and A will be referred to as *elements* of D. A *cycle* is a path from a vertex to itself, and a *loop* is a cycle consisting of a single arc. Let q^-, q^+ denote the number of arcs entering, resp. leaving q, called the *in-*, resp. *out-degree* of q.

An *alphabet* $\mathcal{A}$ is a finite set of symbols, called *letters*. Any alphabet is supposed to be free of the symbols ε and $\varnothing$. A *literal* is a letter, ε or $\varnothing$. A *word* over $\mathcal{A}$ is a juxtaposition of letters from $\mathcal{A}$, and a *language* over $\mathcal{A}$ is a set of words over $\mathcal{A}$. If L and L' are languages over $\mathcal{A}$ and $\mathcal{A}'$, their *product* LL' is a language over $\mathcal{A} \cup \mathcal{A}'$, defined as

$$LL' := \{ww' \mid w \in L \wedge w' \in L'\}.$$

The i-th power of L, denoted L^i is defined as $L^0 = \{\varepsilon\}$ and $L^{n+1} = L^n L$. The *Kleene closure* of L is

$$L^* := \bigcup_{n \in \mathbb{N}} L^n$$

A *regular expression* over $\mathcal{A}$, or just *expression*, is a term that defines a language over $\mathcal{A}$. Lowercase greek letters $\alpha, \beta, \mu, \nu, \kappa, \sigma, \pi$, possibly indexed, always denote expressions. The language defined by α is denoted $L(\alpha)$. The set of regular expressions over $\mathcal{A}$ is denoted $\textsc{reg}(\mathcal{A})$. Expressions and their associated languages are defined inductively:

- The symbols ε and $\varnothing$ are expressions over $\mathcal{A}$, with $L(\varepsilon) := \{\varepsilon\}$ and $L(\varnothing) := \{\}$.
- Every $x \in \mathcal{A}$ is an expression over $\mathcal{A}$, with $L(x) := \{x\}$.

- If α_1 and α_2 are expressions over $\mathcal{A}$, so are
  - The *product* $(\alpha_1 \bullet \alpha_2)$, with *factors* α_i. We generally omit the operator $\bullet$, writing just $(\alpha_1\alpha_2)$. $L((\alpha_1\alpha_2)) := L(\alpha_1)L(\alpha_2)$.
  - The *sum* $(\alpha_1 + \alpha_2)$,with *addends* α_i. $L((\alpha_1 + \alpha_2)) := L(\alpha_1) \cup L(\alpha_2)$.
  - The *iteration* (α_1^*), with *base* α_1. $L((\alpha_1^*)) := L(\alpha_1)^*$.

It is customary to write expressions in a more concise way by omitting certain parentheses. Above semantics suggest that products and sums are associative, thus parentheses will be omitted for nested sums and for nested products. We also agree on the operator precedence $^*, \bullet, +$, mimicking the precedence of Kleene closure over concatenation over union of languages. Lastly, outermost parentheses can be omitted. For example, we write

$$(a + b + c)a^* \quad \text{instead of} \quad (((a + b) + c)(a^*)).$$

An expression α is *nullable* if $\varepsilon \in L(\alpha)$. It is called *complex*, if it contains at least one regular operator, and *trivial* otherwise. The set of *subexpressions* of α, denoted $\mathrm{sub}(\alpha)$, is defined as

- $\mathrm{sub}(\alpha) := \{\alpha\}$ if α is a literal
- $\mathrm{sub}(\alpha) := \{\alpha\} \cup \begin{cases} \mathrm{sub}(\alpha_1) \cup \mathrm{sub}(\alpha_2), & \text{if} \quad \alpha = \alpha_1\alpha_2 \quad \text{or} \quad \alpha = \alpha_1 + \alpha_2; \\ \mathrm{sub}(\alpha_1), & \text{if} \quad \alpha = \alpha_1^*. \end{cases}$

A subexpression of α is *proper* if it does not coincide with α. Every proper subexpression β in α is an operand to one of $\bullet$, $+$, or *, which is called its *parent* and denoted $p_\alpha(\beta)$. The *root* of α, denoted $\mathrm{r}(\alpha)$ is the topmost operator of α; if α is trivial, we set $\mathrm{r}(\alpha) = \alpha$. Exploiting associativity of products and sums, we extend notation to arbitrary arity by setting

$$\prod_{i=1}^{n} \pi_i := \pi_1 \bullet \cdots \bullet \pi_n \qquad \text{and} \qquad \sum_{i=1}^{n} \sigma_i := \sigma_1 + \cdots + \sigma_n.$$

A product or sum in the conventional sense is called *binary*. The product $\pi = \prod \pi_i$ is *maximal* if it is no factor of a larger product and no π_i is a product. Maximal sums are defined respectively. If all arguments in a maximal product or sum are iterations, it is called *star-maximal*. Let $|\alpha|_\bullet$ ($|\alpha|_+$, $|\alpha|_*$, $|\alpha|_{\mathcal{A}}$) denote the number of binary products (binary sums, stars, literals) in α.

An *extended finite automaton* (EFA), is a 5-tuple $E = (Q, \mathcal{A}, \delta, I, F)$, with Q being the finite set of *states*, $\mathcal{A}$ an alphabet, $\delta \subseteq Q \times \mathrm{REG}(\mathcal{A}) \times Q$ the finite *transition* relation, $I \subseteq Q$ and $F \subseteq Q$. A member $(p, \alpha, q) \in \delta$ is an α-transition, or just transition, of E, with *label* α. The relation $\vdash_E \subseteq Q \times \mathcal{A}^*$ is defined via $(q, w_1w_2) \vdash_E (q', w_2)$ iff $(q, \alpha, q') \in \delta \wedge w_1 \in L(\alpha)$. The language accepted by E is

$$L(E) := \{w \mid (q_i, w) \vdash_E^\star (q_f, \varepsilon), q_i \in I, q_f \in F\}.$$

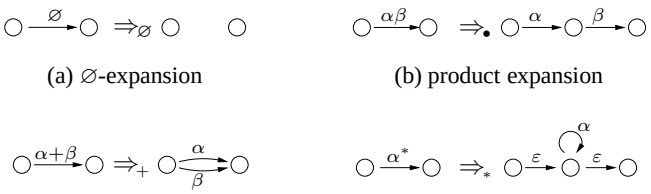
Let $\mathcal{E}$ denote the class of EFAs. A finite automaton with ε-transitions (FA), is an EFA whose transition relation is restricted to $Q \times (\mathcal{A} \cup \varepsilon) \times Q$. We call $E \in \mathcal{E}$ *normalized* if $|I|=|F|=1$. The *size* of E is $|E| := |Q|+|\delta|$. An EFA is essentially an arc-labeled graph with two predicates on its vertices. We will thus reason about EFAs by means of their graph theoretic properties. We use graph theoretic terminology for EFAs, e.g., by referring to a transition (q, α, q) as a loop.

An *abstract rewriting system* (ARS) on a set M is a tuple $\langle M, \Rightarrow_1, \cdots, \Rightarrow_n \rangle$ where each $\Rightarrow_i$ is a binary relation on M. Every $\Rightarrow_i$ is called a *rewriting rule*, or just rule; an element of $\Rightarrow_i$ is called a *rewriting*. A *normal form* of $\Rightarrow_i$ is any m s.t. for every m' we have $m \not\Rightarrow_i m'$. A *rewriting sequence* is any sequence of consecutive rewritings $m_1, m_2, \ldots$, where $m_j \Rightarrow_{i_j} m_{j+1}$ for every j. An ARS is *terminating* if every rewriting sequence is finite. If for some m, $m \Rightarrow_i m_1$ and $m \Rightarrow_j m_2$, then m_1 and m_2 are a *divergence* of m. Two objects m_1 and m_2 *converge*, denoted $m_1 \equiv m_2$, if some m_3 exists with $m_1 \Rightarrow^\star m_3$ and $m_2 \Rightarrow^\star m_3$. An ARS is *locally confluent* if every divergence converges.

3. Converting Regular Expressions to Finite Automata

The presented construction extends those given by Ott & Feinstein [11] and Gulan & Fernau [7]. It is a rewriting system on the class of normalized EFAs. Every rewriting modifies an EFA locally without altering the accepted language. Rewriting rules come in two flavors:

- *Expansions* replace single transitions according to (the root of) their label. Rules are denoted $\Rightarrow_\varnothing$, $\Rightarrow_\bullet$, $\Rightarrow_+$, $\Rightarrow_*$, and sketched in Fig. 1.
- *Eliminations* replace substructures containing ε-transitions with smaller equivalents. Rules are denoted $\Rightarrow_X$, $\Rightarrow_Y$, $\Rightarrow_O$, $\Rightarrow_Z$, and sketched in Fig. 2.

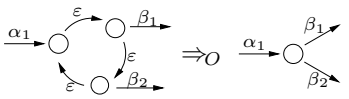


(a) $\varnothing$-expansion (b) product expansion

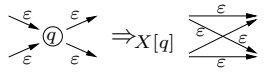
(c) sum expansion (d) star expansion

Figure 1: An expansion replaces a complex transition (p, ϱ, q) depending on $\mathrm{r}(\varrho)$

The expansions are called $\varnothing$-expansion, sum expansion and product expansion, while the eliminations are called O-elimination, X-elimination, Y-elimination and Z-elimination. Note that Y- and Z-elimination consist of two cases each, both of which are sketched in Figs. 2c and 2d. These cases behave symmetrically, which is why we will always consider only one of them. An occurrence of the left-hand side of rule $\Rightarrow_i$

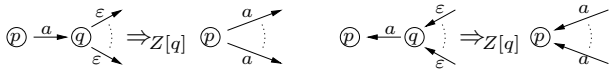


(a) ε-cycle-, or O-elimination



(b) X-elimination: $q^- = q^+ = 2$

(c) Y-elimination(s): $p \neq q$, $q^- = 1$ resp. $q^+ = 1$



(d) Z-elimination(s): $p \neq q$, $a \in \mathcal{A}$, $q^- = 1$ resp. $q^+ = 1$

Figure 2: An elimination replaces a substructure that contains ε-transitions with a smaller equivalent. For X-, Y- and Z-elimination, additional properties must be met.

is referred to as an i-*anchor* [1]. As can be seen from Fig. 2, X-, Y- and Z-anchors are uniquely identified by a central state; this information is occasionally conveyed by writing e.g. $\Rightarrow_{Y[q]}$ instead of $\Rightarrow_Y$, if q is the central state.

This defines the ARS $\langle \mathcal{E}, \Rightarrow_\varnothing, \Rightarrow_\bullet, \Rightarrow_+, \Rightarrow_*, \Rightarrow_O, \Rightarrow_X, \Rightarrow_Y, \Rightarrow_Z \rangle$. The rewritings will also be called *conversions*; if the particular type of a conversion from E to E' is irrelevant, we write just $E \Rightarrow E'$. Formally, we set

$$\Rightarrow := \bigcup_{\mathrm{i} \in I} \Rightarrow_\mathrm{i} \text{ where } I = \{\varnothing, \bullet, +, *, O, X, Y, Z\}.$$

It should be clear from Figs. 1 and 2 that the language accepted by an EFA is invariant under conversions, i.e., we have

Proposition 3.1. *If $E \Rightarrow E'$ then $L(E) = L(E')$.*

Every conversion reduces either the overall number of operators, or the number of states or transitions in an EFA. Therefore, the system is terminating and every normal form is an FA. On input $\alpha \in \mathrm{REG}(\mathcal{A})$ the conversion is initialized with the trivial EFA A^0_α, defined as

$$A^0_\alpha := (\{q_0, q_f\}, \mathcal{A}, (q_0, \alpha, q_f), q_0, q_f).$$

An full conversion starting from some A^0_α is given in Fig. 3. Let A^k_α denote any EFA that can be derived from A^0_α through a sequence of k rewritings, i.e. $A^0_\alpha \Rightarrow^k A^k_\alpha$; this notation allows us to write $A^k_\alpha \Rightarrow A^{k+1}_\alpha$. If A^k_α is in normal form, we denote it just A_α; clearly, the FA A_α is free of O-, X-, Y- or Z-anchors.

A shortcoming of our construction is that its output A_α is generally not unique. This is due to the fact that the ARS $\langle \mathcal{E}, \Rightarrow \rangle$ is not confluent, i.e., some divergences are not guaranteed to converge again. We rectify this behavior in the following subsection.

[1] A more common term is *reducible expression*, or *graph-redex*; since we are already dealing with expressions, a different name seems to be better suited

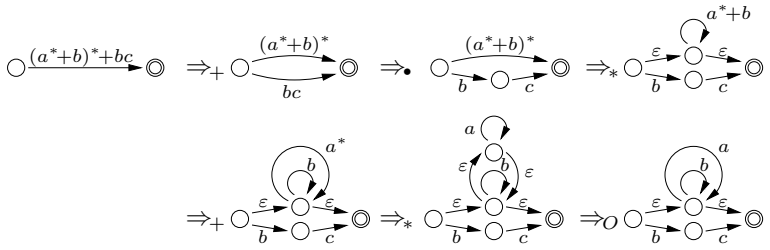
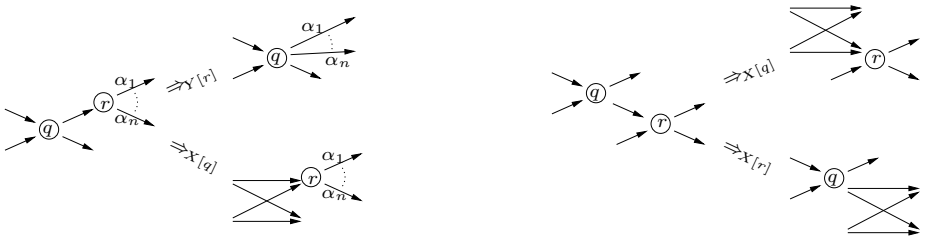
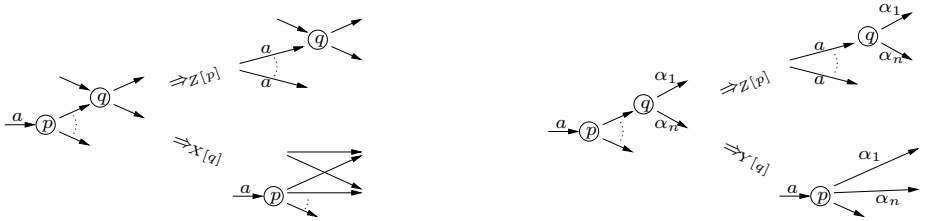


Figure 3: Converting A^0_α into A_α for $\alpha = (a^* + b)^* + bc$



(a) X/Y - divergence

(b) X/X - divergence



(c) Z/X - divergence

(d) Z/Y - divergence

Figure 4: Divergent rewritings due to X- and Z-anchors (ε-labels are omitted).

3.1. Refining the Construction towards Functionality

As mentioned, different conversion sequences may lead to non-isomorphic automata. It turns out that this is due to X- and Z-eliminations — more specifically, certain instances wherein an X- or Z-anchor shares states with a second anchor (possibly of a different type). Typical examples for this kind of divergence are shown in Fig. 4.

We thus restrict applicability of X- and Z-elimination as follows:

Definition 1. An X-elimination $E \Rightarrow_{X[q]} E'$ is *valid*, denoted $E \Rrightarrow_{X[q]} E'$, if

(X1) only X- or Z-elimination are possible in E, and
(X2) there is no $X[p]$-anchor in E with $(p, \varepsilon, q) \in \delta_E$.

A Z-elimination is $E \Rightarrow_{Z[q]} E'$ is valid, denoted $E \Rrightarrow_{Z[q]} E'$, if

(Z1) only Z-elimination is possible in E, and

(Z2) there is no $Z[p]$-anchor in E with any transition from p to q in δ_E.

For example, in Fig. 4a, X-elimination is not valid, because Y-elimination is applicable, whereas in Fig. 4b, $X[r]$-conversion is not valid, because of an $X[q]$-anchor with ε-transition from q to r. Likewise, in Fig. 4c, Z-elimination is not valid, since an X-elimination is applicable. The rules **(X2)** and **(Z2)** are sound: a cyclic elimination-preference among X- or Z-anchors implies the existence of an ε-cycle, which, due to **(X1)** and **(Z1)**, is eliminated first.

Lemma 3.2. *If $E \Rrightarrow_X E'$ holds, then E' contains only X- and/or Z-anchors, if any.*

Proof. Since $E \Rrightarrow_X E'$ respects **(X1)**, E contains only X- and Z-anchors, so every different anchor in E' results from this elimination. X-elimination certainly does not introduce expansion anchors, or cycles, particularly no O-anchors. Let the considered elimination be $E \Rightarrow_{X[p]} E'$ and assume $E' \Rightarrow_{Y[q]} E''$. This implies $(p, \varepsilon, q) \in \delta_E$, so $q^- \geq 2$ in E', and, since q is the center of a Y-anchor, $q^+ = 1$ in E'. However, q^+ is not affected by the X-elimination, so $q^+ = 1$ in E, too. But then, E also contains a $Y[q]$-anchor, contradicting the assumption that $E \Rightarrow_X E'$ is valid. □

Lemma 3.3. *If $E \Rrightarrow_Z E'$ holds, then E' contains only Z-anchors, if any.*

Proof. As for Lem. 3.2 with the obvious modifications. □

Any conversion besides X- or Z-elimination is always valid. Let $\Rightarrow_{\neg\{X,Z\}}$ denote these other conversions, i.e., set

$$\Rightarrow_{\neg\{X,Z\}} \;:=\; \Rightarrow \setminus \{\Rightarrow_X, \Rightarrow_Z\}.$$

It follows from Lems. 3.2 and 3.3 that a sequence of valid conversions is split into three parts: a sequence without X- or Z-elimination, followed by a sequence with only X-eliminations, and a final one with only Z-eliminations.

Corollary 3.4. *If $E \Rightarrow^\star F$ is valid, then there are EFAs E' and E''*

$$E \Rightarrow^\star_{\neg\{X,Z\}} E' \Rrightarrow^\star_X E'' \Rrightarrow^\star_Z F$$

Setting $E = A^0_\alpha$ and $F = A_\alpha$ in Cor. 3.4, we arrive at two intermediate EFAs that appear in the construction of an FA from an expression by means of exhaustive valid conversions.

Definition 2. Let X_α and Z_α denote any two EFAs satisfying

$$A^0_\alpha \Rightarrow^\star_{\neg\{X,Z\}} X_\alpha \Rrightarrow^\star_X Z_\alpha \Rrightarrow^\star_Z A_\alpha.$$

Note that each subsequence in Cor. 3.4 can be of length zero, so some of A^0_α, X_α, Z_α, A_α may coincide. In the remainder of this section we show that X_α, Z_α and A_α are unique. To this end, define the *overlap* of two conversions as the elements shared by their anchors. Non-overlapping conversions clearly yield EFAs that converge, for in that case, conversions take place in "different parts" of the EFA, and their relative order is irrelevant. We consider nontrivial overlaps only, i.e., elements which are shared by distinct anchors.

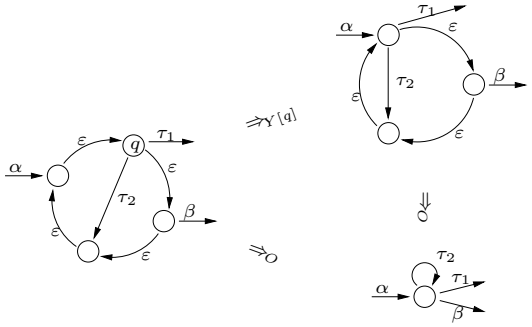


Figure 5: Case in the proof of Lem. 3.5. The ε-transition of the $Y[q]$-anchor is also part of an ε-cycle.

Lemma 3.5. *If $E \Rightarrow_O E_1$ holds and $E \Rightarrow E_2$ is valid, then $E_1 \equiv E_2$.*

Proof. Let Q_ε and T_ε denote the sets of states and transitions of the ε-cycle eliminated in $E \Rightarrow_O E_1$. O-elimination boils down to the following: Choose any $q \in Q_\varepsilon$, called the *representative*, and replace $(p, \alpha, r) \in \delta_E \setminus T_\varepsilon$ with (p, α, q), if $r \in Q_\varepsilon$, resp. with (q, α, r) if $p \in Q_\varepsilon$ and with (q, α, q) if $p, r \in Q_\varepsilon$. Next, remove $Q_\varepsilon \setminus q$ from Q_E and T_ε from δ_E, thus yielding E_1. Assume $E \Rightarrow_\mathrm{i} E_2$ overlaps with this cycle-elimination: since $\Rightarrow_\mathrm{i}$ is valid, $\mathrm{i} \notin \{X, Z\}$, and we distinguish by the remaining cases.

- If $\mathrm{i} \in \{\bullet, +, *, \varnothing\}$, let $t = (p, \rho, q)$ be the transition replaced by $\Rightarrow_\mathrm{i}$. The overlap of $\Rightarrow_O$ and $\Rightarrow_\mathrm{i}$ consists of p or q (or both); we assume that q is part of the overlap and choose it as the representative. If $p \notin Q_\varepsilon$, then t is unaffected by O-elimination. Conversely, the ε-cycle remains unmodified if $\Rightarrow_\mathrm{i}$ is applied. If $p \in Q_\varepsilon$, too, t is replaced with $t' = (q, \rho, q)$ by O-elimination; nevertheless, $\Rightarrow_\mathrm{i}$ is still applicable to t' in E_1. As an example, this is shown for an overlap of $\Rightarrow_O$ and $\Rightarrow_*$ in Fig. 6.

- If $\mathrm{i} = Y$, assume $E \Rightarrow_{Y[q]} E_2$ with $q^- = 1$ and $q^+ \geq 1$; let $t = (p, \varepsilon, q)$ be ε-transition of this anchor. If $q \notin Q_\varepsilon$, a $Y[q]$-anchor is also present in E_1, while conversely, the ε-cycle of E is not affected by Y-elimination at all. This case is straightforward. If $q \in Q_\varepsilon$ we need to distinguish whether t is a part of the ε-cycle. If so, $\Rightarrow_{Y[q]}$ shrinks the ε-cycle into an ε-cycle which can be eliminated in E_2. On the other hand Y-elimination is subsumed by applying O-elimination to E. See Fig. 5 for this case. The final case where $q \in Q_\varepsilon$ and t is no part of the ε-cycle is left to the reader.

- If $\mathrm{i} = O$, the two ε-cycles share a state q which we choose as the representative of either cycle. Regardless of the order of the two eliminations, each transition incident to either cycle is replaced by a transition incident to q in an EFA that is reached from both E_1 and E_2.

□

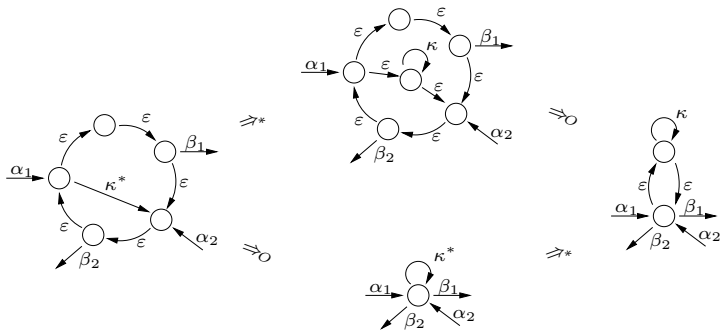


Figure 6: Case in the proof of Lem. 3.5. Convergence of a divergence resulting from $\Rightarrow_*$ and $\Rightarrow_O$.

Lemma 3.6. *If $E \Rightarrow_Y E_1$ holds and $E \Rightarrow E_2$ is valid, then $E_1 \equiv E_2$.*

Proof. Assume $E \Rightarrow_{Y[q]} E_1$ where $q^- = 1$ and $(p, \varepsilon, q) \in \delta_E$. Since $E \Rightarrow_\mathrm{i} E_2$ is valid, again $\mathrm{i} \notin \{X, Z\}$. The case $\mathrm{i} = O$ was already dealt with in Lem. 3.5, we consider the remaining possibilities.

- If $\mathrm{i} \in \{\bullet, +, *, \varnothing\}$, let $t = (q, \rho, r)$ be the transition replaced by $\Rightarrow_\mathrm{i}$. In E_1, t is replaced by $t' = (p, \rho, r)$, which is an i-anchor, too. Let $E_2 \Rightarrow_\mathrm{i} E_3$ be the appropriate conversion, then the elements caused in E_3 by this conversion are the same that are caused in E_1 by replacing t, except that the transitions leaving q in E_1 leave p in E_3. It is easy to see that $E_1 \Rightarrow_{Y[q]} E_3$ is valid, so $E_1 \equiv E_2$.
- If $\mathrm{i} = Y$, there is a $Y[r]$-anchor in E s.t. at at least one transition connects q and r. This gives rise to three sub-cases which are shown in Fig. 7; in one case (Fig. 7(c)) we must consider $p = r$. The truth of the claim should be obvious from the figure.

□

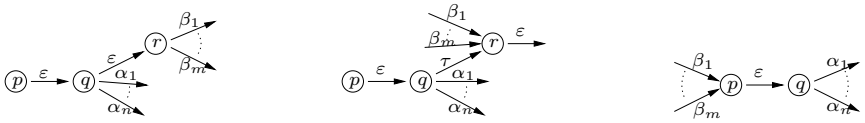


Figure 7: Possible overlaps of Y-anchors

Lemma 3.7. *If $E \Rightarrow_\mathrm{i} E_1$ for some $\mathrm{i} \in \{\bullet, +, *, \varnothing\}$ holds and $E \Rightarrow E_2$ is valid, then $E_1 \equiv E_2$.*

Proof. Let E_2 result from $E \Rightarrow_\mathrm{j} E_2$. Once more, $\mathrm{j} \notin \{X, Z\}$, because $E \Rightarrow_\mathrm{j} E_1$ is valid. Due to Lems. 3.5 and 3.6, only $\mathrm{j} \in \{\bullet, +, *, \varnothing\}$ needs to be considered; in

any case, the overlap consists of either one or two states but no transition. No state is removed upon expansion and the in- and outdegrees of states are irrelevant to $\Rightarrow_{\mathrm{i}}$ and $\Rightarrow_{\mathrm{j}}$, the order of conversions can be switched without altering the resulting EFA. □

Lems. 3.5, 3.6 and 3.7 constitute the cases in the proof of the following

Lemma 3.8. *The ARS* $\langle \mathcal{E}, \Rightarrow_{\varnothing}, \Rightarrow_{+}, \Rightarrow_{\bullet}, \Rightarrow_{*}, \Rightarrow_{Y}, \Rightarrow_{O} \rangle$ *is locally confluent.*

This implies that X_α is unique for any input α (cf. Thm. 3.11). We examine the remaining parts of an exhaustive conversion, i.e., valid X- and Z-eliminations. To this end let $\mathcal{X}$ denote the class of FAs without Y- or O-anchors.

Lemma 3.9. *The ARS* $\langle \mathcal{X}, \Rrightarrow_X \rangle$ *is locally confluent.*

Proof. Anchors of distinct valid X-eliminations can only overlap in a state which is not the central state of either conversion. It is easily seen that the FA resulting from two such elimination does not depend on the relative order of these eliminations. □

Likewise, let $\mathcal{Z}$ denote the class of FAs without Y-, O- or X-anchors.

Lemma 3.10. *The ARS* $\langle \mathcal{Z}, \Rrightarrow_Z \rangle$ *is locally confluent.*

Proof. Suppose $E \Rightarrow_{Z[p]} E_1$ and $E \Rightarrow_{Z[q]} E_2$ are valid and overlapping. If the overlap consists of a state and no transition, the conversions do not interfere since the common state is neither p nor q . If the overlap consists of a transition t, then there is necessarily a transition from p to q, contradicting the assumption that both eliminations are valid, i.e., both respect **(Z2)**. □

The confluence properties of each of the rewriting system we use in an exhaustive conversion imply the main result of this section. In particular it states that, if restricted to valid conversions, the presented construction is functional in the sense that the resulting FA is uniquely determined.

Theorem 3.11. *The automata* X_α, Z_α *and* A_α *are unique.*

Proof. This is an application of Newman's Lemma [12, 13], which implies that the normal forms of a locally confluent ARS without infinite sequences are unique. □

Confluence of valid conversions implies a property that comes in handy later on.

Lemma 3.12. *The relation* $\equiv$ *is an equivalence on* $\mathcal{E}$.

Proof. Reflexivity and symmetry are trivial. To prove transitivity, suppose $E_1 \equiv E_2$ and $E_2 \equiv E_3$. Then there are E_{12} and E_{13} with $E_1 \Rightarrow^\star E_{12}$, $E_2 \Rightarrow^\star E_{12}$, $E_2 \Rightarrow^\star E_{23}$ and $E_3 \Rightarrow^\star E_{23}$. Since E_{12} and E_{23} are both derived from E_2, confluence of valid conversions imply $E_{12} \equiv E_{23}$, so $E_{12} \Rightarrow^\star E_{13}$ and $E_{23} \Rightarrow^\star E_{13}$ for some EFA E_{13}. Therefore, $E_1 \Rightarrow^\star E_{13}$ and $E_3 \Rightarrow^\star E_{13}$, i.e. $E_1 \equiv E_3$. □

4. Star Normal Form

The *star normal form* of an expression was introduced by Brüggemann-Klein [10] as a preprocessing step in the construction of the position-FA from an expression [8]. We use a slightly more succinct definition than in the original work.

Definition 3. The operators $^\circ$ and $^\bullet$ are defined on expressions as follows

- $[\cdot]^\circ$: $\varnothing^\circ = \varepsilon^\circ := \varnothing,\ a^\circ := a,\ (\alpha{+}\beta)^\circ := \alpha^\circ{+}\beta^\circ,\ (\alpha^*)^\circ := \alpha^\circ,$

$$(\alpha\beta)^\circ := \begin{cases} \alpha\beta, & \text{if } \varepsilon \notin L(\alpha\beta); \\ \alpha^\circ{+}\beta^\circ, & \text{else.} \end{cases}$$

- $[\cdot]^\bullet$: $\varnothing^\bullet := \varnothing,\ \varepsilon^\bullet := \varepsilon,\ a^\bullet := a$
  $(\alpha+\beta)^\bullet := \alpha^\bullet + \beta^\bullet,\ (\alpha\beta)^\bullet := \alpha^\bullet\beta^\bullet,\ \alpha^{*\bullet} := \alpha^{\bullet\circ *}$

The star normal form (SNF) of α is defined as $\alpha^\bullet$, and an expression α is in SNF if $\alpha = \alpha^\bullet$. Some elementary properties of SNF, proven in [10], which will be used without further explicit mention, are the following:

1. $L(\alpha) = L(\alpha^\bullet)$
2. $\alpha^\bullet = \alpha^{\bullet\bullet}$
3. $\alpha = \alpha^\bullet \quad \text{iff} \quad \forall \gamma \in \mathrm{sub}(\alpha) : \gamma = \gamma^\bullet$

Brüggemann-Klein further proved that α and $\alpha^\bullet$ yield identical position-FAs. We claim that the same is true for the construction of FAs by exhaustive valid conversions.

Some additional notation is helpful in proving this: given a transition $t = (p, \alpha, q)$, let $t^\circ := (p, \alpha^\circ, q)$ and $t^\bullet := (p, \alpha^\bullet, q)$. If E is an EFA with transition t, let $E[t^\circ]$ and $E[t^\bullet]$ denote the EFA that results from replacing t with t°, resp. $t^\bullet$. The following lemma lies at the heart of proving our claim.

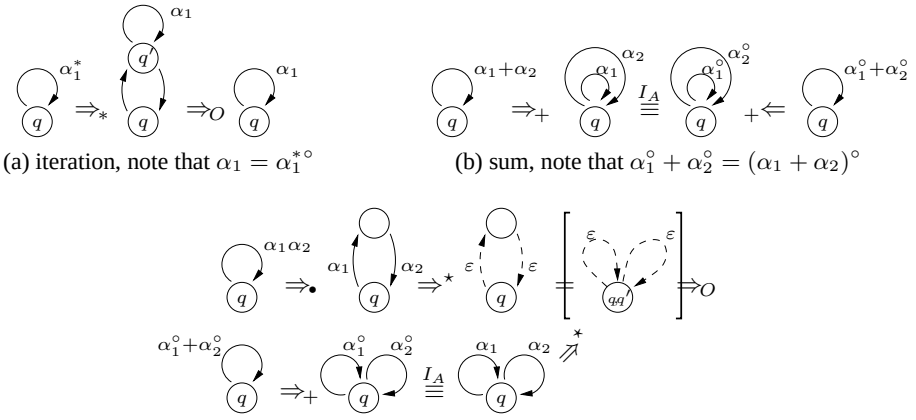
Lemma 4.1. *Let t be a loop of E. Then $E \equiv E[t^\circ]$.*

Proof. Let $t = (q, \alpha, q)$ and proceed by induction on α. For $\alpha \in \mathcal{A} \cup \{\varnothing\}$, we find $\alpha = \alpha^\circ$ and thus $E = E[t^\circ]$. If $\alpha = \varepsilon$, the transition t is a one-arc ε-cycle, which is removed through O-elimination. With $\varepsilon^\circ = \varnothing$, the transition t° is removed by $\varnothing$-expansion; therefore $E \Rightarrow_O E'$ and $E[t^\circ] \Rightarrow_\varnothing E'$.

Now suppose the statement is true for α_i and distinguish by the structure of α (the cases are sketched in Fig. 8):

- α_1^*: star expansion replaces t with a state q', an ε-cycle $\{(q, \varepsilon, q'), (q', \varepsilon, q)\}$, and a loop (q', α_1, q'). Eliminating the cycle identifies q' with q, so the new loop becomes (q, α_1, q), which is just t°. Hence $E \Rightarrow^\star E[t^\circ]$, which implies the claim.

- $\alpha_1 + \alpha_2$: sum expansion in E replaces t with loops $t_1 = (q, \alpha_1, q)$ and $t_2 = (q, \alpha_2, q)$: if E' denotes the resulting EFA, the inductive assumption implies $E' \equiv E'[t_1^\circ][t_2^\circ]$. Sum expansion of $t^\circ = (q, (\alpha_1 + \alpha_2)^\circ, q) = (q, \alpha_1^\circ + \alpha_2^\circ, q)$ in $E[t^\circ]$ yields $E'[t_1^\circ][t_2^\circ]$, too, and since $\equiv$ is an equivalence, $E \equiv E[t^\circ]$ follows.



(a) iteration, note that $\alpha_1 = \alpha_1^{*\circ}$

(b) sum, note that $\alpha_1^\circ + \alpha_2^\circ = (\alpha_1 + \alpha_2)^\circ$

(c) nullable product, note that $(\alpha_1\alpha_2)^\circ = \alpha_1^\circ + \alpha_2^\circ$

Figure 8: Cases in the proof of Lem. 4.1: replacing a loop t with t° yields an EFA that converges with the initial one. Dashed arrows labeled ε represent ε-paths. $\overset{I_A}{\equiv}$ denotes convergence by inductive assumption.

- $\alpha_1\alpha_2$: If $\varepsilon \notin L(\alpha_1\alpha_2)$, then $\alpha = \alpha^\circ$, so $E \equiv E[t^\circ]$ holds. Otherwise let E' be derived from E by exhaustive expansion of t. The first step in this sequence is product expansion of t, which is replaced by a new state q' and transitions $\{(q, \alpha_1, q'), (q', \alpha_2, q)\}$. Since $\alpha_1\alpha_2$ is nullable, so is either α_i, hence E' contains an ε-cycle through q and q'. Consider the elimination of an ε-cycle in "slow motion": it consists of a sequence of successively merging pairs of states from the cycle, preserving the accepted language in each step, and a final removal of ε-loops that stem from the ε-transitions of the original cycle. In E' we may thus merge q and q' to get the intermediate EFA $E'_{[q=q']}$: this EFA can also be constructed from $E[t^\circ]$, where

$$t^\circ = (q, (\alpha_1\alpha_2)^\circ, q) = (q, \alpha_1^\circ + \alpha_2^\circ, q)$$

since $\alpha_1\alpha_2$ is nullable. Thus E and E' admit convergent rewriting sequences which, based on Lem. 3.8, imply $E \equiv E[t^\circ]$ for this case. □

Lemma 4.2. *Let t by any transition of E. Then $E \equiv E[t^\bullet]$.*

Proof. Given $t = (p, \alpha, q)$ we prove the claim by induction on α: the statement is true for $\alpha \in \mathcal{A} \cup \{\varepsilon, \varnothing\}$, so assume it is true for labels α_1, α_2. Now distinguish by the structure of α:

- $\alpha_1\alpha_2$ or $\alpha_1+\alpha_2$: both cases are straightforward.

- α_1^*: let $E \Rightarrow_* E'$ be the expansion of t, then E' is as E except that t is replaced by a new state r, ε-transitions $(p, \varepsilon, r), (r, \varepsilon, q)$ and a loop $t' = (r, \alpha_1, r)$. Applying the inductive assumption to t' we find $E' \equiv E'[t'^\bullet]$. Since t', resp. $t'^\bullet$ is a

loop, Lem. 4.1 yields $E'[t'^\bullet] \equiv E'[t'^{\bullet\circ}]$. Now $\alpha^\bullet = \alpha_1^{*\bullet} = \alpha_1^{\bullet\circ *}$, so expanding $t^\bullet$ in $E[t^\bullet]$ yields $E'[t'^{\bullet\circ}]$, too. Therefore $E[t^\bullet] \equiv E'[t'^{\bullet\circ}]$ and since $\equiv$ is an equivalence, we also have $E \equiv E[t^\bullet]$.

□

Theorem 4.3. *The presented construction satisfies $A_\alpha = A_{\alpha^\bullet}$ for arbitrary α.*

Proof. Applying Lemma 4.2 to the initial EFA of a conversion yields $A^0_\alpha \equiv A^0_{\alpha^\bullet}$, so confluence of $\langle \mathcal{E}, \Rightarrow_{\neg\{X,Z\}} \rangle$ leads to $X_\alpha = X_{\alpha^\bullet}$. From there on, Lems. 3.9 and 3.10 provide the claim. □

Therefore, the presented construction is invariant under taking the SNF of the input; from a different point of view, one might say that the SNF is implicitly computed upon conversion. This property is fundamental for analyzing the size of a A_α relative to the size of α. To this end, we also need the following alternative characterization of expressions in SNF.

Lemma 4.4.

1. $\alpha = \alpha^\circ$ iff $\varepsilon \notin L(\alpha)$
2. $\alpha = \alpha^\bullet$ iff $\forall \kappa^* \in \mathrm{sub}(\alpha) : \varepsilon \notin L(\kappa)$

Proof.

1. By structural induction on α: If $\alpha = \varepsilon$ then α is nullable and $\alpha \neq \alpha^\circ$, whereas if $\alpha \in \mathcal{A} \cup \varnothing$, it is not nullable and $\alpha = \alpha^\circ$. If $\alpha = \sigma_1 + \sigma_2$ is nullable, assume without loss of generality that σ_1 is nullable: the inductive assumption $\sigma_1 \neq \sigma_1^\circ$ then implies $\alpha \neq \alpha^\circ$. If otherwise $\alpha = \sigma_1 + \sigma_2$ is not nullable, neither σ_i is. Then $\sigma_i = \sigma_i^\circ$ holds, which implies $\alpha = \alpha^\circ$. An iteration $\alpha = \kappa^*$ is always nullable and $\alpha \neq \alpha^\circ$ follows directly from the definition of $[\cdot]^\circ$. Finally let $\alpha = \pi_1\pi_2$: if α is nullable, the definition of $[\cdot]^\circ$ again yields $\alpha \neq \alpha^\circ$ immediately, whereas if α is not nullable then $\alpha = \alpha^\circ$ (regardless of whether a π_i actually *is* nullable).
2. The "only if"-part is again straightforward induction on α, we only prove the "if"-part: For trivial α the claim restates the definition of $^\bullet$. The inductive step is easy if α is a product or sum, so let $\alpha = \kappa^*$ and $\varepsilon \notin L(\kappa)$. Then, $\alpha^\bullet = \kappa^{*\bullet} = \kappa^{\bullet\circ *}$; since all bases of α are non-nullable by assumption and $\kappa \in \mathrm{sub}(\alpha)$, we apply the inductive assumption to κ to find $\kappa^{\bullet\circ *} = \kappa^{\circ *}$. Now the first item of this lemma yields $\kappa^{\circ *} = \kappa^* = \alpha$. In all cases, $\alpha = \alpha^\bullet$, therefore α is in SNF.

□

Informally speaking, SNF formalizes the idea that no base needs to be nullable, as an iteration already allows for ε. Consequently, we ask for the maximal number of stars that can be present in an SNF-expression.

Lemma 4.5. *Let α be in* SNF*, then*

$$|\alpha|_* \leq \begin{cases} |\alpha|_\mathcal{A} - 1, & \text{if } \varepsilon \notin L(\alpha); \\ |\alpha|_\mathcal{A}, & \text{if } \varepsilon \in L(\alpha). \end{cases}$$

Proof. Any trivial expression is in SNF and satisfies the claim. Assume the claim holds for expressions ν_1, ν_2 ,and κ, and distinguish cases according to the structure of α.

- $\alpha = \nu_1\nu_2$: If $\varepsilon \in L(\alpha)$, it follows that $\varepsilon \in L(\nu_i)$, so we have

$$|\alpha|_* = |\nu_1|_* + |\nu_2|_* \leq |\nu_1|_{\mathcal{A}} + |\nu_2|_{\mathcal{A}} = |\alpha|_{\mathcal{A}}.$$

  Otherwise, assume without loss of generality $\varepsilon \notin L(\nu_1)$. Then the right-hand side above is decreased by at least one due to the inductive hypothesis.

- $\alpha = \nu_1 + \nu_2$: This case is dual to the previous one: if $\varepsilon \notin L(\alpha)$, it follows that $\varepsilon \notin L(\nu_i)$. Hence,

$$|\alpha|_* = |\nu_1|_* + |\nu_2|_* \leq |\nu_1|_{\mathcal{A}} - 1 + |\nu_2|_{\mathcal{A}} - 1 < |\alpha|_{\mathcal{A}} - 1.$$

  If $\varepsilon \in L(\alpha)$, the right-hand side above is increased by at most two, yet is still bounded from above by $|\alpha|_{\mathcal{A}}$.

- $\alpha = \kappa^*$: Obviously $\varepsilon \in L(\alpha)$; moreover, Lem. 4.4 states $\varepsilon \notin L(\kappa)$, so $|\kappa|_* \leq |\kappa|_{\mathcal{A}} - 1$ by assumption, and therefore

$$|\alpha|_* = |\kappa|_* + 1 \leq |\kappa|_{\mathcal{A}} = |\alpha|_{\mathcal{A}}.$$

□

5. Conversion-Ratio and Worst-Case Expressions

We seek to bound the size of an FA relative to the size of the expression it is derived from. To this end, the *conversion ratio* of an expression α is defined as

$$\mathrm{c}(\alpha) := \frac{|A_\alpha|}{|\alpha|}.$$

We call β *worse than* α if $\mathrm{c}(\beta) > \mathrm{c}(\alpha)$. An expression μ is *worst-case* if no expression is worse than μ. The conversion ratio of a worst-case expression provides an upper bound on the size of a constructed FAs relative to the size of the input, i.e., if α is an arbitrary expression and μ is worst case, then

$$|A_\alpha| \leq \mathrm{c}(\mu)\,|\alpha|.$$

As we will see, the structure of a worst-case expression is unique up to repetition. It will be deduced through the stepwise exclusion of subexpressions that may occur in such an expression, thus narrowing down the possible structural properties it may exhibit. This is frequently done by showing that a subexpression ϕ, which reflects some structural property, can be replaced by a different subexpression ψ yielding a worse expression. This usually involves comparing intermediate EFAs which, just like the expressions they are derived from, differ only locally. At times we will go to great lengths in order to ensure that the effects of replacing a subexpression are not accidentally diametrical to our original intentions. In particular, we always need to rule out the possibility that

elimination-anchors emerge as the result of a replacement, since these might level the size-increase we wished to achieve.

We write $\mu' = \mu[\psi/\phi]$ to express that μ' is derived from μ by replacing a fixed occurrence of ϕ in μ with ψ. This notation is rather informal, since μ might contain multiple instances of ϕ; however, we will always explicitly introduce the occurrence which will be replaced. If we can show that μ' is worse than μ, we have inferred that a worst-case expression does not contain ϕ as a subexpression, resp. that it does not have the structural property expressed by ϕ.

A most helpful structural property of worst-case expressions results from the invariance of our construction under star normal form:

Theorem 5.1. *A worst-case expression is in* SNF.

Proof. It is easy to see from Def. 3 that $|\alpha^\bullet| \leq |\alpha|$ holds, with equality iff α is in SNF. From Thm. 4.3 we deduce $|A_{\alpha^\bullet}| = |A_\alpha|$, so $\alpha^\bullet$ is worse than α iff $\alpha \neq \alpha^\bullet$. □

The size of a constructed FA is determined by the number of times each conversion is applied. The number of times that $\Rightarrow_\varnothing$, $\Rightarrow_\bullet$, $\Rightarrow_+$ and $\Rightarrow_*$ are applied upon constructing A_α obviously equals $|\alpha|_\varnothing$, $|\alpha|_\bullet$, $|\alpha|_+$ and $|\alpha|_*$, respectively. This notation is generalized by setting $|\alpha|_O$, $|\alpha|_X$, $|\alpha|_Y$ and $|\alpha|_Z$ as the numbers of O-, X-, Y- and Z - eliminations, that occur in a conversion. Note that while $|\alpha|_\mathrm{i}$ depends only on α for $\mathrm{i} \in \{\varnothing, \bullet, +, *\}$, this is generally not true for $\mathrm{i} \in \{O, Y\}$, where the values depend on the chosen conversion sequence. We will show that for worst-case expressions no elimination occurs at all, no matter what the chosen conversion is.

Lemma 5.2. *The following statements are equivalent:*

1. α *is in* SNF
2. $\kappa^* \in \mathrm{sub}(\alpha)$ *implies* $\varepsilon \notin L(\kappa)$
3. $|\alpha|_O = 0$

Proof. The equivalence of 1. and 2. has been established in Lem. 4.4. We show the equivalence of 2. and 3. Assume $\kappa^* \in \mathrm{sub}(\alpha)$ and $\varepsilon \in L(\kappa)$. At some point in the construction of A_α a loop (q, κ, q) emerges; expanding this loop necessarily yields an ε-path from q to q, i.e. an ε-cycle. Conversely, any cycle in the construction presupposes a loop s.t. every (sub)word accepted by traversing the cycle is also expressible by the loop label. Therefore, ε-cycles and nullable bases exactly reflect another and the claim follows. □

Corollary 5.3. *If* μ *is worst-case, then* $|\mu|_O = 0$.

The following proposition might be obvious. Still, it provides a first opportunity to reject a feature of worst-case expressions by constructing a worse expression without that feature.

Proposition 5.4. *If* μ *is worst-case, then* $|\mu|_\varnothing = 0$.

Proof. Let μ be worst-case and suppose $\varnothing \in \mathrm{sub}(\mu)$. Let x be a letter and set $\mu' := \mu[x/\varnothing]$ for a fixed occurrence of $\varnothing$ in μ. While at some point in the construction of A_μ an $\varnothing$-transition will be removed, this is not the case for the corresponding

	$\Rightarrow_\varnothing$	$\Rightarrow_\bullet$	$\Rightarrow_+$	$\Rightarrow_*$	$\Rightarrow_X$	$\Rightarrow_Y$	$\Rightarrow_Z$	$\Rightarrow_O$
$\Delta(\lvert Q\rvert)$	0	1	0	1	-1	-1	-1	$-(n-1)$
$\Delta(\lvert\delta\rvert)$	-1	1	1	2	0	-1	-1	$-n$

Table 1: Changes in number of states and transitions resulting from each single conversion. In the case of O-elimination, n denotes the size of the ε-cycle.

x-transition in the construction of $A_{\mu'}$. The number of times any other expansion can be applied remains constant, and the number of times each elimination can be applied in the construction of $A_{\mu'}$ is at most as much as for A_μ. We thus find $|A_{\mu'}| \geq |A_\mu|+1$, and since $|\mu'| = |\mu|$, the claim follows. □

A preliminary upper bound on conversion ratio follows almost directly from the definition of conversion.

Lemma 5.5. *The conversion ratio of any expression α is bounded from above, as follows:*

$$\mathrm{c}(\alpha) \;\leq\; \frac{5}{3} + \frac{8}{3|\alpha|}.$$

Proof. Let μ be worst-case. Then $|\mu|_O = |\mu|_\varnothing = 0$ according to Cor. 5.3 and Prop. 5.4. Other than that, we start with $|A^0_\mu| = 3$ and add the elements contributed or removed by the remaining conversions according to Tab. 1. This yields

$$\begin{aligned} |A_\mu| = \;& 2|\mu|_\bullet + |\mu|_+ + 3|\mu|_* - |\mu|_X - 2|\mu|_Y - 2|\mu|_Z + 3 \\ = \;& |\mu| + |\mu|_\bullet + 2|\mu|_* - |\mu|_\mathcal{A} - |\mu|_X - 2|\mu|_Y - 2|\mu|_Z + 3 \\ = \;& |\mu| - |\mu|_+ + 2|\mu|_* - |\mu|_X - 2|\mu|_Y - 2|\mu|_Z + 2 \end{aligned}$$

where we use $|\alpha|_\mathcal{A} = |\alpha|_\bullet + |\alpha|_+ + 1$ in the second step. Since μ is in SNF, we know from Lem. 4.5 that $|\mu|_* \leq \frac{1}{3}(|\mu|+1)$. Omitting the negative terms we arrive at

$$|A_\mu| \;\leq\; |\mu| + 2|\mu|_* + 2 \;\leq\; |\mu| + \frac{2}{3}(|\mu|+1) + 2 \;=\; \frac{5}{3}|\mu| + \frac{8}{3}$$

Dividing the left- and right-sides by $|\mu|$ yields an upper bound for $\mathrm{c}(\mu)$, which in turn bounds the conversion ratio of any expression. □

In certain cases this bound provides a criterion to decide which of two expressions is worse if this is not obvious at a glance.

Corollary 5.6. *Let μ and ν be expressions s.t. $|\mu| \geq 3$, $|\nu| = |\mu| + k$ and $|A_\nu| = |A_\mu| + l$ for $k, l \in \mathbb{N}$. Then ν is worse than μ if*

$$\frac{l}{k} \geq 2.6$$

Proof. Let μ and ν be as stated. By definition, ν is worse than μ if $\mathrm{c}(\mu) < \mathrm{c}(\nu)$. Written out, this inequality is

$$\frac{|A_\mu|}{|\mu|} < \frac{|A_\mu| + l}{|\mu| + k} \text{ , which holds iff } \mathrm{c}(\mu) < \frac{l}{k}$$

For $|\mu| \geq 3$ Lem. 5.5 yields

$$\mathrm{c}(\mu) \leq \frac{5}{3} + \frac{8}{9} = 2.\overline{5} < 2.6$$

Therefore, if $\frac{l}{k}$ is at least 2.6, it exceeds the conversion ratio of μ, and the claim follows. □

Lemma 5.7. *If μ is worst-case and $|\mu| \geq 3$, then $|\mu|_\varepsilon = 0$.*

Proof. Let μ be worst-case with $|\mu|_\varepsilon > 0$. Fix some ε in μ and let t be the ε-transition which is labeled with our fixed ε. We distinguish whether at some point in the construction t is part of an elimination anchor.

- If t can be removed by some elimination, let $\mu' := \mu[x/\varepsilon]$ for some letter x. Then the expansions for μ and μ' are the same, whereas at lest one elimination less is applied upon constructing $A_{\mu'}$. So $|A_{\mu'}| < |A_\mu|$ and $|\mu'| = |\mu|$, therefore μ' is worse than μ.
- If t does not occur in an elimination anchor, we consider the parent of ε in μ. Since μ is worst-case, it is in SNF, so $p_\varepsilon(\mu) \neq *$. Also, $p_\varepsilon(\mu) \neq \bullet$, or t would be part of an Y-anchor. Only $p_\varepsilon(\mu) = +$ remains possible. We set $\mu' := \mu[x^*/\varepsilon]$, then the ε-transitions introduced by expanding the corresponding x^*-transition are not part of any elimination-anchors, just as was the case with t. In this case, $|\mu'|_* = |\mu|_* + 1$, while all other conversions are applied the same number of times. With $|A_{\mu'}| = |A_\mu| + 3$ and $|\mu'| = |\mu| + 1$ we apply Cor. 5.6 finding that μ' is worse than μ in this case, too.

□

We associate to $\beta \in \mathrm{sub}(\alpha)$ the first star encountered in the parse of α on the upwards path from the root of β to the root of α. Let the *minimal containing base* of $\tau \in \mathrm{sub}(\alpha)$, denoted $\mathrm{mcb}_\alpha(\tau)$, be the smallest $\kappa \in \mathrm{sub}(\alpha)$ s.t. $\tau \in \mathrm{sub}(\kappa)$ and $p_\alpha(\kappa) = *$. If no such κ exists, $\mathrm{mcb}_\alpha(\tau)$ is undefined. Formally that is

$$\mathrm{mcb}_\alpha(\tau) = \begin{cases} \tau, & \text{if } p_\alpha(\tau) = *; \\ \mathrm{mcb}_\alpha(\beta), & \text{if } \beta \in \{\tau\tau', \tau'\tau, \tau + \tau'\}; \\ \text{undefined}, & \text{if } \tau = \alpha. \end{cases}$$

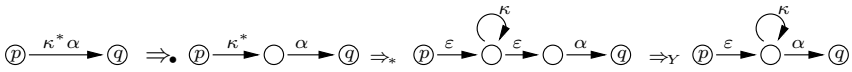
Lemma 5.8. *In a worst-case expression every iteration is an addend.*

Proof. Let μ be worst-case and $\kappa^* \in \mathrm{sub}(\mu)$. Since $\varepsilon \in L(\kappa^*)$ and μ is in SNF, we get $p_\mu(\kappa^*) \neq *$ from Lem. 4.4. If κ^* is a factor, we choose a smallest such iteration: suppose $p_{\kappa^*}(\mu) = \bullet$ and $p_{\varkappa^*}(\kappa) = +$ for all $\varkappa^* \in \mathrm{sub}(\kappa)$. We further assume that κ^* is a factor in the product $\kappa^*\alpha \in \mathrm{sub}(\mu)$ (the case $\alpha\kappa^*$ is symmetric). Set $\mu' := \mu[\kappa^* + \alpha/\kappa^*\alpha]$ and note that $|\mu|$ and $|\mu'|$ are equal. We distinguish whether μ' is in SNF.

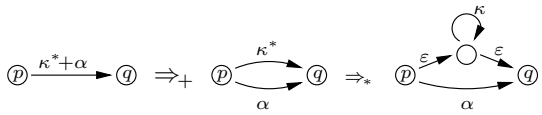
- If μ' is in SNF, we examine how p or q can be removed in the construction of $A_{\mu'}$ (see Fig. 9). First, Lem. 5.2 yields $|\mu'|_O = 0$. If either of p or q is removed by means of X-, Y-, or Z-elimination, this state, possibly along with some additional transitions, can be removed in the construction of A_μ, too. Therefore $|A'_\mu| > |A_\mu|$ while $|\mu'| = |\mu|$, hence μ' is worse than μ.

- If μ' is not in SNF, then Lem. 4.4 yields $\varepsilon \in L(\mathrm{mcb}_{\mu'}(\kappa^*))$. Since this is not the case in μ, i.e., $\varepsilon \notin L(\mathrm{mcb}_\mu(\kappa^*))$, yet obviously $\varepsilon \in L(\kappa^*)$, this implies $\varepsilon \notin L(\alpha)$. Let $\tau = \mathrm{mcb}_{\mu'}(\kappa^*)[{}^{\alpha^*}\!/_{\alpha}]$ and set $\mu'' := \mu'[{}^{\tau}\!/_{\mathrm{mcb}_{\mu'}(\kappa^*)}]$. Intuitively, μ'' is derived from μ' by shifting the parent of $\mathrm{mcb}_{\mu'}(\kappa^*)$ onto α. Now $|\mu''| = |\mu|$ and μ'' is in SNF. Let op_1 be the root of $\mathrm{mcb}_\mu(\kappa^*)$ and assume that μ is not an iteration. Then $\mathrm{mcb}_\mu(\kappa^*)^*$ is an operand to some op_2, the root of $\nu \in \mathrm{sub}(\mu)$. Let A^k_μ be such that ν is the label of a transition $t = (p, \nu, q)$. Then there exists some $A^k_{\mu''}$ which differs from A^k_μ only inasmuch as that t is replaced by $t'' = (p, \nu'', q)$, where ν'' is the replacement subexpression in μ'' constructed above. This gives at least one additional ε-transition in the automaton (Fig. 10), so μ'' is worse than μ.

In either case, the assumption that an iteration in a worst-case expression may be a factor is falsified, so the statement follows. □



(a) Part of an FA arising from a starred factor.



(b) Increase in size from exchanging • for $+$.

Figure 9: First case in the proof of Lem. 5.8: $\kappa^*\alpha$ is replaced with $\kappa^* + \alpha$

The lemma implies that in a worst-case expression, the number of stars is at most twice the number of sum-operators. This allows us to improve the bound given in Lem. 5.5 to the value given by Ilie & Yu [6].

Lemma 5.9. *The conversion ratio of any expression α is bounded from above, as follows:*

$$\mathrm{c}(\alpha) \leq \frac{3}{2} + \frac{5}{2|\alpha|}.$$

Proof. Let μ be worst-case and proceed as in the proof of Lem. 5.5, where we arrived at

$$|A_\mu| \leq |\mu| - |\mu|_+ + 2|\mu|_* + 2 \quad \text{and} \quad |\mu|_* \leq \frac{1}{3}(|\mu| + 1)$$

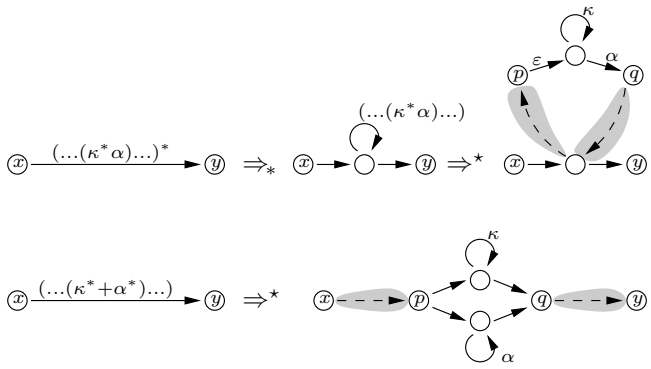


Figure 10: Second case in the proof of Lem. 5.8: shifting the star of $\mathrm{mcb}_\mu(\alpha)$ onto α yields a bigger automaton; the replacement inside the strong subautomaton based in the state z is the same as in Fig. 9.

As we have already observed, Lem. 5.8 implies $|\mu|_+ \geq \frac{1}{2}|\mu|_*$. Plugging this into above inequations yields

$$|A_\mu| \;\leq\; |\mu| + \frac{3}{2}|\mu|_* + 2 \;\leq\; \frac{3}{2}|\mu| + \frac{5}{2}$$

Dividing by $|\mu|$ yields an upper bound for the conversion ratio of a worst-case expression, which bounds the conversion ratio of any expression. □

As before, this upper bound allows us to compare certain expressions that differ in size and are converted to automata that differ in size, too. The proof is the same as for Cor. 5.6, except for the obvious modifications.

Corollary 5.10. *Let μ and ν be expressions s.t. $|\mu| \geq 16$, $|\nu| = |\mu| + k$ and $|A_\nu| = |A_\mu| + l$. Then ν is worse than μ if*

$$\frac{l}{k} > 1.66.$$

This stronger version of Cor. 5.6 is necessary to show that none of the other eliminations occur in the conversion of a worst-case expression. We can apply the same kind of proof with smaller increments in FA size relative to expression size. In particular, the criterion applies if an increment of expression size by 3 increases the size of the corresponding FA by at least 5.

An inconvenience of Cor. 5.10 is its restriction to expressions of size at least 16. All statements building upon the corollary (and their respective proofs) must include a clause similar to "...and suppose further that $|\mu| \geq 16$...". Since this property holds for almost all expressions, and we seek to infer an infinite family of worst-case expressions, this shortcoming is not severe and will be ignored in the remainder of the analysis.

Lemma 5.11. *If μ is worst-case, then $|\mu|_Y = 0$.*

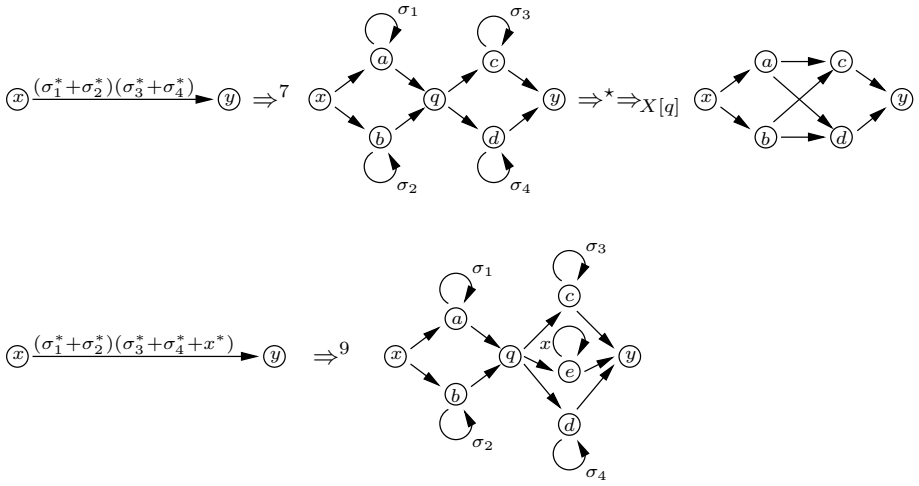


Figure 11: Proof of Lem. 5.12. Adding x^* to a sum of the xy-label introduces four new elements and prevents X-elimination of q (ε-labels are omitted).

Proof. If μ is worst-case then $|\mu|_\varepsilon = 0$ by Lem. 5.7, so the ε-transition appearing in a Y-anchor is introduced by $\Rightarrow_*$. In order to get an Y-anchor from $\Rightarrow_*$, a starred factor is required. Following Lem. 5.8, these do not occur in a worst-case expression. □

Lemma 5.12. *If μ is worst-case, then $|\mu|_X = 0$.*

Proof. As noted in the proof of Lem. 5.11, ε-transitions result from star expansion. In particular, an X-anchor results from a subexpression $\chi = (\sigma_1^* + \sigma_2^*)(\sigma_3^* + \sigma_4^*)$. Let x be a letter and derive ν from μ by replacing χ with $(\sigma_1^* + \sigma_2^*)(\sigma_3^* + \sigma_4^* + x^*)$. The expression size is $|\nu| = |\mu| + 3$, and for automaton size we get $|A_\nu| = |A_\mu| + 5$. This latter increase results from two additional expansions and one prevented X-elimination (see Fig. 11). Applying Cor. 5.10 yields that ν is worse than μ. □

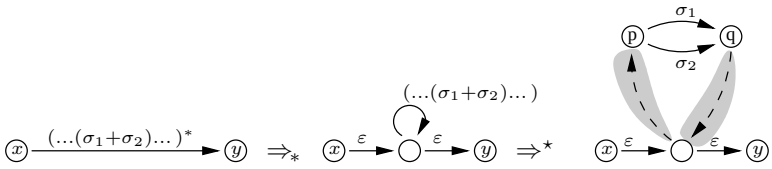
Lemma 5.13. *If μ is worst-case, then $|\mu|_Z = 0$.*

Proof. Let Z_μ contain a $Z[q]$-anchor where all leaving transitions are ε-transitions. Let $t = (p, a, q)$ be the one transition entering q. We replace t's label in μ by setting $\nu := \mu[{}^{a(b+c^*)}/_a]$ for $b, c \in \mathcal{A}$. Then $|A_\nu| = |A_\mu| + 8$, since disabling $Z[q]$-elimination saves a state and a transition, and 6 new elements are required. With $|\nu| = |\mu| + 5$, Cor. 5.6 yields that ν is worse than μ. □

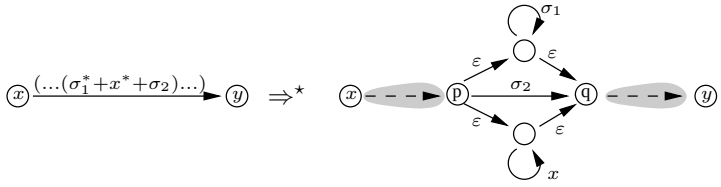
Since no eliminations occur in the conversion of a worst-case expression, the inequality derived in the proof of Lem. 5.9 becomes an equality: if μ is worst-case, then

$$|A_\mu| = |\mu| + 2|\mu|_* - |\mu|_+ + 2.$$

Thus, fixing the size of a worst-case expression leaves the sums and stars as the sole parameters determining the size of the resulting FA. We narrow the structural properties



(a)



(b)

Figure 12: Second case in the proof of Lem. 5.14: the non-nullable addend σ_1 is replaced by two nullable addends, σ_1^* and x^*, which yields a worse expression.

of such expressions further down by investigating the interrelations between those two operators.

Lemma 5.14. *If μ is worst-case then every addend in μ is nullable.*

Proof. Let μ be worst-case and suppose μ contains an addend which is not nullable, say, $\sigma_1 + \sigma_2 \in \mathrm{sub}(\mu)$ where $\varepsilon \notin L(\sigma_1)$. Let $\mu' := \mu[{}^{\sigma_1^*+x^*}/_{\sigma_1}]$ for some letter x and distinguish whether μ' is in SNF.

1. If μ' is in SNF, then $|A_{\mu'}| = |A_\mu| + 7$, while $|\mu'| = |\mu| + 4$. It follows from Cor. 5.10 that μ' is worse than μ.
2. If μ' is not in SNF, replacing σ_1 with $\sigma_1^*+x^*$ introduced a nullable base. Since σ_1 is not nullable by assumption and x is a letter, the base in question is necessarily $\mathrm{mcb}_{\mu'}(\sigma_1^* + x^*)$. To unclutter notation we set $\gamma = \mathrm{mcb}_{\mu'}(\sigma_1^* + x^*)$. The "nullability" of every base besides γ is the same in μ and μ'. We remove the star operating on γ by setting $\mu'' = \mu'[\gamma/\gamma^*]$. Now no base of μ'' is nullable, so μ'' is in SNF. Since no eliminations occur in the construction of A_μ and $A_{\mu''}$, the differences between A_μ and $A_{\mu''}$ are restricted to the subautomaton emerging from expansion of $(x, \mathrm{mcb}_\mu(\sigma_1)^*, y)$ in A^i_μ resp. (x, γ, y) in $A^i_{\mu''}$, as sketched in Fig. 12.
   The sizes of the expressions are related by the equality $|\mu''| = |\mu| + 3$, that of their respective automata by $|A_{\mu''}| = |A_\mu| + 5$. The statement now follows from Cor. 5.10.

Since we find an expression which is worse than μ in either case, every addend in a worst-case expression is nullable. □

Lemma 5.15. *Every maximal sum in a worst-case expression is star-maximal.*

Proof. Let μ be worst-case and suppose μ contains maximal sums that are not star-maximal. We choose a smallest such sum: let $\sigma = \sum \sigma_i$ be maximal but not star-maximal and assume that all maximal sums that are proper subexpressions of σ are star-maximal. Let σ_k be an addend which is not an iteration. Since σ is maximal, σ_k is not a sum. Since σ_k is nullable by Lem. 5.14, it is not a letter either. Therefore σ_k must be a product, which we suppose to be maximal. Commutativity of sums allows us to assume $k = 1$. We are thus looking at

$$\sigma = \sigma_1 + \sum_{i=2}^{n} \sigma_i = \prod_{i=1}^{m} \pi_i + \sum_{i=2}^{n} \sigma_i.$$

As σ_1 is nullable, so is every π_i, and $|\mu|_\varepsilon = 0$ implies that this is due to iterations that occur in each π_i. By our choice of σ and since each iteration in μ is an addend (Lem. 5.8), it follows that every π_i contains sums which are all star-maximal. Since σ_1 is assumed to be a maximal product, each π_i is necessarily such a sum. In greater detail, σ admits the structure

$$\sigma = \prod_{i=1}^{m} \sum_{j=1}^{l_i} \varsigma_{ij}^* + \sum_{i=2}^{n} \sigma_i.$$

Depending on the number n of addends in σ we construct an expression which is worse than μ. Note that σ is a proper sum, so we have $n \geq 2$.

1. $n = 2$: Since $\sigma = \sigma_1 + \sigma_2$ is maximal, σ_2 is not a sum itself, and since σ_2 is nullable (Lem. 5.14) it is not a literal. This leaves two possibilities:

   (a) If σ_2 is an iteration, $\sigma_2 = \kappa^*$, we construct σ' from σ as

   $$\sigma' := (x \bullet \sigma_1)^* + \sigma_2 = (x(\prod \sum \varsigma_{ij}^*))^* + \kappa^*,$$

   and let $\mu' := \mu[\sigma'/\sigma]$. Every elimination in the construction of A'_μ has a counterpart in the construction of A_μ, so we only need to compare the difference in positive contributions to FA size: looking up Tab. 1 we find $|A_{\mu'}| = |A_\mu| + 5$. Since we have $|\mu'| = |\mu| + 3$, Cor. 5.10 applies to yield the statement.

   (b) If σ_2 is a (maximal) product, again Lem. 5.14 implies that its factors are star-maximal sums, i.e.,

   $$\sigma_2 = \prod_{i=1}^{m'} \sum_{j=1}^{l'_i} \varsigma_{ij}^* \quad \text{, which we write as} \quad \sigma_2 = \prod_{i=m+1}^{m+m'} \sum_{j=1}^{l_i} \varsigma_{ij}^*.$$

   We construct the product σ' from σ by exchanging its main operator for a concatenation-symbol and introducing a new starred addend in the first factor of σ_2:

   $$\sigma' = (\prod_{i=1}^{m} \sum_{j=1}^{l_i} \varsigma_{ij}^*)(x^* + \sum_{j=1}^{l_{m+1}} \varsigma_{1j}^*)(\prod_{i=m+2}^{m+m'} \sum_{j=1}^{l_i} \varsigma_{ij}^*)$$

With this, we set $\mu' := \mu[\sigma'/\sigma]$; exchanging the sum for a product introduces an additional state q. No new eliminations become possible from this replacement; in particular, the extra addend x^* in σ' ensures $q^+ \geq 3$, so no $X[q]$-anchor emerges. We find $|\mu'| = |\mu| + 3$ and $|A_{\mu'}| = |A_\mu| + 5$, so μ' is worse than μ according to Cor. 5.10.

2. $n \geq 3$: Again we exchange a sum for a product by setting

$$\sigma' = \sigma_1 \bullet (x^* + \sum_{i=2}^{n} \sigma_i).$$

As before, the product introduces a new state while the extra addend x^* prevents the eventuality of X-elimination. This yields $|A_{\mu'}| = |A_\mu| + 5$ and $|\mu'| = |\mu| + 3$, so Cor. 5.10 yields the claim.

□

Lemma 5.16. *In a worst-case expression every letter is starred.*

Proof. Let μ be worst-case and fix an occurrence of the letter x in μ. If x is a base, we are done. Since all sums in μ are star-maximal (Lem. 5.15), x is not an addend. Assume x is a factor π_k in a maximal product $\pi = \prod \pi_i$, then there is at least one other factor π_{k-1} or π_{k+1} next to x. Suppose wlog. that this is π_{k+1}. Since π is maximal, π_{k+1} is not a product itself, and since all iterations are addends, it is not an iteration either. So π_{k+1} is either a (maximal) sum or a letter.

- If π_{k+1} is a sum, it is star-maximal, i.e., $\pi_{k+1} = \sum_i \varsigma_i^*$. The subexpression $x \sum_i \varsigma_i^*$ would lead to Z-anchors, contradicting Lem. 5.13.
- If π_{k+1} is a letter y, our usual argument shows that $\nu := \mu[x+y^*/xy]$ is worse than μ.

□

Theorem 5.17. *The structure of a worst-case expression is*

$$\prod_{i=1}^{n} \sum_{j=1}^{2+i \bmod 2} x_{ij}^*$$

where $n \in \mathbb{N}$ is arbitrary and the x_{ij} are letters.

Proof. Let μ be a worst case expression. As Lem. 5.16 states, every letter is a base in μ, and since μ is in SNF, no more bases occur by virtue of Lem. 4.5. Every maximal sum in μ is star-maximal according to Lem. 5.15, so the structure of the i-th maximal sum σ_i in μ is

$$\sigma_i = \sum_{j=1}^{k_i} x_{ij}^* \text{ for } x_{ij} \in \mathcal{A}$$

Every σ_i is maximal, so some of those sums are factors of the same maximal product π. Since π is maximal, it is not a factor itself, since all addends are iterations in μ, π is

not an addend either, and as we argued before, π is not a base. Therefore, all maximal sums of μ are factors of this product, which implies that μ itself is π. So the basic structure of μ is

$$\mu = \prod_{i=1}^{n} \sum_{j=1}^{k_i} x_{ij}^*.$$

In this general case, and subject to the condition that no X-eliminations occur upon conversion, we compare the sizes of μ and A_μ, which are

$$\begin{aligned} |\mu| =& (n-1) + \sum_{i=1}^{n} (3k_i - 1) = 3\sum_{i=1}^{n} k_i - 1, \text{ and} \\ |A_\mu| =& \sum_{i=1}^{n} 4k_i + n - 1 = 4\sum_{i=1}^{n} k_i + n - 1, \end{aligned}$$

leading to the following conversion ratio:

$$\mathrm{c}(\mu) = \frac{|A_\mu|}{|\mu|} = \frac{4\sum k_i + n - 1}{3\sum k_i - 1} = 1 + \frac{\sum k_i + n}{3\sum k_i - 1}.$$

The term on the right shows that $\mathrm{c}(\mu)$ is maximal iff n is maximal with respect to $\sum k_i$, or equivalently, if $\sum k_i$ is minimal for fixed n. All sums in μ are proper, i.e., not unary, so $k_i \geq 2$ for all i. Simply setting $k_i = 2$ for all i introduces an X-anchor in X_μ for every pair of adjacent factors. We get rid of X-anchors if k_i alternates between 2 an 3, i.e., $k_i = i \bmod 2 + 1$ (cf. Fig. 11). It is easily seen using the pigeonhole-principle that any smaller $\sum k_i$ leads to X-anchors. □

Corollary 5.18. *Any expression α can be converted to a unique normalized FA A s.t.*

$$|A| \leq \frac{22}{15}|\alpha| + 2 = 1.4\overline{6}|\alpha| + 2$$

Proof. Let μ be a smallest worst-case expression according Thm. 5.17 whose size is at least that of a given expression α. Let n denote the number of factors in μ. We then have $|\mu| = 15n - 1$ and $|A_\mu| = 22n + 1$. Thus the conversion ratio for μ is

$$\mathrm{c}(\mu) = \frac{22n+1}{15n-1} = \frac{22}{15} + \frac{37}{15|\mu|}$$

With $|\alpha| \leq |\mu|$ and $\mathrm{c}(\alpha) \leq \mathrm{c}(\mu)$ we find

$$\begin{aligned} |A_\alpha| = \mathrm{c}(\alpha)|\alpha| \leq \mathrm{c}(\mu)|\alpha| &= \frac{22}{15}|\alpha| + \frac{37}{15|\mu|}|\alpha| \leq \frac{22}{15}|\alpha| + \frac{37}{15} \\ &< \frac{22}{15}|\alpha| + 3 \end{aligned}$$

Therefore $|A_\alpha| \leq \frac{22}{15}|\alpha| + 2$, and setting $A = A_\alpha$ yields the claim. □

This finishes our analysis conversion ratio, i.e., the size of an FA relative to the size of the expression it is derived from. Let us stress again that the conversion ratio of an expression reaches the bound given in Cor. 5.18 iff the expression admits the structure given in Thm. 5.17.

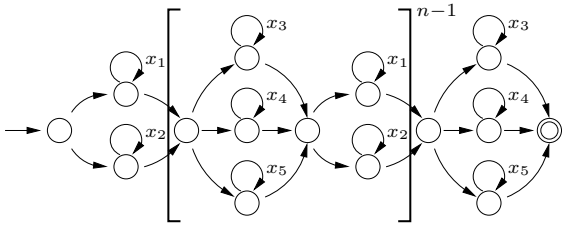


Figure 13: Automaton constructed from μ_n (ε-labels are omitted).

6. Optimality of the Construction

We provide a lower bound on conversion ratio, i.e., one that holds for *any* construction. Our argument is again based on the digraph underlying an FA.

Proposition 6.1. *Let L and R be disjoint sets of vertices in a digraph G, s.t. the following path-conditions are satisfied*

1. *There is a path from each $l \in L$ to every $r \in R$ in G.*
2. *There is no path between any two vertices of L, nor any two vertices of R.*
3. *There is no path from any $r \in R$ to any $l \in L$.*

Then G contains at least $\min\{|L||R|, |L| + |R| + 1\}$ additional elements

Proof. Suppose G contains a path from each $l \in L$ to every $r \in R$. Note that these paths need not be disjoint. We distinguish whether there is a vertex on some path from L to R

1. If there is no such vertex on any path, the members of L and R are pairwise adjacent — thus, at least $|L||R|$ additional arcs are present in G.
2. If a vertex x occurs on a path P from $l \in L$ to $r \in R$, then $x \notin L \cup R$, or the third path-condition would be violated. Now P contains at least three elements, x and two arcs. Consider the remaining vertices of L and R: every $r' \in R \setminus r$ is the endpoint of a path from l, so every such r' is the endpoint of an arc $a_{r'}$. Likewise, every $l' \in L \setminus l$ is the tail of an arc $a_{l'}$. By above path-properties, no pair $a_{r'}$, $a_{l'}$ coincides. Hence G contains $3+(|L|-1)+(|R|-1) = |L|+|R|+1$ additional elements.

□

We investigate the FAs built from a family of expressions with worst-case structure according to Thm. 5.17. These expressions are defined over an alphabet of size 5, which is the least size for which the proof of the following Lemma works.

$$\mu_n := \prod_{i=1}^{n} (x_1^* + x_2^*)(x_3^* + x_4^* + x_5^*)$$

The FA constructed from μ_n by our construction is sketched in Fig. 13. In the proof of the following, we consider maximal repetitions of a letter in a word. Given a word $w = w_1 x^n w_2$, s.t. $w_1 \neq w_1' x$ and $w_2 \neq x w_2'$, we call x^n an *x-block*, or just *block*, of w. For example, the blocks of $aaabba$ are aaa, bb and a.

Lemma 6.2. *The size of a normalized FA accepting $L(\mu_n)$ is at least $22n + 1$.*

Proof. Let A be a normalized FA accepting $L(\mu_n)$. In particular, A accepts all words of $L(\mu_n)$ that contain the maximal number of blocks, which is $2n$. Every such word admits the structure

$$w = b_1 b_2 \cdots b_{2n},$$

where b_i is an x_1- or x_2-block if i is odd, and an x_3-, x_4- or x_5-block if i is even. A block can be arbitrarily long, so it must be read in a cycle. Let $\gamma_{i,j}$ denote the cyclic part of A reading a block b_i which is an x_j-block and let $q_{i,j}$ be a state of $\gamma_{i,j}$.
First we show that if $\gamma_{i,j}$ reads a block b_i of $w \in L(\mu_n)$ and $\gamma_{i+k,j'}$ reads a block appearing afterwards in w, then A contains a path from $q_{i,j}$ to $q_{i+k,j}$ but no path back: Since both cycles are involved in an accepting run of A on w, there is a path from $\gamma_{i,j}$ to $\gamma_{i+k,j'}$, hence also from $q_{i,j}$ to $q_{i+k,j'}$. Now assume that A also contains a path from $q_{i+k,j}$ back to $q_{i,j'}$. If $j \neq j'$, meaning that $\gamma_{i,j}$ and $\gamma_{i+k,j'}$ read blocks consisting of distinct letters, then the number of x_j- and $x_{j'}$-blocks in a word accepted by A is unbounded, since an accepting path can go back and forth between the two cycles, possibly reading some interspersed subwords along the way. In case $j = j'$, i.e., both blocks are x_j-blocks, some $x_{j''}$-block b_{i+l} occurs between b_i and b_{i+k} in w, s.t. $j \neq j''$ and $1 \leq l < k$. Again, this would imply that the number of blocks in a word accepted by A is unbounded. In either case the existence of such a path violates the property that the number of blocks in a word of $L(\mu_n)$ is bounded, hence no backward path exists.
Next we show that there is no path between a pair of cycles $\gamma_{i,j}$ and $\gamma_{i,j'}$ that allow for the i-th block to be an x_j- resp. an $x_{j'}$-block. Assume A contains a path from $q_{i,j}$ to $q_{i,j'}$, then a word could be accepted as follows: first, i blocks are read on a run from the initial state to $q_{i,j}$, then the path to $q_{i,j'}$ is taken, then an $x_{j'}$-block is read and finally $2n - i$ more blocks from $q_{i,j'}$ to the unique final state. This adds up to $2n + 1$ blocks, contradicting the bound on the number of blocks in $w \in L(\mu_n)$. Therefore, no such path exists.
The properties above imply that each distinct pair $\gamma_{i,j}$, $\gamma_{i',j'}$ is disjoint. Considering $\gamma_{i,j}$, there are 2 values possible for j if i is even and 3 values if i is odd. Therefore, there are at least $5n$ distinct cyclic structures in A. Every such structure consists of at least a state and a transition, hence A contains no less than $10n$ elements to read the single blocks. We have further shown that A contains a path from $\gamma_{i,j}$ to $\gamma_{i',j'}$ *iff* $i < i'$; transitive reduction shows that this requires at least a path from $q_{i,j}$ to $q_{i+1,j'}$. Assume i is even (the case that i is odd is symmetric), then the sets of states $L = \{q_{i,1}, q_{i,2}\}$ and $R = \{q_{i+1,3}, q_{i+1,4}, q_{i+1,5}\}$ satisfy the conditions of Prop. 6.1. So A contains at least 6 additional elements for each $1 \leq i \leq n - 1$ to realize reachability among the $\gamma_{i,j}$. Moreover there is a separate initial state q_0 with paths from q_0 to $\gamma_{1,1}$ and $\gamma_{1,2}$, as well as a separate final state q_f with paths from $\gamma_{n,3}$, $\gamma_{n,4}$ and $\gamma_{n,5}$ to q_f; every such path has at least one transition. Thus A contains at least $22n + 1$ elements. □

Corollary 6.3. *Let A be any normalized FA accepting μ_n. Then the size of A is bounded from below as*

$$|A| \geq 1.4\overline{6}|\mu_n| + 2$$

Proof. If A is a normalized FA accepting $L(\mu_n)$ Lem. 6.2 yields $|A| \geq 22n+1$ which relates to $|\mu_n| = 15n-1$ as follows

$$|A| \geq 22n+1 = \frac{22}{15}|\mu_n| + \frac{22}{15} + 1 \geq 1.4\overline{6}|\mu_n| + 2$$

□

7. Implementation Details

We propose three preprocessing-steps on an input, all of which can be realized in a bottom-up fashion on the parse of an expression in linear time. These steps considerably reduce the effort necessary to implement EFA-conversions, as will be discussed below.

Remove all occurrences of $\varnothing$: It is well-known that any expression either describes the empty language or can be made free of $\varnothing$, i.e., either $L(\alpha) = \{\}$ or $|\alpha|_\varnothing = 0$. This is realized by rewriting an expression according to the following rules:

- replace $\alpha\varnothing$ and $\varnothing\alpha$ with $\varnothing$
- replace $\alpha + \varnothing$ and $\varnothing + \alpha$ with α
- replace $\varnothing^*$ with ε

Remove redundant occurrences of ε: Every ε that does not occur as an addend in an otherwise non-nullable sum can be removed. Applying the following rewritings in a bottom-up fashion to an expression removes all instances of ε as mentioned, without altering the denoted language.

- replace $\alpha\varepsilon$ and $\varepsilon\alpha$ with α
- replace $\alpha + \varepsilon$ and $\varepsilon + \alpha$ with α, if α is nullable
- replace ε^* with ε

Let $reduce(\rho)$ denote the expression yielded from applying above preprocessings in their given order to ρ, the resulting expression is called *reduced*. The removal of ε in a sum is not reflected by the EFA-conversions; still, it reduces the size of the final FA by at lest one and possibly allows for additional Y-eliminations. Notice that this step does not influence worst-case analysis.

Compute the star normal form: since SNF is implicitly realized by the given conversions, it can as well be computed directly from α.

Let ρ be the SNF of a reduced expression, i.e., $\rho = (reduce(\rho'))^\bullet$. Then ρ comes with several properties that can be exploited in order to reduce code complexity and absolute running time of an implementation (asymptotic running time is not affected).

1. If $|\rho|_\varnothing > 0$, either $\rho = \varnothing$, in which case a trivial FA is returned, whereas otherwise, $\varnothing$ may occur as an addend. This is due to the fact that $\varnothing$ can be introduced again by computing the SNF, as seen with the example

$$((x + \varepsilon)^*)^\bullet = (x + \varnothing)^*$$

   In a reduced expression, ε only occurs as an addend; this carries over to $\varnothing$ after computing the SNF. Thus $\varnothing$-elimination can be anticipated by immediately discarding $\varnothing$-transitions that emerge from sum expansion. Moreover, every state of the resulting FA lies on some accepting path, so cleaning up useless states needs not be implemented.
2. As we argued in Lem. 5.2, an expression in SNF does not lead to O-anchors upon conversion. Hence, the detection and elimination of ε-cycles needs not be implemented. Still, note that ε-cycles can be detected in linear time, as discussed by Ilie & Yu [6].
3. We eliminate Y-anchors as soon as they appear. This saves the effort of bookkeeping or later searches. An Y-anchor emerges with the introduction of an ε-transition $t = (p, \varepsilon, q)$ where $p^+ = 1$ or $q^- = 1$. For preprocessed inputs, ε-transitions only arise from sum- or star expansion; however, in the case of sum expansion the degree constraints are certainly not satisfied. Therefore, it suffices to check right after star expansion whether one of the involved states is central to an Y-anchor, and if so, apply Y-elimination.

Starting from A^0_α, for preprocessed α, the algorithm proceeds in three phases according to Cor. 3.4. First, exhaustive expansion with embedded Y-elimination computes X_α. Then an elimination order that satisfies **(X2)** is computed on the X-anchors, resp. their central states. The subgraph of X_α consisting of these states and the ε-transitions among them is acyclic, therefore it is sufficient to compute a topological sort on this subgraph (see e.g. [14]). Eliminating the X-anchors in descending topological order yields Z_α. Finally, Z-elimination is carried out respecting **(Z2)**; this is realized analogous to X-elimination.

In Alg. 1, Q_X denotes the states that are central to X-anchors and $\prec_X$ denotes a topological order on Q_X. Note that a single X-elimination can remove several adjacent X-anchors (see Fig. 4b), so when choosing a$\prec_X$-minimal state q, we need to test whether $X[q]$-elimination is still applicable. Since no new X-anchors are introduced by X-elimination, these actions suffice to compute Z_α through valid conversions. This process is repeated for Z-elimination which has the same properties in that regard.

Each step of the algorithm can be implement in time linear in the size of the input expression. Thus A_α can be constructed in $\mathcal{O}(|\alpha|)$.

8. Conclusions

We presented a construction of finite automata from regular expressions by means of EFA-rewritings. This system is a refinement of a very early method proposed by Ott & Feinstein in 1961. Our version comes with a set of rules that decrease the size of intermediates in the construction by replacing substructures that are rich in ε-transition with smaller equivalents.

Algorithm 1: Proposed implementation with preprocessing.

Input: Regular expression ρ over $\mathcal{A}$, s.t. $L(\rho) \neq \emptyset$
Output: Finite automaton A accepting $L(\rho)$
$\rho \leftarrow \mathrm{reduce}(\rho)$
$\rho \leftarrow \rho^{\bullet}$
$A \leftarrow (\{q_0, q_f\}, \mathcal{A}, (q_0, \rho, q_f), q_0, q_f)$
while A *has complex labels* **do**
  choose and remove complex transition $t = (p, \tau, q)$
  switch τ **do**
    case $\alpha\beta$
      add state r
      add transitions (p, α, r), (r, β, q)
    end
    case $\alpha + \beta$
      if $\alpha \neq \varnothing$ **then** add transition (p, α, q)
      if $\beta \neq \varnothing$ **then** add transition (p, β, q)
    end
    case α^*
      add state r
      add transitions (p, ε, r), (r, α, r), (r, ε, q)
      if $p^+ = 1$ **then** apply $Y[p]$-elimination
      if $q^- = 1$ **then** apply $Y[q]$-elimination
    end
  end
end
compute $Q_X \subseteq Q_A$ and $\prec_X$
while $Q_X \neq \emptyset$ **do**
  $Q_X \leftarrow Q_X \setminus q$ for $\prec_X$-minimal q
  if $q^- = q^+ = 2$ **then** apply $X[q]$-elimination
end
compute $Q_Z \subseteq Q_A$ and $\prec_Z$
while $Q_Z \neq \emptyset$ **do**
  $Q_Z \leftarrow Q_Z \setminus q$ for $\prec_Z$-minimal $q \in Q_Z$
  if $Z[q]$*-elimination is applicable* **then** apply $Z[q]$-elimination
end

We showed that the resulting automata are unique if a partial order is enforced on the applicability of conversions. A further feature of our construction is that the output is invariant under taking the star-normal-form of the input. Both properties are crucial in providing a minute comparison of input- to output size.

The size of automata constructed by our method was analyzed by inferring the structure of expressions that maximize the ratio of input- to output size. These expressions are enumerated by repeating an atomic subexpression; their conversion ratio, for all practical purposes, is $\frac{22}{15}$, or $1.4\overline{6}$. For every expression our construction therefore provides an automaton whose size is at most $1.4\overline{6}$ times the size of the expression. It was shown that this value is tight for expressions over alphabets of size at least five, which makes the construction optimal. It remains an open question whether this bound is tight for smaller alphabets, too.

We finally proposed some modifications of the formal rewriting system with respect to an actual implementation. These consist of a series of preprocessing steps and several shortcuts in the construction. In effect, these modifications reduce the effort to detect or keep track of certain elimination anchors, which considerably reduces code complexity.

[1] S. C. Kleene, Representation of events in nerve nets and finite automata, Annals of Mathematics Studies, 3–41, 1956.

[2] B. W. Watson, A taxonomy of finite automata construction algorithms, Tech. Rep. Computing Science Note 93/43, Eindhoven University of Technology, URL `citeseer.ist.psu.edu/watson94taxonomy.html`, 1994.

[3] J. E. Friedl, Mastering Regular Expressions, O'Reilly, 3 edn., 2006.

[4] K. Ellul, B. Krawetz, J. Shallit, M.-W. Wang, Regular expressions: new results and open problems, Journal of Automata, Languages and Combinatorics 10 (4) (2005) 407–437.

[5] M. Holzer, M. Kutrib, Descriptional Complexity — An Introductory Survey, in: C. Martín-Vide (Ed.), Scientific Applications of Language Methods, 1–58, 2010.

[6] L. Ilie, S. Yu, Follow automata, Information and Computation (186) (2003) 140–162.

[7] S. Gulan, H. Fernau, An Optimal Construction of Finite Automata from Regular Expressions, in: R. Hariharan, M. Mukund, V. Vinay (Eds.), 28th International Conference on Foundations of Software Technology and Theoretical Computer Science, 211–222, 2008.

[8] V. M. Glushkov, The abstract theory of automata, Russian Mathematical Surveys 16 (1961) 1–53.

[9] R. McNaughton, H. Yamada, Regular Expressions and State Graphs for Automata, IRE Transactions on Electronic Computers 9 (1) (1960) 39–47.

[10] A. Brüggemann-Klein, Regular expressions into finite automata, Theoretical Computer Science 120 (1993) 197–312.

[11] G. Ott, N. H. Feinstein, Design of Sequential Machines from Their Regular Expressions, Journal of the ACM 8 (4) (1961) 585–600.

[12] M. Newman, On theories with a combinatorial definition of ”equivalence”, Annals of Mathematics 43 (2) (1942) 223–243.

[13] G. Huet, Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems, Journal of the ACM 27 (4) (1980) 797–821.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, 3 edn., 2009.