Reordering Derivatives of Trace Closures of Regular Languages

Hendrik Maarand

Department of Software Science, Tallinn University of Technology, Estonia
hendrik@cs.ioc.ee

Tarmo Uustalu

School of Computer Science, Reykjavik University, Iceland
Department of Software Science, Tallinn University of Technology, Estonia
tarmo@ru.is

Abstract

We provide syntactic derivative-like operations, defined by recursion on regular expressions, in the styles of both Brzozowski and Antimirov, for trace closures of regular languages. Just as the Brzozowski and Antimirov derivative operations for regular languages, these syntactic reordering derivative operations yield deterministic and nondeterministic automata respectively. But trace closures of regular languages are in general not regular, hence these automata cannot generally be finite. Still, as we show, for star-connected expressions, the Antimirov and Brzozowski automata, suitably quotiented, are finite. We also define a refined version of the Antimirov reordering derivative operation where parts-of-derivatives (states of the automaton) are nonempty lists of regular expressions rather than single regular expressions. We define the uniform scattering rank of a language and show that, for a regexp whose language has finite uniform scattering rank, the truncation of the (generally infinite) refined Antimirov automaton, obtained by removing long states, is finite without any quotienting, but still accepts the trace closure. We also show that star-connected languages have finite uniform scattering rank.

2012 ACM Subject Classification Theory of computation → Regular languages; Theory of computation → Concurrency

Keywords and phrases Mazurkiewicz traces, trace closure, regular languages, finite automata, language derivatives, scattering rank, star-connected expressions

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2019.40

Related Version A full version of this article is available as a preprint [12], `https://arxiv.org/abs/1908.03551`.

Funding *Both authors:* supported by the Estonian Ministry of Education and Research institutional grant no. IUT33-13.
Hendrik Maarand: supported by the ERDF funded Estonian national CoE project EXCITE (2014-2020.4.01.15-0018).

Acknowledgements We thank Pierre-Louis Curien, Jacques Sakarovitch, Simon Doherty, Georg Struth and Ralf Hinze for inspiring discussions, and our anonymous reviewers for the exceptionally thorough and constructive feedback they gave us.

1 Introduction

Traces were introduced to concurrency theory by Mazurkiewicz [13, 14] as an alternative to words. A word can be seen as a linear order that is labelled with letters of the alphabet. Intuitively, the main idea of traces is that the linear order, corresponding to sequentiality, is replaced with a partial order. Sets of words (or word languages) can be used to describe the behaviour of concurrent systems. Similarly, sets of traces (or trace languages) can also be used for this purpose. The difference is that descriptions in terms of traces do not distinguish between different linear extensions (words) of the same partial order (trace) – they are considered equivalent. Different linear extensions of the same partial order can be seen as different observations of the same behaviour.

Given a word language L and a letter a, the derivative of L along a is the language consisting of all the words v such that av belongs to L. An essential difference between words and traces is that a nonempty word (a linear order) has its first letter as the unique minimal element, but a nonempty trace (a partial order) may have several minimal elements. A trace from a trace language can be derived along any of its minimal letters. Clearly, a minimal letter of a trace need not be the first letter of a word representing this trace.

It is well-known that the derivative of a regular word language along a letter is again regular. Brzozowski [5] showed that a regexp for it can be computed from a regexp for the given language, and Antimirov [2] then further optimized this result. We show that these syntactic derivative operations generalize to trace closures (i.e., closures under equivalence) of regular word languages in the form of syntactic *reordering derivative* operations.

The syntactic derivative operations for regular word languages provide ways to construct automata from a regexp. The Brzozowski derivative operation is a function on regexps while the Antimirov derivative operation is a relation. Accordingly, they yield deterministic and nondeterministic automata. The set of Brzozowski derivatives of a regexp (modulo appropriate equations) and the set of Antimirov parts-of-derivatives are finite, hence so are the resulting automata. Our generalizations to trace closures of regular languages similarly give deterministic and nondeterministic automata, but these cannot be finite in general. Still, as we show, for a star-connected expression, the Antimirov and Brzozowski automata, suitably quotiented, are finite. We also develop a finer version of the Antimirov reordering derivative, where parts-of-derivatives are nonempty lists of regexps rather than single regexps, and we show that the set of expressions that can appear in these lists for a given initial regexp is finite. We introduce a new notion of *uniform scattering rank* of a language (a variant of Hashiguchi's scattering rank [7]) and show that, for a regexp whose language has finite uniform rank, a truncation of the refined reordering Antimirov automaton accepts its trace closure despite the removed states, and is finite, without any quotienting.

A full version of this article, with proofs and background material on classical language derivatives and trace closures of regular languages is available as a preprint [12].

2 Preliminaries on Word Languages

An *alphabet* Σ is a finite set (of *letters*). A *word* over Σ is a finite sequence of letters. The set Σ^* of all words over Σ is the free monoid on Σ with the empty word ε as the unit and concatenation of words (denoted by $\cdot$ that can be omitted) as the multiplication. We write $|u|$ for the length of a word u and also $|X|$ for the size of a subalphabet X. By $|u|_a$ we mean the number of occurrences of a in u. By $\Sigma(u)$ we denote the set of letters that appear in u.

A *(word) language* is a subset of Σ^*. The empty word and concatenation of words lift to word languages via $\mathbf{1} =_{\mathrm{df}} \{\varepsilon\}$ and $L \cdot L' =_{\mathrm{df}} \{uv \mid u \in L \wedge v \in L'\}$.

2.1 Derivatives of a Language

A word language L is said to be *nullable* $L{\downarrow}$, if $\varepsilon \in L$. The *derivative* (or *left quotient*)[1] of L along a word u is defined by $D_u L =_{\mathrm{df}} \{v \mid uv \in L\}$. For any L, we have $D_\varepsilon L = L$ as well as $D_{uv} L = D_v(D_u L)$ for any $u, v \in \Sigma^*$, i.e., the operation $D : \mathcal{P}\Sigma^* \times \Sigma^* \to \mathcal{P}\Sigma^*$ is a right action of Σ^* on $\mathcal{P}\Sigma^*$. We also have $L = \{\varepsilon \mid L{\downarrow}\} \cup \bigcup\{\{a\} \cdot D_a L \mid a \in \Sigma\}$, and for any $u \in \Sigma^*$, we have $u \in L$ iff $(D_u L){\downarrow}$.

[1] We use the word "derivative" both for languages and expressions, reserving the word "quotient" for quotients of sets by equivalence relations.

2.2 Regular Languages

The set RE of *regular expressions* (in short, *regexps*) over Σ is given by the grammar $E, F ::= a \mid 0 \mid E + F \mid 1 \mid EF \mid E^*$ where a ranges over Σ.

The *word-language semantics* of regular expressions is given by a function $[\![\_]\!] : \mathsf{RE} \to \mathcal{P}\Sigma^*$ defined recursively by

$$\begin{array}{rcl@{\qquad}rcl}
[\![a]\!] & =_{\mathrm{df}} & \{a\} & [\![1]\!] & =_{\mathrm{df}} & \mathbf{1} \\
[\![0]\!] & =_{\mathrm{df}} & \emptyset & [\![EF]\!] & =_{\mathrm{df}} & [\![E]\!] \cdot [\![F]\!] \\
[\![E+F]\!] & =_{\mathrm{df}} & [\![E]\!] \cup [\![F]\!] & [\![E^*]\!] & =_{\mathrm{df}} & \mu X.\, \mathbf{1} \cup [\![E]\!] \cdot X
\end{array}$$

A word language L is said to be *regular* (or *rational*) if $L = [\![E]\!]$ for some regexp E. Kleene algebras are defined by an equational theory. It was shown by Kozen [11] that the set $\{[\![E]\!] \mid E \in \mathsf{RE}\}$ of all regular languages together with the language operations $\emptyset$, $\cup$, $\mathbf{1}$, $\cdot$, $(\_)^*$ is the free Kleene algebra on Σ. An important property for us is that $E \doteq F$ iff $[\![E]\!] = [\![F]\!]$ where $\doteq$ refers to valid equations in the Kleene algebra theory.

Kleene's theorem [9] says that a word language is rational iff it is *recognizable*, i.e., accepted by a finite deterministic automaton (acceptance by a finite nondeterministic automaton is an equivalent condition because of determinizability [17]).

2.3 Brzozowski and Antimirov Derivatives

Derivatives of regular languages are regular. A remarkable fact is that they can be computed syntactically, on the level of regular expressions. There are two constructions for this, due to Brzozowski [5] and Antimirov [2]. The Brzozowski and Antimirov derivative operations yield deterministic resp. nondeterministic automata accepting the language of a regular expression E. The Antimirov automaton is finite. The Brzozowski automaton becomes finite when quotiented by associativity, commutativity and idempotence for $+$. Identified up to the Kleene algebra theory, the states of the Brzozowski automaton correspond to the derivatives of the language $[\![E]\!]$. Regular languages can be characterized as languages with finitely many derivatives.

3 Trace Closures of Regular Languages

3.1 Trace Closure of a Word Language

An *independence alphabet* is an alphabet Σ together with an irreflexive and symmetric relation $I \subseteq \Sigma \times \Sigma$ called the *independence* relation. The complement D of I, which is reflexive and symmetric, is called *dependence*. We extend independence to words by saying that two words u and v are independent, uIv, if aIb for all a, b such that $a \in \Sigma(u)$ and $b \in \Sigma(v)$.

Let $\sim^I \subseteq \Sigma^* \times \Sigma^*$ be the least congruence relation on the free monoid Σ^* such that aIb implies $ab \sim^I ba$ for all $a, b \in \Sigma$. If uIv, then $uv \sim^I vu$.

A *(Mazurkiewicz) trace* is an equivalence class of words wrt. $\sim^I$. The equivalence class of a word w is denoted by $[w]^I$.

A word $a_1 \ldots a_n$ where $a_i \in \Sigma$ yields a directed node-labelled acyclic graph as follows. Take the vertex set to be $V =_{\mathrm{df}} \{1, \ldots, n\}$ and label vertex i with a_i. Take the edge set to be $E =_{\mathrm{df}} \{(i, j) \mid i < j \wedge a_i D a_j\}$. This graph (V, E) for a word w is called the *dependence graph* of w and is denoted by $\langle w \rangle_D$. If $w \sim^I z$, then the dependence graphs of w and z are isomorphic, i.e., traces can be identified with dependence graphs up to isomorphism.

The set $\Sigma^*/{\sim^I}$ of all traces is the free partially commutative monoid on (Σ, I). If $I = \emptyset$, then $\Sigma^*/{\sim^I} \cong \Sigma^*$, the set of words, i.e., we recover the free monoid. If $I = \{(a, b) \mid a \neq b\}$, then $\Sigma^*/{\sim^I} \cong \mathcal{M}_{\mathrm{f}}(\Sigma)$, the set of finite multisets over Σ, i.e., the free commutative monoid.

A *trace language* is a subset of $\Sigma^*/{\sim^I}$. Trace languages are in bijection with word languages that are *(trace) closed* in the sense that, if $z \in L$ and $w \sim^I z$, then also $w \in L$. If T is a trace language, then its flattening $L =_{\mathrm{df}} \bigcup T$ is a closed word language. On the other hand, the trace language corresponding to a closed word language L is $T =_{\mathrm{df}} \{t \in \Sigma^*/{\sim^I} \mid \exists z \in t.\, z \in L\} = \{t \in \Sigma^*/{\sim^I} \mid \forall z \in t.\, z \in L\}$.

Given a general (not necessarily closed) word language L, we define its *(trace) closure* $[L]^I$ as the least closed word language that contains L. Clearly $[L]^I = \{w \in \Sigma^* \mid \exists z \in L.\, w \sim^I z\}$ and also $[L]^I = \bigcup\{t \in \Sigma^*/{\sim^I} \mid \exists z \in t.\, z \in L\}$. For any L, we have $[[L]^I]^I = [L]^I$, so $[\_]^I$ is a closure operator. Note also that L is closed iff $[L]^I = L$.

As seen in Section 2.1, the derivative of a word language is the set of all suffixes for a prefix. We now look at what the prefixes and suffixes of a word as a representative of a trace should be. For a word vuv' such that vIu, we can consider u to be its prefix, up to reordering, and vv' to be the suffix. This is because an equivalent word uvv' strictly has u as a prefix and vv' as the suffix. Similarly, we may also want to consider u' to be a prefix of vuv' when $u' \sim^I u$ since $u'vv' \sim^I uvv' \sim^I vuv'$. Note that if a is such a prefix of z, then, by irreflexivity of I, this a is the first a of z. In general, when u is a prefix of z, then the letter occurrences in u uniquely map to letter occurrences in z. We scale these ideas to allow u to be scattered in z as $z = v_0u_1v_1 \ldots u_nv_n$ in either the sense that $u = u_1 \ldots u_n$ or $u \sim^I u_1 \ldots u_n$. We also define bounded versions of scattering that become relevant in Section 5.

▶ **Definition 1.** *For all* $u_1, \ldots, u_n \in \Sigma^+, v_0 \in \Sigma^*, v_1 \ldots, v_{n-1} \in \Sigma^+, v_n \in \Sigma^*, z \in \Sigma^*$, $u_1, \ldots, u_n \lhd z \rhd v_0, \ldots, v_n =_{\mathrm{df}} z = v_0u_1v_1 \ldots u_nv_n \wedge \forall i.\, \forall j < i.\, v_jIu_i$.

▶ **Definition 2.** *For all* $u, v, z \in \Sigma^*$,

1. $u \lhd z \rhd v =_{\mathrm{df}} \exists n \in \mathbb{N}, u_1, \ldots, u_n, v_0, \ldots, v_n.\, u = u_1 \ldots u_n \wedge v = v_0 \ldots v_n \wedge$ $u_1, \ldots, u_n \lhd z \rhd v_0, \ldots, v_n$;
2. $u \sim\!\lhd z \rhd v =_{\mathrm{df}} \exists u'.\, u \sim^I u' \wedge u' \lhd z \rhd v$;
3. $u \sim\!\lhd z \rhd\!\sim v =_{\mathrm{df}} \exists u', v'.\, u \sim^I u' \wedge u' \lhd z \rhd v' \wedge v' \sim^I v$.

▶ **Lemma 3.** *For any* $u, v, z \in \Sigma^*$,
$u \lhd z \rhd v \iff \exists! n \in \mathbb{N}, u_1, \ldots, u_n, v_0, \ldots, v_n.\, u = u_1 \ldots u_n \wedge v = v_0 \ldots v_n \wedge$ $u_1, \ldots, u_n \lhd z \rhd v_0, \ldots, v_n$.

▶ **Definition 4.** *For all* $u, v, z \in \Sigma^*$ *and* $N \in \mathbb{N}$,

1. $u \lhd_N z \rhd v =_{\mathrm{df}} \exists n \leq N, u_1, \ldots, u_n, v_0, \ldots, v_n.\, u_1, \ldots, u_n \lhd z \rhd v_0, \ldots, v_n$;
2. *(and* $u \sim\!\lhd_N z \rhd v$ *and* $u \sim\!\lhd_N z \rhd\!\sim v$ *are defined analogously).*

▶ **Example 5.** Let $\Sigma =_{\mathrm{df}} \{a, b, c\}$ and aIb and aIc. Take $z =_{\mathrm{df}} aabcba$. We have $ab \lhd z \rhd acba$ since $a, b \lhd z \rhd \varepsilon, a, cba$. We can visualize this by underlining the subwords of $u =_{\mathrm{df}} ab$ in $z = \varepsilon\underline{a}a\underline{b}cba$. This scattering is valid because εIa, εIb and aIb: recall that Def. 1 requires all underlined subwords u_i to be independent with all non-underlined subwords v_i to their left in z. Similarly we have $aa, a \lhd z \rhd \varepsilon, bcb, \varepsilon$ because $z = \varepsilon\underline{aa}bcb\underline{a}\varepsilon$, εIaa, εIa and $bcbIa$. Note that neither $aa\underline{b}cb\underline{a}\varepsilon$ nor $aabc\underline{ba}\varepsilon$ satisfies the conditions about independence and thus there is no v such that $ba \lhd z \rhd v$. We do have $ba \sim\!\lhd z \rhd acba$ though, since $ba \sim^I ab$ and $a, b \lhd z \rhd \varepsilon, a, cba$.

▶ **Proposition 6.** *For all* $u, v, z \in \Sigma^*, uv \sim^I z \iff u \sim\!\lhd z \rhd\!\sim v$.

3.2 Trace-Closing Semantics of Regular Expressions

We now define a nonstandard word-language semantics of regexps that directly interprets E as the trace closure $[\llbracket E \rrbracket]^I$ of its standard regular word-language denotation $\llbracket E \rrbracket$.

We have $[\{a\}]^I = \{a\}$, $[\emptyset]^I = \emptyset$, $[L \cup L']^I = [L]^I \cup [L']^I$ and $[\mathbf{1}]^I = \mathbf{1}$. But for general I, we do not have $[L \cdot L']^I = [L]^I \cdot [L']^I$. For example, for $\Sigma =_{\mathrm{df}} \{a, b\}$ and aIb, we have $[\{a\}]^I = \{a\}$, $[\{b\}]^I = \{b\}$ whereas $[\{ab\}]^I = \{ab, ba\} \neq \{ab\} = [\{a\}]^I \cdot [\{b\}]^I$. Hence we need a different concatenation operation.

▶ **Definition 7.**

1. *The* I-reordering concatenation of words $\cdot^I : \Sigma^* \times \Sigma^* \to \mathcal{P}\Sigma^*$ *is defined by*

$$\begin{array}{rcl} \varepsilon \cdot^I v & =_{\mathrm{df}} & \{v\} \\ u \cdot^I \varepsilon & =_{\mathrm{df}} & \{u\} \\ au \cdot^I bv & =_{\mathrm{df}} & \{a\} \cdot (u \cdot^I bv) \cup \{b \mid auIb\} \cdot (au \cdot^I v) \end{array}$$

2. *The lifting of I-reordering concatenation to* languages *is defined by*

$$L \cdot^I L' =_{\mathrm{df}} \bigcup \{u \cdot^I v \mid u \in L \wedge v \in L'\}$$

Note that $\{b \mid auIb\}$ acts as a test: it is either $\emptyset$ or $\{b\}$.

▶ **Example 8.** Let $\Sigma =_{\mathrm{df}} \{a, b\}$ and aIb. Then $a \cdot^I b = \{ab, ba\}$, $aa \cdot^I b = \{aab, aba, baa\}$, $a \cdot^I bb = \{abb, bab, bba\}$ and $ab \cdot^I ba = \{abba\}$. The last example shows that although I-reordering concatenation is defined quite similarly to shuffle, it is different.

▶ **Proposition 9.** *For any $u, v, z \in \Sigma^*$, $z \in u \cdot^I v \iff u \lhd z \rhd v$.*

▶ **Proposition 10.** *For any languages L and L', $[L \cdot L']^I = [L]^I \cdot^I [L']^I$.*

Evidently, if $I = \emptyset$, then reordering concatenation is just ordinary concatenation: $u \cdot^\emptyset v = \{uv\}$. For $I = \Sigma \times \Sigma$, which is forbidden in independence alphabets, as I is required to be irreflexive, it is shuffle: $u \cdot^{\Sigma \times \Sigma} v = u \sqcup\!\sqcup v$. For general I, it has properties similar to concatenation. In particular, we have

$$\begin{array}{rclrcl} \mathbf{1} \cdot^I L & = & L & \emptyset \cdot^I L & = & \emptyset \\ L \cdot^I \mathbf{1} & = & L & (L_1 \cup L_2) \cdot^I L & = & L_1 \cdot^I L \cup L_2 \cdot^I L \\ (L \cdot^I L') \cdot^I L'' & = & L \cdot^I (L' \cdot^I L'') & (L_1 \sqcup\!\sqcup L_2) \cdot^I (L_1' \sqcup\!\sqcup L_2') & \subseteq & (L_1 \cdot^I L_1') \sqcup\!\sqcup (L_2 \cdot^I L_2') \end{array}$$

but also other equations of the concurrent Kleene algebra theory introduced in [8].

We are ready to introduce the closing semantics of regular expressions.

▶ **Definition 11.** *The* trace-closing semantics $[\![\_]\!]^I : \mathsf{RE} \to \mathcal{P}\Sigma^*$ *of regular expressions is defined recursively by*

$$\begin{array}{rclrcl} [\![a]\!]^I & =_{\mathrm{df}} & \{a\} & [\![1]\!]^I & =_{\mathrm{df}} & \mathbf{1} \\ [\![0]\!]^I & =_{\mathrm{df}} & \emptyset & [\![EF]\!]^I & =_{\mathrm{df}} & [\![E]\!]^I \cdot^I [\![F]\!]^I \\ [\![E+F]\!]^I & =_{\mathrm{df}} & [\![E]\!]^I \cup [\![F]\!]^I & [\![E^*]\!]^I & =_{\mathrm{df}} & \mu X.\, \mathbf{1} \cup [\![E]\!]^I \cdot^I X \end{array}$$

Compared to the standard semantics of regular expressions, the difference is in the handling of the EF case (and consequently also the E^* case) due to the cross-commutation that happens in concatenation of traces and must be accounted for by $\cdot^I$.

With $I = \emptyset$, we fall back to the standard interpretation of regular expressions: $[\![E]\!]^\emptyset = [\![E]\!]$. For I a general independence relation, we obtain the desired property that the semantics delivers the trace closure of the language of the regexp.

▶ **Proposition 12.** *For any E, $[\![E]\!]^I$ is trace closed; moreover, $[\![E]\!]^I = [[\![E]\!]]^I$.*

3.3 Properties of Trace Closures of Regular Languages

Trace closures of regular languages are theoretically interesting, as they have intricate properties. For a thorough survey, see Ochmański's handbook chapter [16].

The most important property for us is that the trace closure of a regular language is not necessarily regular. Consider $\Sigma =_{\mathrm{df}} \{a, b\}$, aIb. Let $L =_{\mathrm{df}} [\![(ab)^*]\!]$. The language $[L]^I = \{u \mid |u|_a = |u|_b\}$ is not regular.

The class of trace closures of regular languages over an independence alphabet behaves quite differently from the class of regular languages over an alphabet. For example, the class of trace closures of regular languages over (Σ, I) is closed under complement iff I is quasi-transitive (i.e., its reflexive closure is transitive) [3, 1, 18] (cf. [16, Thm. 6.2.5]); the question of whether the trace closure of the language of a regexp over (Σ, I) is regular is decidable iff I is quasi-transitive [19] (cf. [16, Thm. 6.2.7]).

A closed language is regular iff the corresponding trace language is accepted by a finite asynchronous (a.k.a. Zielonka) automaton [21, 22]. In Section 4.4, we will see further characterizations of regular closed languages based on star-connected expressions.

4 Reordering Derivatives

We are now ready to generalize the Brzozowski and Antimirov constructions for trace closures of regular languages. To this end, we switch to what we call reordering derivatives.

4.1 Reordering Derivative of a Language

Let (Σ, I) be a fixed independence alphabet. We generalize the concepts of (semantic) nullability and derivative of a language to concepts of reorderable part and reordering derivative.

▶ **Definition 13.** *We define the* I-reorderable part *of a language L wrt. a word u by* $R^I_u L =_{\mathrm{df}} \{v \in L \mid vIu\}$ *and the* I-reordering derivative *along u by* $D^I_u L =_{\mathrm{df}} \{v \mid \exists z \in L.\, u \sim\!\lhd z \rhd v\}$.

By Prop. 9, we can equivalently say that $D^I_u L = \{v \mid \exists z \in L.\, z \in [u]^I \cdot^I v\}$. For a single-letter word a, we get $D^I_a L = \{v_l v_r \mid v_l a v_r \in L \wedge v_l I a\} = \{v \mid \exists z \in L.\, z \in a \cdot^I v\}$. That is, we require some reordering of u (resp. a) to be a prefix, up to reordering, of some word z in L with v as the corresponding strict suffix. (In other words, for the sake of precision and emphasis, we allow reordering of letters within u and across u and v, but not within v.)

▶ **Example 14.** Let $\Sigma =_{\mathrm{df}} \{a, b, c\}$ and aIb. Take $L =_{\mathrm{df}} \{\varepsilon, a, b, ca, aa, bbb, babca, abbaba\}$. We have $R^I_a L = R^I_{aa} L = \{\varepsilon, b, bbb\}$, $D^I_a L = \{\varepsilon, a, bbca, bbaba\}$ and $D^I_{aa} L = \{\varepsilon, bbba\}$.

In the special case $I = \emptyset$, we have $R^\emptyset_\varepsilon L = L$, $R^\emptyset_u L = \{\varepsilon \mid L\!\downarrow\}$ for any $u \neq \varepsilon$, and $D^\emptyset_u L = D_u L$. In the general case, the reorderable part and reordering derivative enjoy the following properties.

▶ **Lemma 15.** *For every L,*
1. $R^I_\varepsilon L = L$*; for every* $u, v \in \Sigma^*$, $R^I_v(R^I_u L) = R^I_{uv} L$*;*
2. *for every* $u, u' \in \Sigma^*$, $R^I_{\Sigma(u)} L = R^I_{\Sigma(u')} L$.

We extend R^I to subsets of Σ: by $R^I_X L$, we mean $R^I_u L$ where u is any enumeration of X.

▶ **Lemma 16.** *For every L,*
1. $D^I_\varepsilon L = L$*; for any* $u, v \in \Sigma^*$, $D^I_v(D^I_u L) = D^I_{uv} L$*;*
2. *for any* $u, u' \in \Sigma^*$ *such that* $u \sim^I u'$*, we have* $D^I_u L = D^I_{u'} L$.

▶ **Proposition 17.** *For every* L,

1. *for any* $u \in \Sigma^*$, $D_u([L]^I) = [D^I_u L]^I$;
   if L *is closed (i.e.,* $[L]^I = L$*), then, for any* $u \in \Sigma^*$, $D^I_u L$ *is closed and* $D_u L = D^I_u L$;
2. *for any* $u, v \in \Sigma^*$, $uv \in [L]^I$ *iff* $v \in [D^I_u L]^I$;
3. *for any* $u \in \Sigma^*$, $u \in [L]^I$ *iff* $(D^I_u L){\downarrow}$;
4. $[L]^I = \{\varepsilon \mid L{\downarrow}\} \cup \bigcup_{a\in\Sigma} \{a\} \cdot [D^I_a L]^I$.

▶ **Example 18.** Let $\Sigma =_{\mathrm{df}} \{a, b\}$ and aIb. Take L to be the regular language $[\![(ab)^*]\!]$. We have already noted that the language $[L]^I = \{u \mid |u|_a = |u|_b\}$ is not regular. For any $n \in \mathbb{N}$, $D^I_{b^n} L = \{a^n\} \cdot L = [\![a^n(ab)^*]\!]$ whereas $D_{b^n}([L]^I) = \{a^n\} \cdot^I [L]^I = \{u \mid |u|_a = |u|_b + n\}$. We can see that $[L]^I$ has infinitely many derivatives, none of which are regular, and L has infinitely many reordering derivatives, all regular.

4.2 Brzozowski Reordering Derivative

The reorderable parts and reordering derivatives of regular languages turn out to be regular. We now show that they can be computed syntactically, generalizing the classical syntactic nullability and Brzozowski derivative operations [5].

▶ **Definition 19.** *The* I-reorderable part *and the* Brzozowski I-reordering derivative *of a regexp are given by functions* $R^I, D^I : \mathsf{RE} \times \Sigma \to \mathsf{RE}$ *and* $R^I, D^I : \mathsf{RE} \times \Sigma^* \to \mathsf{RE}$ *defined recursively by*

$$
\begin{array}{rcl@{\qquad}rcl}
R^I_a b & =_{\mathrm{df}} & \text{if } aIb \text{ then } b \text{ else } 0 & D^I_a b & =_{\mathrm{df}} & \text{if } a = b \text{ then } 1 \text{ else } 0 \\
R^I_a 0 & =_{\mathrm{df}} & 0 & D^I_a 0 & =_{\mathrm{df}} & 0 \\
R^I_a (E+F) & =_{\mathrm{df}} & R^I_a E + R^I_a F & D^I_a (E+F) & =_{\mathrm{df}} & D^I_a E + D^I_a F \\
R^I_a 1 & =_{\mathrm{df}} & 1 & D^I_a 1 & =_{\mathrm{df}} & 0 \\
R^I_a (EF) & =_{\mathrm{df}} & (R^I_a E)(R^I_a F) & D^I_a (EF) & =_{\mathrm{df}} & (D^I_a E)F + (R^I_a E)(D^I_a F) \\
R^I_a (E^*) & =_{\mathrm{df}} & (R^I_a E)^* & D^I_a (E^*) & =_{\mathrm{df}} & (R^I_a E)^*(D^I_a E)E^* \\[1ex]
R^I_\varepsilon E & =_{\mathrm{df}} & E & D^I_\varepsilon E & =_{\mathrm{df}} & E \\
R^I_{ua} E & =_{\mathrm{df}} & R^I_a (R^I_u E) & D^I_{ua} E & =_{\mathrm{df}} & D^I_a (D^I_u E)
\end{array}
$$

The regexp $R_u E$ is nothing but E with all occurrences of letters dependent with u replaced with 0. The definition of D is more interesting. Compared to the classical Brzozowski derivative, the nullability condition $E{\downarrow}$ in the EF case has been replaced with concatenation with the reorderable part $R^I_a E$, and the E^* case has also been adjusted.

The functions R and D on regexps compute their semantic counterparts on the corresponding regular languages.

▶ **Proposition 20.** *For any* E,

1. *for any* $a \in \Sigma$, $R^I_a [\![E]\!] = [\![R^I_a E]\!]$ *and* $D^I_a [\![E]\!] = [\![D^I_a E]\!]$;
2. *for any* $u \in \Sigma^*$, $R^I_u [\![E]\!] = [\![R^I_u E]\!]$ *and* $D^I_u [\![E]\!] = [\![D^I_u E]\!]$.

▶ **Proposition 21.** *For any* E,

1. *for any* $a \in \Sigma$, $v \in \Sigma^*$, $av \in [\![E]\!]^I \iff v \in [\![D^I_a E]\!]^I$;
2. *for any* $u, v \in \Sigma^*$, $uv \in [\![E]\!]^I \iff v \in [\![D^I_u E]\!]^I$;
3. *for any* $u \in \Sigma^*$, $u \in [\![E]\!]^I \iff (D^I_u E){\downarrow}$.

▶ **Example 22.** Let $\Sigma =_{\mathrm{df}} \{a, b\}$, aIb and $E =_{\mathrm{df}} aa + ab + b$.

$$\begin{array}{rcl} D^I_b E & = & D^I_b aa + D^I_b ab + D^I_b b \\ & = & ((D^I_b a)a + (R^I_b a)(D^I_b a)) + ((D^I_b a)b + (R^I_b a)(D^I_b b)) + D^I_b b \\ & = & (0a + a0) + (0b + a1) + 1 \doteq a + 1 \\[1ex] D^I_b(E^*) & = & (R^I_b E)^*(D^I_b E)E^* \\ & = & (aa + a0 + 0)^*((0a + a0) + (0b + a1) + 1)E^* \doteq (aa)^*(a + 1)E^* \\[1ex] D^I_{bb}(E^*) & \doteq & D^I_b((aa)^*(a + 1)E^*) \doteq (aa)^*(a + 1)(aa)^*(a + 1)E^* \end{array}$$

As with the classical Brzozowski derivative, we can use the reordering Brzozowski derivative to construct deterministic automata. For a regexp E, take $Q^E =_{\mathrm{df}} \{D^I_u E \mid u \in \Sigma^*\}$, $q^E_0 =_{\mathrm{df}} E$, $F^E =_{\mathrm{df}} \{E' \in Q^E \mid E'{\downarrow}\}$, $\delta^E_a E' =_{\mathrm{df}} D^I_a E'$ for $E' \in Q^E$. By Prop. 21, this automaton accepts the closure $[\![E]\!]^I$. But even quotiented by the full Kleene algebra theory, the quotient of Q^E is not necessarily finite, i.e., we may be able to construct infinitely many different languages by taking reordering derivatives. For the regexp from Example 18, we have $D^I_{b^n}((ab)^*) \doteq a^n(ab)^*$, so it has infinitely many Brzozowski reordering derivatives even up to the Kleene algebra theory. This is only to be expected, as the closure $[\![(ab)^*]\!]^I$ is not regular and cannot possibly have an accepting finite automaton.

4.3 Antimirov Reordering Derivative

Like the classical Brzozowski derivative that was optimized by Antimirov [2], the Brzozowski reordering derivative construction can be optimized by switching from functions on regexps to multivalued functions or relations.

▶ **Definition 23.** *The* Antimirov I-reordering parts-of-derivatives *of a regexp along a letter and a word are relations* $\to^I \subseteq \mathsf{RE} \times \Sigma \times \mathsf{RE}$ *and* $\to^{I*} \subseteq \mathsf{RE} \times \Sigma^* \times \mathsf{RE}$ *defined inductively by*

$$\frac{}{a \to^I (a, 1)} \qquad \frac{E \to^I (a, E')}{E + F \to^I (a, E')} \qquad \frac{F \to^I (a, F')}{E + F \to^I (a, F')}$$

$$\frac{E \to^I (a, E')}{EF \to^I (a, E'F)} \qquad \frac{F \to^I (a, F')}{EF \to^I (a, (R^I_a E)F')} \qquad \frac{E \to^I (a, E')}{E^* \to^I (a, (R^I_a E)^* E' E^*)}$$

$$\frac{}{E \to^{I*} (\varepsilon, E)} \qquad \frac{E \to^{I*} (u, E') \quad E' \to^I (a, E'')}{E \to^{I*} (ua, E'')}$$

Here R^I is defined as before. Similarly to the Brzozowski reordering derivative from the previous subsection, the condition $E{\downarrow}$ in the second EF rule has has been replaced with concatenation with $R^I_a E$, and the E^* rule has been adjusted.

Collectively, the Antimirov reordering parts-of-derivatives of a regexp E compute the semantic reordering derivative of the language $[\![E]\!]$.

▶ **Proposition 24.** *For any* E,
1. *for any* $a \in \Sigma$, $D^I_a[\![E]\!] = \bigcup\{[\![E']\!] \mid E \to^I (a, E')\}$;
2. *for any* $u \in \Sigma^*$, $D^I_u[\![E]\!] = \bigcup\{[\![E']\!] \mid E \to^{I*} (u, E')\}$.

▶ **Proposition 25.** *For any* E,
1. *for any* $a \in \Sigma$, $v \in \Sigma^*$, $av \in [\![E]\!]^I \iff \exists E'.\, E \to^I (a, E') \wedge v \in [\![E']\!]^I$;
2. *for any* $u, v \in \Sigma^*$, $uv \in [\![E]\!]^I \iff \exists E'.\, E \to^{I*} (u, E') \wedge v \in [\![E']\!]^I$;
3. *for any* $u \in \Sigma^*$, $u \in [\![E]\!]^I \iff \exists E'.\, E \to^{I*} (u, E') \wedge E'{\downarrow}$.

▶ **Example 26.** Let us revisit Example 22. The Antimirov reordering parts-of-derivatives of E along b are $a1$ and 1:

$$\dfrac{\dfrac{\dfrac{\overline{b \to^I (b,1)}}{ab \to^I (b,a1)}}{ab+b \to^I (b,a1)}}{aa+ab+b \to^I (b,a1)} \qquad \dfrac{\dfrac{\overline{b \to^I (b,1)}}{ab+b \to^I (b,1)}}{aa+ab+b \to^I (b,1)}$$

The Antimirov reordering parts-of-derivatives of E^* along b are therefore $E_b^*(a1)E^*$ and $E_b^*1E^*$ where $E_b =_{\mathrm{df}} R_b^I E = aa+a0+0$. Recall that, for the Brzozowski reordering derivative, we computed $D_b^I E = (0a+a0)+(0b+a1)+1$ and $D_b^I E^* = E_b^*((0a+a0)+(0b+a1)+1)E^*$.

Like the classical Antimirov construction, the Antimirov reordering parts-of-derivatives of a regexp E give a nondeterministic automaton by $Q^E =_{\mathrm{df}} \{E' \mid \exists u \in \Sigma^*.\, E \to^{I*} (u, E')\}$, $I^E =_{\mathrm{df}} \{E\}$, $F^E =_{\mathrm{df}} \{E' \in Q^E \mid E'{\downarrow}\}$, $E' \to^E (a, E'') =_{\mathrm{df}} E' \to^I (a, E'')$ for $E', E'' \in Q^E$. This automaton accepts $[\![E]\!]^I$ by Prop. 25, but is generally infinite, also if quotiented by the full Kleene algebra theory. Revisiting Example 18 again, $(ab)^*$ must have infinitely many Antimirov reordering parts-of-derivatives modulo the Kleene algebra theory since $[\![(ab)^*]\!]^I$ is not regular and cannot have a finite accepting nondeterministic automaton. Specifically, it has $(a0)^*((a1)\ldots((a0)^*((a1)(ab)^*))\ldots) \doteq a^n(ab)^*$ as its single reordering part-of-derivative along b^n.

However, if quotienting the Antimirov automaton for E by some sound theory (a theory weaker than the Kleene algebra theory) makes it finite, then the Brzozowski automaton can also be quotiented to become finite.

▶ **Proposition 27.** *For any E,*
1. *for any $a \in \Sigma$, $D_a^I E \doteq \sum\{E' \mid E \to^I (a, E')\}$;*
2. *for any $u \in \Sigma^*$, $D_u^I E \doteq \sum\{E' \mid E \to^{I*} (u, E')\}$*

(using the semilattice equations for $0, +$, that 0 is zero, and distributivity of $\cdot$ over $+$).

▶ **Corollary 28.** *If some quotient of the Antimirov automaton for E (accepting $[\![E]\!]^I$) is finite, then also some quotient of the Brzozowski automaton is finite.*

4.4 Star-Connected Expressions

Star-connected expressions are important as they characterize regular closed languages. A corollary of that is a further characterization of such languages in terms of a "concurrent" semantics of regexps that interprets Kleene star nonstandardly as "concurrent star".

▶ **Definition 29.** *A word $w \in \Sigma^*$ is connected if its dependence graph $\langle w \rangle_D$ is connected. A language $L \subseteq \Sigma^*$ is connected if every word $w \in L$ is connected.*

▶ **Definition 30.**
1. Star-connected expressions *are a subset of the set of all regexps defined inductively by:* 0, 1 *and $a \in \Sigma$ are star-connected. If E and F are star-connected, then so are $E + F$ and EF. If E is star-connected and $[\![E]\!]$ is connected, then E^* is star-connected.*
2. *A language L is said to be* star-connected *if $L = [\![E]\!]$ for some star-connected regexp.*

Ochmański [15] proved that a closed language is regular iff it is the closure of a star-connected language (cf. [16, Thm. 6.3.13]). This means that, for any regexp E, the language $[\![E]\!]^I$ is regular iff there exists a (generally different!) star-connected expression E' such that $[\![E]\!]^I = [\![E']\!]^I$. As a corollary, a closed language is regular iff it is the closure of the concurrent denotation of some regexp (cf. [16, Thm. 6.3.16]).

4.5 Automaton Finiteness for Star-Connected Expressions

We now show that the set of Antimirov reordering parts-of-derivatives of a star-connected expression is finite modulo suitable equations.

▶ **Lemma 31.** *If $[\![E]\!]$ is connected, then, for every $u \in \Sigma^+$ and E' such that $E \to^{I*} (u, E')$, either $R^I_u E' \doteq 0$ or $R^I_u E' \doteq 1$ (using the equations involving 0 and 1 only and that 0 is zero).*

▶ **Lemma 32.** *If $[\![E]\!]$ is connected and $E^* \to^{I*} (u, E')$, then there exists E'' such that $E' \doteq E''$ (using, additionally, the monoid equations for $1, \cdot$ and the equation $F^* \cdot F^* \doteq F^*$) and E'' contains at most $|\Sigma|$ subexpressions of the form $(R^I_X E)^*$ where $\emptyset \subset X \subseteq \Sigma$.*

▶ **Proposition 33.** *If E is star-connected, then a suitable sound quotient of the state set $\{E' \mid \exists u \in \Sigma^*.\ E \to^{I*} (u, E')\}$ of the Antimirov automaton for E (accepting $[\![E]\!]^I$) is finite.*

5 Uniform Scattering Rank of a Language

We proceed to defining the notion of uniform scattering rank of a language and show that star-connected expressions define languages with uniform scattering rank.

5.1 Scattering Rank vs. Uniform Scattering Rank

The notion of scattering rank of a language (a.k.a. distribution rank, k-block testability) was introduced by Hashiguchi [7].

▶ **Definition 34.** *A language L has* (I-scattering) rank *at most N if*
$\forall u, v.\ uv \in [L]^I \implies \exists z \in L.\ u \sim\!\lhd_N z \rhd\!\sim v$.

Hashiguchi [7] showed that having rank is a sufficient condition for regularity of the trace closure of a regular language (cf. [16, Prop. 6.3.2]). But it is not a necessary condition: for $\Sigma =_{\mathrm{df}} \{a, b\}$, aIb, the language $L =_{\mathrm{df}} [\![(aa + ab + ba + bb)^*]\!]$ has rank 1 (as does any nontrivial closed language), but it is not star-connected.

We wanted to show that a truncation of the refined Antimirov automaton (which we define in Section 6) is finite for regexps whose language has rank (i.e., the language has rank at most N for some $N \in \mathbb{N}$). But it turns out, as we shall see, that rank does not quite work for this. For this reason, we introduce a stronger notion that we call uniform scattering rank.

▶ **Definition 35.** *A language L has* uniform (I-scattering) rank *at most N if*
$\forall w \in [L]^I.\ \exists z \in L.\ \forall u, v.\ w = uv \implies u \sim\!\lhd_N z \rhd\!\sim v$.

The difference between the two definitions is that, in the uniform case, the choice of z depends only on w whereas, in the non-uniform case, it depends on the particular split of w as $w = uv$, i.e., for every such split of w we may choose a different z.

▶ **Lemma 36.** *If L has uniform rank at most N, then L has rank at most N.*

The converse of the above lemma does not hold – there are languages with uniform rank greater than rank. Furthermore, there are languages that have rank but no uniform rank.

▶ **Proposition 37.** *Let $\Sigma =_{\mathrm{df}} \{a, b, c\}$, aIb and $E =_{\mathrm{df}} a^*b^*c(ab)^*(a^*+b^*)+(ab)^*(a^*+b^*)ca^*b^*$. The language $[\![E]\!]$ has rank 2, but no uniform rank.*

5.2 Star-Connected Languages Have Uniform Rank

Klunder et al. [10] established that star-connected languages have rank. We will now show that star-connected languages also have uniform rank, by strengthening their proof.

It can be seen that if L_1 and L_2 have uniform rank at most N_1 and N_2, then $L_1 \cup L_2$ has uniform rank at most $\max(N_1, N_2)$ and $L_1 \cdot L_2$ has uniform rank at most $N_1 + N_2$. If a general L has uniform rank at most N, then L^* need not have uniform rank. For example, for $\Sigma =_{\mathrm{df}} \{a, b\}$, aIb, the language $\{ab\}$ has uniform rank 1, but $\{ab\}^*$ is without rank, so also without uniform rank. But if L is also connected, then L^* has uniform rank at most $(N + 1) \cdot |\Sigma|$.

▶ **Proposition 38.** *If E is star-connected, then the language $[\![E]\!]$ has uniform rank.*

6 Antimirov Reordering Derivative and Uniform Rank

We have seen that the reordering language derivative $D^I_u L$ allows u to be scattered in a word $z \in L$ as $u_1, \ldots, u_n \lhd z \rhd v_0, \ldots, v_n$ where $u \sim^I u_1 \ldots u_n$. We will now consider a version of the Antimirov reordering derivative operation that delivers lists of regexps for the possible $v_0, \ldots, v_n$ rather than just single regexps for their concatenations $v_0 \ldots v_n$.

6.1 Refined Antimirov Reordering Derivative

The refined reordering parts-of-derivative of a regexp E along a letter a are pairs of regexps E_l, E_r. For any word $w = av \in [\![E]\!]^I$, there must be an equivalent word $z = v_l a v_r \in [\![E]\!]$. Instead of describing the words $v_l v_r$ obtainable by removing a minimal occurrence of a in a word $z \in [\![E]\!]$, the refined parts-of-derivative describe the subwords v_l, v_r that were to the left and right of this a in z: it must be the case that $v_l \in [\![E_l]\!]$ and $v_r \in [\![E_r]\!]$ for one of the pairs E_l, E_r. For a longer word u, the refined reordering derivative operation gives lists of regexps $E_0, \ldots, E_n$ fixing what the lists of subwords $v_0, \ldots, v_n$ can be in words $z = v_0 u_1 v_1 \ldots u_n v_n \in [\![E]\!]$ equivalent to a given word $w = uv \in [\![E]\!]^I$.

▶ **Definition 39.** *The* (unbounded and bounded) refined Antimirov I-reordering *parts-of-derivatives of a regexp along a letter and a word are given by relations* $\rightarrow^I \subseteq \mathsf{RE} \times \Sigma \times \mathsf{RE} \times \mathsf{RE}$, $\Rightarrow^I \subseteq \mathsf{RE}^+ \times \Sigma \times \mathsf{RE}^+$, $\rightarrow^{I*} \subseteq \mathsf{RE} \times \Sigma^* \times \mathsf{RE}^+$, $\Rightarrow^I_N \subseteq \mathsf{RE}^{+\leq N+1} \times \Sigma \times \mathsf{RE}^{+\leq N+1}$, *and* $\rightarrow^{I*}_N \subseteq \mathsf{RE} \times \Sigma^* \times \mathsf{RE}^{+\leq N+1}$ *defined inductively by*

$$\frac{}{a \rightarrow^I (a; 1, 1)} \qquad \frac{E \rightarrow^I (a; E_l, E_r)}{E + F \rightarrow^I (a; E_l, E_r)} \qquad \frac{F \rightarrow^I (a; F_l, F_r)}{E + F \rightarrow^I (a; F_l, F_r)}$$

$$\frac{E \rightarrow^I (a; E_l, E_r)}{EF \rightarrow^I (a; E_l, E_r F)} \qquad \frac{F \rightarrow^I (a; F_l, F_r)}{EF \rightarrow^I (a; (R^I_a E) F_l, F_r)} \qquad \frac{E \rightarrow^I (a; E_l, E_r)}{E^* \rightarrow^I (a; (R^I_a E)^* E_l, E_r E^*)}$$

$$\frac{E \rightarrow^I (a; E_l, E_r) \quad |\Gamma, \Delta| < N}{\Gamma, E, \Delta \Rightarrow^I_N (a; R^I_a \Gamma, E_l, E_r, \Delta)} \qquad \frac{E \rightarrow^I (a; E_l, E_r) \quad E_l{\downarrow} \quad |\Gamma| > 0}{\Gamma, E, \Delta \Rightarrow^I_N (a; R^I_a \Gamma, E_r, \Delta)}$$

$$\frac{E \rightarrow^I (a; E_l, E_r) \quad E_r{\downarrow} \quad |\Delta| > 0}{\Gamma, E, \Delta \Rightarrow^I_N (a; R^I_a \Gamma, E_l, \Delta)} \qquad \frac{E \rightarrow^I (a; E_l, E_r) \quad E_l{\downarrow} \quad E_r{\downarrow} \quad |\Gamma| > 0 \quad |\Delta| > 0}{\Gamma, E, \Delta \Rightarrow^I_N (a; R^I_a \Gamma, \Delta)}$$

$$\frac{}{E \rightarrow^{I*}_N (\varepsilon; E)} \qquad \frac{E \rightarrow^{I*}_N (u; \Gamma) \quad \Gamma \Rightarrow^I_N (a; \Gamma')}{E \rightarrow^{I*}_N (ua; \Gamma')}$$

By $\mathsf{RE}^{+\leq N+1}$ *we mean nonempty lists of regexps of length at most $N+1$. The relations* $\Rightarrow^I$ *and* $\rightarrow^{I*}$ *are defined exactly as* $\Rightarrow^I_N$ *and* $\rightarrow^{I*}_N$ *but with the condition* $|\Gamma, \Delta| < N$ *of the first rule of* $\Rightarrow^I_N$ *dropped. The operation* R^I_a *is extended to lists of regexps in the obvious way.*

We have several rules for deriving a list of regexps along a. If E is split into E_l, E_r and neither of them is nullable, then, in the N-bounded case, we require that the given list is shorter than $N+1$ since the new list will be longer by 1. If one of E_l, E_r is nullable, not the first resp. last in the list and we choose to drop it, then the new list will be of the same length. If both are nullable, not the first resp. last and we opt to drop both, then the new list will be shorter by 1. They must be droppable under these conditions to handle the situation when a word z has been split as $v_0 u_1 v_1 \dots u_k v_k u_{k+1} \dots u_n v_n$ and v_k is further being split as $v_l a v_r$ while v_l or v_r is empty. If $k \neq 0$ and v_l is empty, we must join u_k and a into $u_k a$. If $k \neq n$ and v_r is empty, we must join a and u_{k+1} into $a u_{k+1}$. If k is neither 0 nor n and both v_l and v_r are empty, we must join all three of u_k, a and u_{k+1} into $u_k a u_{k+1}$. The length of the new list of regexps is always at least 2.

▶ **Proposition 40.** *For any E,*

1. *for any $a \in \Sigma, v_l, v_r \in \Sigma^*$,*

$$v_l I a \wedge v_l a v_r \in [\![E]\!] \iff \exists E_l, E_r.\, E \to^I (a; E_l, E_r) \wedge v_l \in [\![E_l]\!] \wedge v_r \in [\![E_r]\!];$$

2. *for any $u \in \Sigma^*, n \in \mathbb{N}, v_0 \in \Sigma^*, v_1, \dots, v_{n-1} \in \Sigma^+, v_n \in \Sigma^*$,*

$$\begin{aligned}&\exists z \in \Sigma^*, u_1, \dots, u_n \in \Sigma^+.\, z \in [\![E]\!] \wedge u \sim^I u_1 \dots u_n \wedge u_1, \dots, u_n \lhd z \rhd v_0, \dots, v_n\\ &\iff\\ &\exists E_0, \dots, E_n.\, E \to^{I*} (u; E_0, \dots, E_n) \wedge \forall j.\ v_j \in [\![E_j]\!].\end{aligned}$$

▶ **Proposition 41.** *For any E,*

1. *for any $a \in \Sigma, v \in \Sigma^*$, the following are equivalent:*
   - **a.** $av \in [\![E]\!]^I$;
   - **b.** $\exists v_l, v_r \in \Sigma^*.\, v \sim^I v_l v_r \wedge v_l I a \wedge v_l a v_r \in [\![E]\!]$;
   - **c.** $\exists v_l, v_r \in \Sigma^*.$
     $v \sim^I v_l v_r \wedge \exists E_l, E_r.\, E \to^I (a; E_l, E_r) \wedge v_l \in [\![E_l]\!] \wedge v_r \in [\![E_r]\!]$;
   - **d.** $\exists v_l, v_r \in \Sigma^*.$
     $v \in v_l \cdot^I v_r \wedge \exists E_l, E_r.\, E \to^I (a; E_l, E_r) \wedge v_l \in [\![E_l]\!]^I \wedge v_r \in [\![E_r]\!]^I$.
2. *for any $u, v \in \Sigma^*$, the following are equivalent:*
   - **a.** $uv \in [\![E]\!]^I$;
   - **b.** $\exists z \in [\![E]\!].\ u \sim\lhd z \rhd\sim v$;
   - **c.** $\exists n \in \mathbb{N}, v_0 \in \Sigma^*, v_1, \dots, v_{n-1} \in \Sigma^+, v_n \in \Sigma^*.\, v \sim^I v_0 v_1 \dots v_n \wedge$
     $\exists E_0, \dots, E_n.\, E \to^{I*} (u; E_0, \dots, E_n) \wedge \forall j.\, v_j \in [\![E_j]\!]$;
   - **d.** $\exists n \in \mathbb{N}, v_0 \in \Sigma^*, v_1, \dots, v_{n-1} \in \Sigma^+, v_n \in \Sigma^*.\, v \in v_0 \cdot^I v_1 \cdot^I \dots \cdot^I v_n \wedge$
     $\exists E_0, \dots, E_n.\, E \to^{I*} (u; E_0, \dots, E_n) \wedge \forall j.\, v_j \in [\![E_j]\!]^I$.
3. *for any $u \in \Sigma^*$,*
   $u \in [\![E]\!]^I \iff (u = \varepsilon \wedge E{\downarrow}) \vee (u \neq \varepsilon \wedge \exists E_0, E_1.\, E \to^{I*} (u; E_0, E_1) \wedge E_0{\downarrow} \wedge E_1{\downarrow})$.

▶ **Corollary 42.** *For any E such that $[\![E]\!]$ has uniform rank at most N,*

1. *for any $u, v \in \Sigma^*$, the following are equivalent:*
   - **a.** $uv \in [\![E]\!]^I$;
   - **b.** $\exists z \in [\![E]\!].\ \forall u', u''.\, u = u'u'' \Longrightarrow u' \sim\lhd_N z \rhd\sim u''v$;
   - **c.** $\exists n \leq N, v_0 \in \Sigma^*, v_1, \dots, v_{n-1} \in \Sigma^+, v_n \in \Sigma^*.\, v \sim^I v_0 v_1 \dots v_n \wedge$
     $\exists E_0, \dots, E_n.\, E \to^{I*}_N (u; E_0, \dots, E_n) \wedge \forall j.\, v_j \in [\![E_j]\!]$;
   - **d.** $\exists n \leq N, v_0 \in \Sigma^*, v_1, \dots, v_{n-1} \in \Sigma^+, v_n \in \Sigma^*.\, v \in v_0 \cdot^I v_1 \cdot^I \dots \cdot^I v_n \wedge$
     $\exists E_0, \dots, E_n.\, E \to^{I*}_N (u; E_0, \dots, E_n) \wedge \forall j.\, v_j \in [\![E_j]\!]^I$.
2. *for any $u \in \Sigma^*$,*
   $u \in [\![E]\!]^I \iff (u = \varepsilon \wedge E{\downarrow}) \vee (u \neq \varepsilon \wedge \exists E_0, E_1.\, E \to^{I*}_N (u; E_0, E_1) \wedge E_0{\downarrow} \wedge E_1{\downarrow})$.

▶ **Example 43.** We go back to Example 22. Recall that $E =_{\mathrm{df}} aa + ab + b$ and $E_b =_{\mathrm{df}} R^I_b E = aa + a0 + 0$. Here is one of the refined reordering parts-of-derivatives of E^* along bb.

$$\dfrac{\dfrac{\dfrac{}{E^* \to_2^{I*} (\varepsilon; E^*)} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b \to^I (b;1,1)}}{ab \to^I (b;a1,1)}}{ab+b \to^I (b;a1,1)}}{aa+ab+b \to^I (b;a1,1)}}{E^* \to^I (b;E_b^*(a1),1E^*)} \quad 0<2}{E^* \Rightarrow_2^I (b;E_b^*(a1),1E^*)}}{E^* \to_2^{I*} (b;E_b^*(a1),1E^*)} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b \to^I (b;1,1)}}{ab \to^I (b;a1,1)}}{ab+b \to^I (b;a1,1)}}{aa+ab+b \to^I (b;a1,1)}}{E^* \to^I (b;E_b^*(a1),1E^*)}}{1E^* \to^I (b;1(E_b^*(a1)),1E^*)} \quad 1<2}{E_b^*(a1),1E^* \Rightarrow_2^I (b;E_b^*(a1),1(E_b^*(a1)),1E^*)}}{E^* \to_2^{I*} (bb;E_b^*(a1),1(E_b^*(a1)),1E^*)}$$

In this example, we chose $N =_{\mathrm{df}} 2$. The regexp $1(E_b^*(a1)) \doteq (aa)^*a$ is not nullable, so we could not have dropped it. From here we cannot continue by deriving along a third b by again taking it from the summand ab of E in $1E^*$, as this would produce another nondroppable $1(E_b^*(a1))$ and make the list too long (longer than 3). For example, we are not allowed to establish $w =_{\mathrm{df}} bbbaaa \in [\![E^*]\!]^I$ (by deriving E^* along w and checking if we can arrive at E_0, E_1 with both E_0, E_1 nullable), mandated by $z =_{\mathrm{df}} ababab \in [\![E^*]\!]$, but we are allowed to do so because of $z' =_{\mathrm{df}} bbabaa \in [\![E^*]\!]$. The word z is not useful since among the splits of w as $w = uv$ there is $u =_{\mathrm{df}} bbb$, $v =_{\mathrm{df}} aaa$, which splits z as $u \sim\!\lhd\; z \;\rhd\!\sim v$ scattering u into 3 blocks as $z = a\underline{b}a\underline{b}a\underline{b}$ (we underline the letters from u); the full sequence of these corresponding splits of z is $ababab$, $a\underline{b}abab$, $a\underline{b}a\underline{b}ab$, $a\underline{b}a\underline{b}a\underline{b}$, $\underline{ab}a\underline{b}a\underline{b}$, $\underline{abab}a\underline{b}$, $\underline{ababab}$. The word z', on the contrary, is fine because, for every split of w as $w = uv$, there are at most two blocks of letters from u in z': $bbabaa$, $\underline{b}babaa$, $\underline{bb}abaa$, $\underline{bb}a\underline{b}aa$, $\underline{bbab}aa$, $\underline{bbaba}a$, $\underline{bbabaa}$. The choice $N = 2$ suffices for accepting all of $[\![E^*]\!]^I$, since $[\![E^*]\!]$ happens to have uniform rank 2.

The refined Antimirov reordering parts-of-derivatives of a regexp E give a nondeterministic automaton by $Q^E =_{\mathrm{df}} \{\Gamma \mid \exists u \in \Sigma^*.\, E \to^{I*} (u;\Gamma)\}$, $I^E =_{\mathrm{df}} \{E\}$, $F^E =_{\mathrm{df}} \{E \mid E\!\downarrow\} \cup \{E_0, E_1 \in Q^E \mid E_0\!\downarrow \wedge E_1\!\downarrow\}$, $\Gamma \to^E (a;\Gamma') =_{\mathrm{df}} \Gamma \Rightarrow^I (a;\Gamma')$ for $\Gamma, \Gamma' \in Q^E$. By Prop. 41, this automaton accepts $[\![E]\!]^I$. It is generally not finite as Q^E can contain states Γ of any length.

Given $N \in \mathbb{N}$, another automaton is obtained by restricting Q^E, F^E and $\to^E$ to $Q^E_N =_{\mathrm{df}} \{\Gamma \mid \exists u \in \Sigma^*.\, E \to_N^{I*} (u;\Gamma)\}$, $F^E_N =_{\mathrm{df}} \{E \mid E\!\downarrow\} \cup \{E_0, E_1 \in Q^E_N \mid E_0\!\downarrow \wedge E_1\!\downarrow\}$, $\Gamma \to^E_N (a;\Gamma') =_{\mathrm{df}} \Gamma \Rightarrow^I_N (a;\Gamma')$ for $\Gamma, \Gamma' \in Q^E_N$. By Cor. 42, if $[\![E]\!]$ has uniform rank at most N, then this smaller automaton accepts $[\![E]\!]^I$ despite the truncation. If $[\![E]\!]$ does not have uniform rank or we choose N smaller than the uniform rank, then the N-truncated automaton recognizes a proper subset of $[\![E]\!]^I$. Prop. 37 gives an example of this: however we choose N, the N-truncated automaton fails to accept the word $a^n b^n c a^n b^n$ for $n > N$. This happens because $[\![E]\!]$ does not have uniform rank (and that it has rank 2 does not help).

6.2 Automaton Finiteness for Regular Expressions with Uniform Rank

Is the N-truncated Antimirov automaton finite? The states Γ of Q^E_N are all of length at most $N+1$, so there is hope. The automaton will be finite if we can find a finite set containing all the individual regexps E' appearing in the states Γ. We now define such a set $E^{\to *}$.

▶ **Definition 44.** *We define functions* $(\_)^{\rightsquigarrow+}, \mathbf{R}, (\_)^{\to+}, (\_)^{\to*} : \mathsf{RE} \to \mathcal{P}\mathsf{RE}$ *by*

$$
\begin{array}{rcl@{\qquad}rcl}
a^{\rightsquigarrow+} & =_{\mathrm{df}} & \{1\} & (E+F)^{\rightsquigarrow+} & =_{\mathrm{df}} & E^{\rightsquigarrow+} \cup F^{\rightsquigarrow+} \\
0^{\rightsquigarrow+} & =_{\mathrm{df}} & \emptyset & 1^{\rightsquigarrow+} & =_{\mathrm{df}} & \emptyset
\end{array}
$$

$$
\begin{array}{rcl}
(EF)^{\rightsquigarrow+} & =_{\mathrm{df}} & E^{\rightsquigarrow+} \cup F^{\rightsquigarrow+} \cup E^{\rightsquigarrow+} \cdot \{F\} \cup \{E\} \cdot F^{\rightsquigarrow+} \cup E^{\rightsquigarrow+} \cdot F^{\rightsquigarrow+} \\
(E^*)^{\rightsquigarrow+} & =_{\mathrm{df}} & E^{\rightsquigarrow+} \cup \{E^*\} \cdot E^{\rightsquigarrow+} \cup E^{\rightsquigarrow+} \cdot \{E^*\} \cup E^{\rightsquigarrow+} \cdot (\{E^*\} \cdot E^{\rightsquigarrow+}) \cup \ (E^{\rightsquigarrow+} \cdot \{E^*\}) \cdot E^{\rightsquigarrow+}
\end{array}
$$

$$
\begin{array}{rcl}
\mathbf{R}E & =_{\mathrm{df}} & \{R^I_X E \mid X \subseteq \Sigma\} \\
E^{\to+} & =_{\mathrm{df}} & \mathbf{R}(E^{\rightsquigarrow+}) \\
E^{\to*} & =_{\mathrm{df}} & \{E\} \cup E^{\to+}
\end{array}
$$

▶ **Proposition 45.**
1. *For any E, the set $E^{\to*}$ is finite.*
2. *For any E and X, we have $(R^I_X E)^{\to*} \subseteq R^I_X(E^{\to*})$.*
3. *For any E, a and E_l, E_r, if $E \to^I (a; E_l, E_r)$, then $E_l \in R^I_a(E^{\rightsquigarrow+})$ and $E_r \in E^{\rightsquigarrow+}$.*
4. *For any E, E', X, a, E'_l, E'_r, if $E' \in R^I_X(E^{\rightsquigarrow+})$ and $E' \to^I (a; E'_l, E'_r)$, then $E'_l \in R^I_{Xa}(E^{\rightsquigarrow+})$ and $E'_r \in R^I_X(E^{\rightsquigarrow+})$.*
5. *For any E, u and $E_0, \ldots, E_n$, if $E \to^{I*} (u; E_0, \ldots, E_n)$, then $\forall j.\, E_j \in E^{\to*}$.*

▶ **Proposition 46.** *For every E and N, the state set $\{\Gamma \mid \exists u \in \Sigma^*.\, E \to^{I*}_N (u; \Gamma)\}$ of the N-truncated refined Antimirov automaton for E (accepting $[\![E]\!]^I$ if $[\![E]\!]$ has uniform rank at most N) is finite.*

7 Related Work

Syntactic derivative constructions for regular expressions extended with constructors for (versions of) the shuffle operation have been considered, for example, by Sulzmann and Thiemann [20] for the Brzozowski derivative and by Broda et al. [4] for the Antimirov derivative. This is relevant to our derivatives since $L \cdot^I L'$ is by definition a language between $L \cdot L'$ and $L \sqcup\!\sqcup L'$. Thus our Brzozowski and Antimirov reordering derivatives of EF must be between the classical Brzozowski and Antimirov derivatives of EF and $E \sqcup\!\sqcup F$.

8 Conclusion and Future Work

We have shown that the Brzozowski and Antimirov derivative operations generalize to trace closures of regular languages in the form of reordering derivative operations. The sets of Brzozowski resp. Antimirov reordering (parts-of-)derivatives of a regexp are generally infinite, so the deterministic and nondeterministic automata that they give, accepting the trace closure, are generally infinite. Still, if the regexp is star-connected, their appropriate quotients are finite. Also, the set of N-bounded refined Antimirov reordering parts-of-derivatives is finite without quotienting, and we showed that, if the language of the regexp has uniform rank at most N, the N-truncated refined Antimirov automaton accepts the trace closure. We also proved that star-connected expressions define languages with finite uniform rank.

Our intended application for this is operational semantics in the context of relaxed memory (where, e.g., shadow writes, i.e., writes from local buffers to shared memory, can be reorderable with other actions). For sequential composition EF it is usually required that, to execute any action from F, execution of E must have completed. In the jargon of derivatives, this is to say that for an action from F to become executable, what is left of E has to have become nullable (i.e., one can consider the execution of E completed). With reordering derivatives, we can execute an action from F successfully even when what is left of E is not nullable. It suffices that some sequence of actions to complete the residual of E is reorderable with the selected action of F.

In the definitions of the derivative operations we only use I in one direction, i.e., we do not make use of its symmetry. It would be interesting to see if our results can be generalized to the setting of semi-commutations [6] and which changes are required for that.

References

1 IJsbrand Jan Aalbersberg and Emo Welzl. Trace Languages Defined by Regular String Languages. *Theor. Inf. Appl.*, 20(2):103–119, 1986. doi:10.1051/ita/1986200201031.

2 Valentin M. Antimirov. Partial Derivatives of Regular Expressions and Finite Automaton Constructions. *Theor. Comput. Sci.*, 155(2):291–319, 1996. doi:10.1016/0304-3975(95)00182-4.

3 Alberto Bertoni, Giancarlo Mauri, and Nicoletta Sabadini. Unambiguous Regular Trace Languages. In Janos Demetrovics, Gyula Katona, and Arto Salomaa, editors, *Algebra, Combinatorics, and Logic in Computer Science*, volume 42 of *Colloquia Mathematica Societas János Bolyai*, pages 113–123. North-Holland, 1986.

4 Sabine Broda, António Machiavelo, Nelma Moreira, and Rogério Reis. Partial Derivative Automaton for Regular Expressions with Shuffle. In Jeffrey Shallit and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems: 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25-27, 2015, Proceedings*, volume 9118 of *Lecture Notes in Computer Science*, pages 21–32. Springer, 2015. doi:10.1007/978-3-319-19225-3_2.

5 Janusz A. Brzozowski. Derivatives of Regular Expressions. *J. ACM*, 11(4):481–494, 1964. doi:10.1145/321239.321249.

6 Mireille Clerbout and Michel Latteux. Semi-commutations. *Inf. Comput.*, 73(1):59–74, 1987. doi:10.1016/0890-5401(87)90040-X.

7 Kosaburo Hashiguchi. Recognizable Closures and Submonoids of Free Partially Commutative Monoids. *Theor. Comput. Sci.*, 86(2):233–241, 1991. doi:10.1016/0304-3975(91)90019-X.

8 Tony Hoare, Bernhard Möller, Georg Struth, and Ian Wehrman. Concurrent Kleene Algebra and Its Foundations. *J. Log. Algebr. Program.*, 80(6):266–296, 2011. doi:10.1016/j.jlap.2011.04.005.

9 Stephen C. Kleene. Representation of Events in Nerve Sets and Finite Automata. In Claude E. Shannon and John McCarthy, editors, *Automata Studies*, volume 34 of *Annals of Mathematics Studies*, pages 3–42. Princeton University Press, 1956.

10 Barbara Klunder, Edward Ochmański, and Krystyna Stawikowska. On Star-Connected Flat Languages. *Fund. Inf.*, 67(1–3):93–105, 2005. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi67-1-3-08.

11 Dexter Kozen. A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events. *Inf. Comput.*, 110(2):366–390, 1994. doi:10.1006/inco.1994.1037.

12 Hendrik Maarand and Tarmo Uustalu. Reordering Derivatives of Trace Closures of Regular Languages. arXiv preprint 1908.03551, 2019. arXiv:1908.03551.

13 Antoni Mazurkiewicz. Concurrent Program Schemes and Their Interpretations. DAIMI Rep. PB-78, University of Aarhus, 1978.

14 Antoni Mazurkiewicz. Introduction to Trace Theory. In Volker Diekert, editor, *The Book of Traces*, pages 3–41. World Scientific, 1995. doi:10.1142/9789814261456_0001.

15 Edward Ochmański. Regular Behaviour of Concurrent Systems. *Bull. EATCS*, 27:56–67, 1985.

16 Edward Ochmański. Recognizable Trace Languages. In Volker Diekert, editor, *The Book of Traces*, pages 167–204. World Scientific, 1995. doi:10.1142/9789814261456_0006.

17 Michael O. Rabin and Dana S. Scott. Finite Automata and Their Decision Problems. *IBM J. Res. Devel.*, 3(2):114–125, 1959. doi:10.1147/rd.32.0114.

18 Jacques Sakarovitch. On Regular Trace Languages. *Theor. Comput. Sci.*, 52:59–75, 1987. doi:10.1016/0304-3975(87)90080-6.

19 Jacques Sakarovitch. The "Last" Decision Problem for Rational Trace Languages. In Imre Simon, editor, *LATIN '92, 1st Latin American Symposium on Theoretical Informatics, São Paulo, Brazil, April 6-10, 1992, Proceedings*, volume 583 of *Lecture Notes in Computer Science*, pages 460–473. Springer, 1992. doi:10.1007/BFb0023848.

20 Martin Sulzmann and Peter Thiemann. Derivatives for Regular Shuffle Expressions. In Adrian-Horia Dediu, Enrico Formenti, Carlos Martín-Vide, and Bianca Truthe, editors, *Language and Automata Theory and Applications: 9th International Conference, LATA 2015, Nice, France, March 2-6, 2015, Proceedings*, volume 8977 of *Lecture Notes in Computer Science*, pages 275–286. Springer, 2015. doi:10.1007/978-3-319-15579-1_21.

21 Wiesław Zielonka. Notes on Finite Asynchronous Automata. *Theor. Inf. Appl.*, 21(2):99–135, 1987. doi:10.1051/ita/1987210200991.

22 Wiesław Zielonka. Asynchronous Automata. In Volker Diekert, editor, *The Book of Traces*, pages 205–247. World Scientific, 1995. doi:10.1142/9789814261456_0007.