On the Average Complexity of Partial Derivative Transducers

Stavros Konstantinidis [b], António Machiavelo [a], Nelma Moreira[a,*], Rogério Reis [a]

[a] CMUP & DM-DCC, Faculdade de Ciências da Universidade do Porto, Rua do Campo Alegre, 4169-007, Porto, Portugal

[b] Saint Mary's University, 923, Robie Str, Halifax, Nova Scotia, Canada

Abstract

2D regular expressions represent rational relations over two alphabets Σ and Δ. In standard 2D expressions (S2D-RE) the basic terms are generators of $\Sigma^\star \times \Delta^\star$, while in generalised 2D expressions (2D-RE) the basic terms are pairs of (ordinary) regular expressions over one alphabet (1D). In this paper we study the average state complexity of partial derivative standard transducers ($\mathcal{T}_{\text{PD}}$) for both S2D-RE and 2D-RE. For S2D-RE we obtain the same asymptotic bounds as for partial derivative automata. For 2D-RE, while in the worst case the number of states of $\mathcal{T}_{\text{PD}}$ can be $O(n^2)$, where n is the size of the expression, asymptotically and on average that value is bounded from above by $O(n^{\frac{3}{2}})$. We also show that asymptotically and on average the alphabetic size of a 2D-RE is half of its size. All results are obtained in the framework of analytic combinatorics considering generating functions of parametrised combinatorial classes defined implicitly by algebraic curves. In particular, we generalise the methods developed in previous work to a broad class of analytic functions.

Keywords: Transducers, Rational Expressions, Partial Derivatives, Analytic Combinatorics, Average Case Complexity

[⋆] This is a completely revised and expanded version of a paper presented at the 46th International Conference on Current Trends in Theory and Practice of Informatics, SOFSEM 2020 [1].

[⋆⋆] This work was partially supported by NSERC, Canada and CMUP, which is financed by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the project with reference UIDB/00144/2020.

[*] Corresponding author

Email addresses: `s.konstantinidis@smu.ca` (Stavros Konstantinidis), `antonio.machiavelo@fc.up.pt` (António Machiavelo), `nelma.moreira@fc.up.pt` (Nelma Moreira), `rogerio.reis@fc.up.pt` (Rogério Reis)

1. Introduction

We consider 2D expressions that represent rational (word) relations over two alphabets. Expressions and transducers with labels over finitely generated monoids were studied by Konstantinidis et al. [2, 3], and also by Demaille [4]. Partial derivative methods have become a standard method to manipulate several kinds of expressions [5, 6, 4, 7, 8, 9], not only because they are in general more succinct than other equivalent constructions, but for some operators they are easier to define (e.g. for intersection [7]). For regular languages, the average complexity of partial derivative automata ($\mathcal{A}_{\mathrm{PD}}$), considering different kinds of expressions, has been studied [10, 7, 9, 11]. Using the framework of analytic combinatorics, for ordinary (1D) regular expressions of (tree-)size n (with concatenation, union and Kleene star) it was shown that, asymptotically and on average, the number of states of $\mathcal{A}_{\mathrm{PD}}$ is $\frac{1}{4}n$, (with the worst-case being $O(n^2)$) while for expressions with intersection of (tree-)size n that number is upper bounded by $(1.056 + o(1))^n$ (with the worst-case being $O(2^n)$) [10, 12, 7].

In this paper we consider standard 2D expressions (S2D-RE) where basic terms are generators of the product of two free monoids, and generalised 2D expressions (2D-RE) where basic terms are pairs of ordinary 1D regular expressions over an alphabet. For these two kinds of expressions we define a partial derivative standard transducer ($\mathcal{T}_{\mathrm{PD}}$), and study its average state complexity. While for S2D-REs the analytic combinatorial methods used for ordinary 1D regular expressions could still be applied, for the 2D-REs this was not the case. In particular, to get explicit expressions for the generating functions involved would be unmanageable. So, generating functions implicitly defined by algebraic curves must be used, and in previous work it was shown how to get the required information for the asymptotic estimates with an indirect use of the existence of Puiseux expansions at singularities [8]. In this paper, as the involved algebraic curves are more intricate, we needed to refine the methods described in the literature, and use Puiseux expansions together with the Newton's polygon technique to find the estimates for the asymptotic behaviours of parametrised families of combinatorial classes.

The paper structure and the main results are the following. Section 2 reviews the partial derivative construction for ordinary 1D regular expressions. In Section 3 we define the two kinds of 2D expressions, and present the corresponding constructions of partial derivative transducers ($\mathcal{T}_{\mathrm{PD}}$). While for standard 2D expressions partial derivatives have been considered before [13, 3], for generalised 2D expressions our presentation is novel. In Section 4 we consider the analytic combinatorics framework and the refined method developed to deal with the parametrised families of combinatorial classes. This method is an enhancement of the one represented before [8], and a more detailed exposition is presented in [11]. Section 5 presents the average complexity results obtained using the framework of Section 4. For generalised 2D expressions, 2D-RE, while in the worst case the number of states of $\mathcal{T}_{\mathrm{PD}}$ can be $O(n^2)$, where n is the size of the expression, asymptotically and on average, that value is bounded from above by $O(n^{\frac{3}{2}})$ (Section 5.1). Restricting to pairs of 1D expressions, the previous bound is already reached, showing that these expressions are responsible for the increasing of complexity (Section 5.2). For ordinary 1D regular expressions, the number of alphabetic symbols in an expression is, asymptotically and on average, one half of the size of the expression [14, 10]. Finally, in Section 5.3, a similar result is obtained for generalised 2D expressions, 2D-RE. In Section 5.4, we show that for S2D-RE, asymptotically and on average, the number of states of $\mathcal{T}_{\mathrm{PD}}$ is $\frac{1}{4}$ of the size of the expression. Some experimental results are discussed in Section 6 and Section 7 concludes.

The present paper contains the proofs of propositions and lemmata of Section 3.1 that were missing from the conference paper [1]. Several illustrative examples are included. We added the study of standard 2D expressions and, in particular, the average state complexity of $\mathcal{T}_{\mathrm{PD}}$ (sections 3.2 and 5.4). The average complexity of the alphabetic size of generalised 2D expressions is also new, as well as, the experimental results.

2. Preliminares

A *nondeterministic finite automaton* (NFA) is a five-tuple $A = \langle Q, \Sigma, \delta, I, F\rangle$ where Q is a finite set of states, Σ is a finite alphabet, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, and $\delta : Q \times \Sigma \to 2^Q$ is the transition function. The size of an NFA is its number of states. The transition function can be extended to words and to sets of states in the natural way. When $I = \{q_0\}$, we use $I = q_0$. The *language accepted* by A is $\mathcal{L}(A) = \{w \in \Sigma^\star \mid \delta(I, w) \cap F \neq \emptyset\}$. Given an alphabet Σ, the set RE of (1D) *regular expressions*, $\boldsymbol{r}$, over Σ consists of $\emptyset$ and the expressions defined by the following grammar:

$$\boldsymbol{r} \quad := \quad \varepsilon \mid \sigma \in \Sigma \mid (\boldsymbol{r} + \boldsymbol{r}) \mid (\boldsymbol{r} \cdot \boldsymbol{r}) \mid (\boldsymbol{r}^\star), \tag{1}$$

where the operator $\cdot$ (concatenation) and the outermost parentheses are often omitted. The *language* associated to $\boldsymbol{r}$ is denoted by $\mathcal{L}(\boldsymbol{r})$ and defined as usual (with ε representing the empty word). If $S \subseteq \mathrm{RE}$, $\mathcal{L}(S) = \cup_{\boldsymbol{r}\in S}\mathcal{L}(\boldsymbol{r})$. Two expressions $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are *equivalent*, $\boldsymbol{r}_1 \sim \boldsymbol{r}_2$ if $\mathcal{L}(\boldsymbol{r}_1) = \mathcal{L}(\boldsymbol{r}_2)$, and this notion extends to sets of expressions. The *(tree-)size* $|\boldsymbol{r}|$ of $\boldsymbol{r} \in \mathrm{RE}$ is the number of symbols in $\boldsymbol{r}$ (disregarding parentheses). The *alphabetic size* $|\boldsymbol{r}|_\Sigma$ is the number of letters occurring in $\boldsymbol{r}$. We define the *constant part* of $\boldsymbol{r}$, $\mathsf{c}(\boldsymbol{r})$, by $\mathsf{c}(\boldsymbol{r}) = \varepsilon$ if $\varepsilon \in \mathcal{L}(\boldsymbol{r})$, and $\mathsf{c}(\boldsymbol{r}) = \emptyset$ otherwise. This function is extended to sets of expressions by $\mathsf{c}(S) = \varepsilon$ if and only if exists $\boldsymbol{r} \in S$ such that $\mathsf{c}(\boldsymbol{r}) = \varepsilon$. In the case of a singleton $\{s\}$ we write it simply as s. Given $L \subseteq \Sigma^\star$ and $\sigma \in \Sigma$, the quotient of L by σ is $\sigma^{-1}L = \{\, w \mid \sigma w \in L\}$. This notion can be extended to words and languages. The partial derivative automaton of a regular expression was introduced independently by Mirkin [15] and Antimirov [5]. For a regular expression $\boldsymbol{r} \in \mathrm{RE}$, let the *linear form* of $\boldsymbol{r}$, $\mathsf{n} : \mathrm{RE} \to 2^{\Sigma\times\mathrm{RE}}$, be inductively defined by

$$\begin{aligned} \mathsf{n}(\emptyset) &= \mathsf{n}(\varepsilon) = \emptyset, & \mathsf{n}(\boldsymbol{r}+\boldsymbol{r}') &= \mathsf{n}(\boldsymbol{r}) \cup \mathsf{n}(\boldsymbol{r}'),\\ \mathsf{n}(\sigma) &= \{(\sigma, \varepsilon)\}, & \mathsf{n}(\boldsymbol{r}\boldsymbol{r}') &= \mathsf{n}(\boldsymbol{r})\boldsymbol{r}' \cup \mathsf{c}(\boldsymbol{r})\,\mathsf{n}(\boldsymbol{r}'),\\ && \mathsf{n}(\boldsymbol{r}^\star) &= \mathsf{n}(\boldsymbol{r})\boldsymbol{r}^\star, \end{aligned} \tag{2}$$

where for any $S \subseteq \Sigma \times \mathrm{RE}$, we define $S\emptyset = \emptyset S = \emptyset$, $S\varepsilon = \varepsilon S = S$, and $S\boldsymbol{r}' = \{\, (\sigma, \boldsymbol{r}\boldsymbol{r}') \mid (\sigma, \boldsymbol{r}) \in S \wedge \boldsymbol{r} \neq \varepsilon \,\} \cup \{\, (\sigma, \boldsymbol{r}') \mid (\sigma, \varepsilon) \in S \,\}$ if $\boldsymbol{r}' \neq \emptyset, \varepsilon$ (and analogously for $\boldsymbol{r}'S$).

Proposition 1 ([5])**.** *For all $\boldsymbol{r} \in \mathrm{RE}$, $\boldsymbol{r} \sim \bigcup_{(\sigma,\boldsymbol{r}')\in \mathsf{n}(\boldsymbol{r})} \sigma\boldsymbol{r}' \cup \mathsf{c}(\boldsymbol{r})$.*

For a regular expression $\boldsymbol{r} \in \mathrm{RE}$ and a symbol $\sigma \in \Sigma$, *the set of partial derivatives of $\boldsymbol{r}$ w.r.t. σ* is defined by $\partial_\sigma(\boldsymbol{r}) = \{\, \boldsymbol{r}' \mid (\sigma, \boldsymbol{r}') \in \mathsf{n}(\boldsymbol{r}) \,\}$. We have $\mathcal{L}(\partial_\sigma(\boldsymbol{r})) = \sigma^{-1}\mathcal{L}(\boldsymbol{r})$. Partial derivatives can be extended w.r.t words. The set of partial derivatives of an expression $\boldsymbol{r}$ can be defined by iterating the linear form. Let $\pi_0(\boldsymbol{r}) = \downarrow_2(\mathsf{n}(\boldsymbol{r}))$, where $\downarrow_2(s,t) = t$ is the standard second projection on pairs of objects and naturally extended to sets of pairs. Iteratively applying the operator π_0 we have,

$$\pi_i(\boldsymbol{r}) = \pi_0(\pi_{i-1}(\boldsymbol{r})),$$

for $i \in \mathbb{N}$, and

$$\pi(\boldsymbol{r}) = \bigcup_{i\in\mathbb{N}_0} \pi_i(\boldsymbol{r}).$$

The set $\mathsf{PD}(\boldsymbol{r}) = \pi(\boldsymbol{r}) \cup \{\boldsymbol{r}\}$ is the *set of partial derivatives* of $\boldsymbol{r}$ and $\pi(\boldsymbol{r})$ is the *support*[1].

Proposition 2 ([15])**.** *The support $\pi(\boldsymbol{r})$ is inductively defined by*

$$\begin{array}{rcl@{\qquad}rcl}
\pi(\emptyset) &=& \emptyset, & \pi(\boldsymbol{r}+\boldsymbol{r}') &=& \pi(\boldsymbol{r}) \cup \pi(\boldsymbol{r}'),\\
\pi(\varepsilon) &=& \emptyset, & \pi(\boldsymbol{r}\boldsymbol{r}') &=& \pi(\boldsymbol{r})\boldsymbol{r}' \cup \pi(\boldsymbol{r}'),\\
\pi(\sigma) &=& \{\varepsilon\}, & \pi(\boldsymbol{r}^\star) &=& \pi(\boldsymbol{r})\boldsymbol{r}^\star,
\end{array}$$

where, for any $S \subseteq \mathrm{RE}$, we define $S\emptyset = \emptyset S = \emptyset$, $S\varepsilon = \varepsilon S = S$, and $S\boldsymbol{r}' = \{\, \boldsymbol{r}\boldsymbol{r}' \mid \boldsymbol{r} \in S \wedge \boldsymbol{r} \neq \varepsilon \,\} \cup \{\, \boldsymbol{r}' \mid \varepsilon \in S \,\}$ if $\boldsymbol{r}' \neq \emptyset, \varepsilon$ (and analogously for $\boldsymbol{r}'S$).

Proposition 3 ([5, 15])**.** $|\pi(\boldsymbol{r})| \leq |\boldsymbol{r}|_\Sigma$ *and* $|\mathsf{PD}(\boldsymbol{r})| \leq |\boldsymbol{r}|_\Sigma + 1$.

The *partial derivative automaton* of $\boldsymbol{r}$ is defined to be the NFA $\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}) = \langle \mathsf{PD}(\boldsymbol{r}), \Sigma, \delta_{\mathrm{PD}}, \boldsymbol{r}, F \rangle$, where $F = \{\, \boldsymbol{r}_1 \in \mathsf{PD}(\boldsymbol{r}) \mid \mathsf{c}(\boldsymbol{r}_1) = \varepsilon \,\}$, and $\delta_{\mathrm{PD}} = \{\, (\boldsymbol{r}_1, \sigma, \boldsymbol{r}') \mid \boldsymbol{r}_1 \in \mathsf{PD}(\boldsymbol{r}) \wedge (\sigma, \boldsymbol{r}') \in \mathsf{n}(\boldsymbol{r}_1) \,\}$.

Proposition 4 ([5, 15])**.** *For all $\boldsymbol{r} \in \mathrm{RE}$, $\mathcal{L}(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r})) = \mathcal{L}(\boldsymbol{r})$.*

3. 2D Expressions

Let Σ and Δ be two alphabets. A *relation* R is any subset of $\Sigma^\star \times \Delta^\star$. The *concatenation* of two relations R and S is the relation $RS = \{(u_1u_2, v_1v_2) \mid (u_1, v_1) \in R \wedge (u_2, v_2) \in S\}$. The *Kleene closure* of the relation R is the relation

[1] Extending partial derivatives w.r.t. words, one could also define $\mathsf{PD}(\boldsymbol{r}) = \bigcup_{w\in\Sigma^\star} \partial_w(\boldsymbol{r})$.

$R^\star = \bigcup_{n\geq 0} R^n$. The monoid $\Sigma^\star \times \Delta^\star$ has the identity $(\varepsilon, \varepsilon)$, and the following set of generators $\{(\sigma,\varepsilon),(\varepsilon,\tau) \mid \sigma \in \Sigma \wedge \tau \in \Delta\}$ with the set of equations

$$\{\, (\sigma,\varepsilon)(\varepsilon,\tau) \doteq (\sigma,\tau), (\varepsilon,\tau)(\sigma,\varepsilon) \doteq (\sigma,\tau) \mid \sigma \in \Sigma \wedge \tau \in \Delta \,\}. \tag{3}$$

For a relation $R \subseteq \Sigma^\star \times \Delta^\star$, the quotient of R by a symbol is defined as before, but one needs to take into account the above equations. For $\sigma \in \Sigma$ and $\tau \in \Delta$, one has

$$\begin{aligned} (\sigma,\varepsilon)^{-1}R &= \{\, (u,v) \mid (\sigma u, v) \in R \,\}, \\ (\varepsilon,\tau)^{-1}R &= \{\, (u,v) \mid (u, \tau v) \in R \,\}. \end{aligned}$$

The set of *rational relations* is the smallest set of relations that contains the finite relations and is closed under union, concatenation and Kleene closure. Rational relations are accepted by transducers. A *finite transducer in standard-form* (SFT) over two alphabets Σ and Δ is defined as an NFA, except that the transition function is $\delta : Q \times (\Sigma_\varepsilon \times \Delta_\varepsilon) \to 2^Q$, where for a set X, $X_\varepsilon = X \cup \{\varepsilon\}$. The relation realised by an SFT t is denoted by $\mathcal{R}(t)$. More details about rational relations and transducers can be found in [16, 17].

In the following subsections we consider two types of expressions that represent rational relations: standard 2D expressions and generalised 2D expressions. The notions of linear form, of partial derivative and of partial derivative transducers are extend to 2D expressions. However, some differences occur related with the equations (3). For both types of 2D expressions we define partial derivative transducers. In Section 5 we study the average state complexity of these transducers.

Expressions representing n-ary relations definable by multitape automata with n tapes [18], were first considered by Rosenberg [19], Makarevskii and Stotskaya [20] and Mirkin [13]. For $n = 2$ those expressions coincide with the standard 2D expressions. Using techniques similar to those in [15], Mirkin introduced the notions of partial derivatives and of partial derivative (multitape) automata to prove the equivalence between expressions and (multitape) automata[2]. The definition of $\mathcal{T}_{\mathrm{PD}}$ in Section 5.4 differs from that work in notation and the use of linear forms. Kaplan and Kay [22] use generalised 2D expressions to represent phonological and morphological rewrite rules, but the authors did not present the construction of equivalent transducers. These formalisms were widely used and extended for Natural Language Processing [23, 24, 25]. Recently, Demaille [4] defined derivative automata for multitape weighted regular expressions. The generalised 2D expressions and transducers studied in this paper are restrictions of those models to two tapes and the Boolean semiring, however, our results address different aspects of these objects.

[2] The author presents a Kleene's Theorem for 2-tape automata and 2D expressions but the claim that the proofs would directly extend to n-tape automata turned out to be wrong [21].

3.1. Generalised 2D-RE *Expressions*

A *generalised* 2D *regular expression* (2D-RE) over Σ and Δ, where Σ is the *input alphabet* and Δ the *output alphabet*, is an expression that is either $\emptyset$, or can be defined by the following grammar

$$\boldsymbol{g} \quad := \quad \boldsymbol{r}/\boldsymbol{r}' \mid (\boldsymbol{g}+\boldsymbol{g}) \mid (\boldsymbol{g}\cdot\boldsymbol{g}) \mid (\boldsymbol{g}^\star), \tag{4}$$

where $\boldsymbol{r} \in$ RE over Σ and $\boldsymbol{r}' \in$ RE over Δ. The *relation* $\mathcal{R}(\boldsymbol{g}) \subseteq \Sigma^\star \times \Delta^\star$ realised by a 2D-RE $\boldsymbol{g}$ is defined inductively as follows

$$\begin{aligned} \mathcal{R}(\boldsymbol{r}/\boldsymbol{r}') &= \mathcal{L}(\boldsymbol{r}) \times \mathcal{L}(\boldsymbol{r}'), \\ \mathcal{R}(\boldsymbol{g}\cdot\boldsymbol{g}') &= \mathcal{R}(\boldsymbol{g})\mathcal{R}(\boldsymbol{g}'), \\ \mathcal{R}(\boldsymbol{g}^\star) &= (\mathcal{R}(\boldsymbol{g}))^\star. \end{aligned}$$

Two expressions $\boldsymbol{g}, \boldsymbol{g}'$ are *equivalent*, $\boldsymbol{g} \sim \boldsymbol{g}'$, if $\mathcal{R}(\boldsymbol{g}) = \mathcal{R}(\boldsymbol{g}')$. A relation is *rational* if and only if it is represented by a 2D-RE[3]. The *size* and the *alphabetic size* of a 2D-RE is defined as for RE. The *constant part* of a 2D-RE expression $\boldsymbol{g}$ is given by c : 2D-RE $\longrightarrow \{\emptyset, \varepsilon/\varepsilon\}$ such that $\mathsf{c}(\boldsymbol{g}) = \varepsilon/\varepsilon$ if $(\varepsilon,\varepsilon) \in \mathcal{R}(\boldsymbol{g})$, and $\mathsf{c}(\boldsymbol{g}) = \emptyset$, otherwise. For $S \subseteq$ 2D-RE or $S \subseteq (\Sigma_\varepsilon \times \Delta_\varepsilon) \times$ 2D-RE and an expression $\boldsymbol{g}$, we adopt the same conventions as for 1D expressions regarding $\boldsymbol{g}S$ and $S\boldsymbol{g}$. In particular, we let $(\varepsilon/\varepsilon)S = S(\varepsilon/\varepsilon) = S$.

Now we define the *linear form* function n : 2D-RE $\to 2^{(\Sigma_\varepsilon \times \Delta_\varepsilon) \times \text{2D-RE}}$. For an expression $\boldsymbol{g} \in$ 2D-RE, the linear form $\mathsf{n}(\boldsymbol{g})$ is defined as in Equations (2), except for the case where $\boldsymbol{g}$ is of the form $\boldsymbol{r}_1/\boldsymbol{r}_2$:[4]

$$\begin{aligned} \mathsf{n}(\boldsymbol{r}_1/\boldsymbol{r}_2) \quad = \quad & (\mathsf{n}(\boldsymbol{r}_1) \,||\, \mathsf{n}(\boldsymbol{r}_2)) \;\cup\; \mathsf{c}(\boldsymbol{r}_2)(\mathsf{n}(\boldsymbol{r}_1) \,||\{(\varepsilon,\varepsilon)\}) \\ & \cup\; \mathsf{c}(\boldsymbol{r}_1)(\{(\varepsilon,\varepsilon)\} \,||\, \mathsf{n}(\boldsymbol{r}_2)), \end{aligned} \tag{5}$$

where for $N \subseteq \Sigma_\varepsilon \times$ RE and $M \subseteq \Delta_\varepsilon \times$ RE,

$$N \,||\, M = \{\, ((\gamma,\gamma')\,, \boldsymbol{r}/\boldsymbol{r}') \mid (\gamma,\boldsymbol{r}) \in N \;\wedge\; (\gamma',\boldsymbol{r}') \in M \,\}.$$

Example 1. *Let $\boldsymbol{r}_1 = ab$ and $\boldsymbol{r}_2 = abc$. We have*

$$\begin{aligned} &\mathsf{n}(ab) = \{(a,b)\}, \quad \mathsf{n}(abc) = \{(a,bc)\}, \\ &\mathsf{n}(ab/abc) = \mathsf{n}(ab) \,||\, \mathsf{n}(abc) = \{(a,a),\, b/bc\}, \\ &\mathsf{n}(b/bc) = \mathsf{n}(b) \,||\, \mathsf{n}(bc) = \{(b,b),\, \varepsilon/c\}, \\ &\mathsf{n}(\varepsilon/c) = \mathsf{c}(\varepsilon)\big(\{(\varepsilon,\varepsilon)\} \,||\{(c,\varepsilon)\},\big) = \{(\varepsilon,c),\, \varepsilon/\varepsilon\}. \quad \square \end{aligned}$$

The correctness of the definition of $\mathsf{n}(\boldsymbol{r}_1/\boldsymbol{r}_2)$ is given by the following proposition.

[3]This follows from the definition above.

[4]We note that one possibility was to consider $\mathsf{n}(\boldsymbol{r}_1/\boldsymbol{r}_2) = \{(\boldsymbol{r}_1/\boldsymbol{r}_2, \varepsilon/\varepsilon)\}$ (see [3]), but then one could not construct directly an SFT for $\boldsymbol{r}_1/\boldsymbol{r}_2$.

Proposition 5. *For all* $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathrm{RE}$*,*

$$\boldsymbol{r}_1/\boldsymbol{r}_2 \sim \bigcup_{((\gamma,\gamma'),\boldsymbol{g}')\in\mathsf{n}(\boldsymbol{r}_1/\boldsymbol{r}_2)} (\gamma/\gamma')\boldsymbol{g}' \cup \mathsf{c}(\boldsymbol{r}_1/\boldsymbol{r}_2).$$

Proof. By Proposition 1 each expression $\boldsymbol{r} \in \mathrm{RE}$ satisfies

$$\boldsymbol{r} \sim \bigcup_{(\sigma,\boldsymbol{r}')\in\mathsf{n}(\boldsymbol{r})} \sigma\boldsymbol{r}' \cup \mathsf{c}(\boldsymbol{r}).$$

Thus, one can check that

$$\begin{aligned}\boldsymbol{r}_1/\boldsymbol{r}_2 \quad \sim \quad & \bigcup_{\substack{(\sigma,\boldsymbol{r})\in\mathsf{n}(\boldsymbol{r}_1)\\(\tau,\boldsymbol{r}'')\in\mathsf{n}(\boldsymbol{r}_2)}} (\sigma/\tau)(\boldsymbol{r}/\boldsymbol{r}'') \;\cup\; \mathsf{c}(\boldsymbol{r}_2)\left(\bigcup_{(\sigma,\boldsymbol{r})\in\mathsf{n}(\boldsymbol{r}_1)} (\sigma/\varepsilon)(\boldsymbol{r}/\varepsilon)\right)\\ & \cup\; \mathsf{c}(\boldsymbol{r}_1)\left(\bigcup_{(\tau,\boldsymbol{r}'')\in\mathsf{n}(\boldsymbol{r}_2)} (\varepsilon/\tau)(\varepsilon/\boldsymbol{r}'')\right) \;\cup\; \mathsf{c}(\boldsymbol{r}_1/\boldsymbol{r}_2).\end{aligned}$$

In the first three terms, the unions go exactly through the elements of $\mathsf{n}(\boldsymbol{r}_1/\boldsymbol{r}_2)$ in Equation (5). And $\mathsf{c}(\boldsymbol{r}_1/\boldsymbol{r}_2) = \varepsilon/\varepsilon$ iff $\mathsf{c}(\boldsymbol{r}_1) = \mathsf{c}(\boldsymbol{r}_2) = \varepsilon$, from which the result follows. □

Then, we have

Proposition 6. *For all* $\boldsymbol{g} \in$ 2D-RE*,* $\boldsymbol{g} \sim \bigcup_{((\gamma,\gamma'),\boldsymbol{g}')\in\mathsf{n}(\boldsymbol{g})} (\gamma/\gamma')\boldsymbol{g}' \cup \mathsf{c}(\boldsymbol{g})$*.*

As before, one can obtain the *support* $\pi(\boldsymbol{g})$ of an expression $\boldsymbol{g}$ by iterating the linear form. Only the base cases differ from the ones in Proposition 2. Instead of considering partial derivatives, one can consider the following operators. Again, let $\pi_0(\boldsymbol{r}) = \downarrow_2(\mathsf{n}(\boldsymbol{r}))$, where $\downarrow_2(s,t) = t$ is the standard second projection on pairs of objects. Iteratively applying the operator π_0 we have,

$$\pi_i(\boldsymbol{r}) = \pi_0(\pi_{i-1}(\boldsymbol{r})),$$

for $i \in \mathbb{N}$, and

$$\pi(\boldsymbol{r}) = \bigcup_{i\in\mathbb{N}_0} \pi_i(\boldsymbol{r}).$$

The following lemma is used in the proof of Proposition 8 and is easy to prove. For $S, T \subseteq \mathrm{RE}$, we have the notation $S \,||\, T = \{\, \boldsymbol{r}/\boldsymbol{r}' \mid \boldsymbol{r} \in S \;\wedge\; \boldsymbol{r}' \in T \,\}$.

Lemma 7. *For all* $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathrm{RE}$*,*

$$\begin{aligned}\pi_0(\boldsymbol{r}_1/\boldsymbol{r}_2) &= \pi_0(\boldsymbol{r}_1) \,||\, \pi_0(\boldsymbol{r}_2) \cup \mathsf{c}(\boldsymbol{r}_2)(\pi_0(\boldsymbol{r}_1) \,||\{\varepsilon\}) \;\cup\; \mathsf{c}(\boldsymbol{r}_1)(\{\varepsilon\} \,||\, \pi_0(\boldsymbol{r}_2)),\\ \pi_0(\varepsilon/\boldsymbol{r}_2) &= \{\varepsilon\} \,||\, \pi_0(\boldsymbol{r}_2),\\ \pi_0(\boldsymbol{r}_1/\varepsilon) &= \pi_0(\boldsymbol{r}_1) \,||\{\varepsilon\}.\end{aligned}$$

From this we obtain the following

Proposition 8. *For all $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathrm{RE}$,*

$$\pi(\boldsymbol{r}_1/\boldsymbol{r}_2) \quad \subseteq \quad \pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2) \;\cup\; \pi(\boldsymbol{r}_1) \,||\{\varepsilon\} \;\cup\; \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2).$$

Proof. We prove by induction on $i \geq 0$ that

$$\pi_i(\boldsymbol{r}_1/\boldsymbol{r}_2) \quad \subseteq \quad \pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2) \;\cup\; \pi(\boldsymbol{r}_1) \,||\{\varepsilon\} \;\cup\; \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2).$$

For $i = 0$, and from Equation (5), we have

$$\begin{aligned} \pi_0(\boldsymbol{r}_1/\boldsymbol{r}_2) \quad &= \quad \pi_0(\boldsymbol{r}_1) \,||\, \pi_0(\boldsymbol{r}_2) \;\cup\; \mathsf{c}(\boldsymbol{r}_2)(\pi_0(\boldsymbol{r}_1) \,||\{\varepsilon\}) \;\cup\; \mathsf{c}(\boldsymbol{r}_1)(\{\varepsilon\} \,||\, \pi_0(\boldsymbol{r}_2)) \\ &\subseteq \quad \pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2) \;\cup\; \pi(\boldsymbol{r}_1) \,||\{\varepsilon\} \;\cup\; \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2). \end{aligned}$$

Now suppose that

$$\pi_i(\boldsymbol{r}_1/\boldsymbol{r}_2) \subseteq \Delta(\boldsymbol{r}_1, \boldsymbol{r}_2),$$

where

$$\Delta(\boldsymbol{r}_1, \boldsymbol{r}_2) = \pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2) \;\cup\; \pi(\boldsymbol{r}_1) \,||\{\varepsilon\} \;\cup\; \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2).$$

Then, $\pi_{i+1}(\boldsymbol{r}_1/\boldsymbol{r}_2) = \pi_0(\pi_i(\boldsymbol{r}_1/\boldsymbol{r}_2))$ and

$$\pi_0(\pi_i(\boldsymbol{r}_1/\boldsymbol{r}_2)) \quad \subseteq \quad \pi_0(\pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2)) \;\cup\; \pi_0(\{\varepsilon\} \,||\, \pi(\boldsymbol{r}_1)) \;\cup\; \pi_0(\{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2)).$$

For every $\boldsymbol{r}/\boldsymbol{r}'' \in \pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2))$, $\pi_0(\boldsymbol{r}/\boldsymbol{r}'') \subseteq \Delta(\boldsymbol{r}_1, \boldsymbol{r}_2)$. So, we conclude that $\pi_0(\pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2)) \subseteq \Delta(\boldsymbol{r}_1, \boldsymbol{r}_2)$. Moreover, $\pi_0(\varepsilon/\boldsymbol{r}'') = \{\varepsilon\}||\pi_0(\boldsymbol{r}'')$, and thus

$$\pi_0(\{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2)) \quad \subseteq \quad \{\varepsilon\} \,||\, \pi_0(\pi(\boldsymbol{r}_3)) \subseteq \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2)).$$

In the same manner,

$$\pi_0(\pi(\boldsymbol{r}_1) \,||\{\varepsilon\}) \quad \subseteq \quad \pi_0(\pi(\boldsymbol{r}_1)) \,||\{\varepsilon\} \subseteq \pi(\boldsymbol{r}_1) \,||\{\varepsilon\}),$$

from which we conclude that $\pi_{i+1}(\boldsymbol{r}_1/\boldsymbol{r}_2) = \pi_0(\pi_i(\boldsymbol{r}_1/\boldsymbol{r}_2)) \subseteq \Delta(\boldsymbol{r}_1, \boldsymbol{r}_2)$. □

The following example shows that the inclusion in Proposition 8 may be strict.

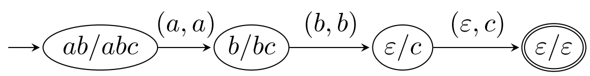
Example 2. *Let $\boldsymbol{r}_1 = ab$ and $\boldsymbol{r}_2 = abc$. We have $\pi(ab) = \{\varepsilon, b\}$, $\pi(abc) = \{\varepsilon, bc, c\}$ and $\mathsf{c}(ab) = \mathsf{c}(abc) = \emptyset$. To calculate $\pi(ab/abc)$ consider the following*

$$\begin{aligned} \pi_0(ab) \quad &= \quad \downarrow_2(\mathsf{n}(ab)) = \downarrow_2(\{(a, b)\}) = \{b\}, \\ \pi_0(abc) \quad &= \quad \downarrow_2(\mathsf{n}(abc)) = \downarrow_2(\{(a, bc)\}) = \{bc\}, \\ \pi_0(ab/abc) \quad &= \quad \{b/bc\}, \\ \pi_1(ab/abc) \quad &= \quad \pi_0(\{b/bc\}) = \{\varepsilon/c\}, \\ \pi_2(ab/abc) \quad &= \quad \pi_0(\{\varepsilon/c\}) = \{\varepsilon/\varepsilon\}. \end{aligned}$$

Thus, we have

$$\pi(ab/abc) = \{b/bc, \varepsilon/c, \varepsilon/\varepsilon\} \subset \pi(\boldsymbol{r}_1) \,||\, \pi(\boldsymbol{r}_2) \cup \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_1) \cup \{\varepsilon\} \,||\, \pi(\boldsymbol{r}_2).$$

The transducer $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{r}_1/\boldsymbol{r}_2)$ is depicted below.



$\square$

Proposition 8 and Proposition 2 ensure that for every $\boldsymbol{g} \in$ 2D-RE, the support $\pi(\boldsymbol{g})$ is finite and in the worst-case of size $O(n^2)$, where n is the alphabetic size of $\boldsymbol{g}$.

Corollary 1. *For all* $\boldsymbol{g} \in$ 2D-RE, $|\pi(\boldsymbol{g})| \leq (|\boldsymbol{g}|_{\Sigma\cup\Delta})^2$.

Proof. The proof follows by induction on the structure on $\boldsymbol{g}$, being trivial for $\emptyset$. For $\boldsymbol{r}_1/\boldsymbol{r}_2$, one has $|\boldsymbol{r}_1/\boldsymbol{r}_2|_{\Sigma\cup\Delta} = |\boldsymbol{r}_1|_\Sigma + |\boldsymbol{r}_2|_\Delta$. Thus,

$$\begin{aligned} |\pi(\boldsymbol{r}_1/\boldsymbol{r}_2)| &\leq |\pi(\boldsymbol{r}_1)||\pi(\boldsymbol{r}_2)| + |\pi(\boldsymbol{r}_1)| + |\pi(\boldsymbol{r}_2)| \\ &\leq |\boldsymbol{r}_1|_\Sigma|\boldsymbol{r}_2|_\Delta + |\boldsymbol{r}_1|_\Sigma + |\boldsymbol{r}_2|_\Delta \\ &\leq (|\boldsymbol{r}_1|_\Sigma + |\boldsymbol{r}_2|_\Delta)^2 \\ &= (|\boldsymbol{r}_1/\boldsymbol{r}_2|_{\Sigma\cup\Delta})^2. \end{aligned}$$

For $\boldsymbol{g}_1 + \boldsymbol{g}_2$, we have

$$\begin{aligned} |\pi(\boldsymbol{g}_1 + \boldsymbol{g}_2)| &\leq |\pi(\boldsymbol{g}_1)| + |\pi(\boldsymbol{g}_2)| \\ &\leq (|\boldsymbol{g}_1|_{\Sigma\cup\Delta})^2 + (|\boldsymbol{g}_2|_{\Sigma\cup\Delta})^2 \\ &\leq (|\boldsymbol{g}_1 + \boldsymbol{g}_2|_{\Sigma\cup\Delta})^2. \end{aligned}$$

In the same way we have the result for $\boldsymbol{g}_1\boldsymbol{g}_2$. For $\boldsymbol{g}_1^\star$,

$$\begin{aligned} |\pi(\boldsymbol{g}_1^\star)| &= |\pi(\boldsymbol{g}_1)| \\ &\leq (|\boldsymbol{g}_1|_{\Sigma\cup\Delta})^2 \\ &= (|\boldsymbol{g}_1^\star|_{\Sigma\cup\Delta})^2. \end{aligned}$$

$\square$

The following example shows that the quadratic blow-up is achieved.

Example 3. *Let* $\boldsymbol{r}_n = (a^\star)^n$, $n \geq 1$, *and consider the* 2D-RE *expressions* $\boldsymbol{r}_n/\boldsymbol{r}_n$. *Since* $\pi(\boldsymbol{r}_n) = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}$, *we have* $|\pi(\boldsymbol{r}_n)| = |\boldsymbol{r}_n|_\Sigma = n$. *Thus, we have* $|\boldsymbol{r}_n/\boldsymbol{r}_n|_{\Sigma\cup\Delta} = 2n$ *and*

$$|\pi(\boldsymbol{r}_n/\boldsymbol{r}_n)| \quad = \quad |\pi(\boldsymbol{r}_n)\,||\,\pi(\boldsymbol{r}_n) \;\cup\; \pi(\boldsymbol{r}_n)\,||\{\varepsilon\} \;\cup\; \{\varepsilon\}\,||\,\pi(\boldsymbol{r}_n)| = n^2 + 2n. \quad \square$$

The *partial derivative transducer* of a $\boldsymbol{g} \in$ 2D-RE is defined as follows

$$\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g}) = \langle \pi(\boldsymbol{g}) \cup \{\boldsymbol{g}\}, \Sigma, \Delta, \delta_{\mathrm{PD}}, \boldsymbol{g}, F\rangle,$$

where $F = \{\, \boldsymbol{g}_1 \in \pi(\boldsymbol{g}) \cup \{\boldsymbol{g}\} \mid \mathsf{c}(\boldsymbol{g}_1) = \varepsilon/\varepsilon \,\}$, and $\delta_{\mathrm{PD}} = \{\, (\boldsymbol{g}_1, (\gamma, \gamma')\,, \boldsymbol{g}') \mid \boldsymbol{g}_1 \in \pi(\boldsymbol{g}) \cup \{\boldsymbol{g}\} \wedge ((\gamma, \gamma')\,, \boldsymbol{g}') \in \mathsf{n}(\boldsymbol{g}_1) \,\}$.

Theorem 9. *For all $\boldsymbol{g} \in$ 2D-RE, $\mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})) = \mathcal{R}(\boldsymbol{g})$.*

Proof. The proof is by induction on the structure of $\boldsymbol{g}$. We only consider the case where $\boldsymbol{g}$ is $\boldsymbol{r}_1/\boldsymbol{r}_2$. The other cases follow from the proof for 1D regular expressions or from a more general proof considering expressions with user-defined labels over a finitely generated graded monoid [3]. First we note that if a state is labeled by an expression of the form $\boldsymbol{r}/\varepsilon$ (or $\varepsilon/\boldsymbol{r}$, respectively) it can only have transitions for states of the same form and labeled by pairs (σ, ε) ((ε, τ), respectively). This follows from the definition of the linear form and it means that from that point on we are in a (copy of a) subautomaton of one of the initial expressions. We begin by proving that $\mathcal{R}(\boldsymbol{r}_1/\boldsymbol{r}_2) \subseteq \mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{r}_1/\boldsymbol{r}_2))$. Let $(u, v) \in \mathcal{R}(\boldsymbol{r}_1/\boldsymbol{r}_2)$. If $(u, v) = (\varepsilon, \varepsilon)$ then $\mathsf{c}(\boldsymbol{r}_1) = \mathsf{c}(\boldsymbol{r}_2) = \varepsilon$ and $\mathsf{c}(\boldsymbol{r}_1/\boldsymbol{r}_2) = \varepsilon/\varepsilon$, and by construction the initial state of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{r}_1/\boldsymbol{r}_2)$ is final; thus $(u, v) \in \mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{r}_1/\boldsymbol{r}_2))$.

Now let $(u, v) \in \mathcal{R}(\boldsymbol{r}_1/\boldsymbol{r}_2)$ with $u = u_1 \cdots u_\ell \in \mathcal{L}(\boldsymbol{r}_1)$, $v = v_1 \cdots v_k \in \mathcal{L}(\boldsymbol{r}_2)$, $u_j \in \Sigma$, $v_i \in \Delta$ for $1 \leq j \leq \ell$ and $1 \leq i \leq k$ (and ℓ or k is greater than 0). We also know that $u \in \mathcal{L}\big(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_1)\big)$ and $v \in \mathcal{L}\big(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_2)\big)$. If $\ell \geq 1$, there will be an accepting path $P_1 = \langle s_{j-1}, u_j, s_j \rangle_{j=1}^{\ell}$ of $\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_1)$ with $s_0 = \boldsymbol{r}_1$, $\mathsf{c}(s_\ell) = \varepsilon$ and $(u_j, s_j) \in \mathsf{n}(s_{j-1})$ for all j. In the same way, if $k \geq 1$, there will be an accepting path $P_2 = \langle t_{i-1}, v_i, t_i \rangle_{i=1}^{k}$ of $\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_2)$ with $t_0 = \boldsymbol{r}_2$, $\mathsf{c}(t_k) = \varepsilon$ and $(v_i, t_i) \in \mathsf{n}(s_{i-1})$ for all i. We assume that $k \leq \ell$—the case $\ell \leq k$ is symmetric. Then, $((u_i, v_i), s_i/t_i) \in \mathsf{n}(s_{i-1}/t_{i-1})$ for all $i = 1, \ldots, k$, and $((u_j, \varepsilon), s_j/\varepsilon) \in \mathsf{n}(s_{j-1}/\varepsilon)$ for all $j = k+1, \ldots, \ell$. Let $t_j = v_j = \varepsilon$ for $j = k+1, \ldots, \ell$. Then $R = \langle s_{j-1}/t_{j-1}, (u_j, v_j), s_j/t_j \rangle_{j=1}^{\ell}$ is a path of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$ such that $\mathsf{c}(s_\ell/t_\ell) = (\varepsilon/\varepsilon)$, which implies that the path R is accepting and $(u, v) \in \mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g}))$ as required.

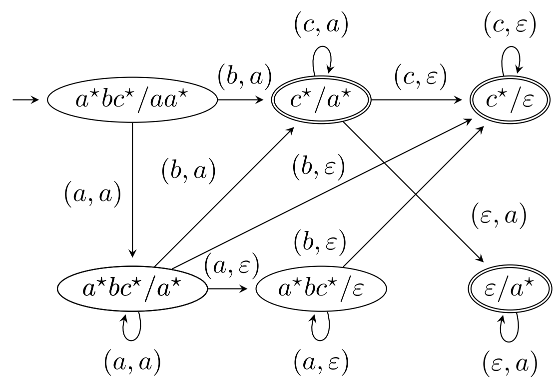
We now prove that $\mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})) \subseteq \mathcal{R}(\boldsymbol{g})$. Let $(u, v) \in \mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g}))$. If $(u, v) = (\varepsilon, \varepsilon)$, then the initial state of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$ is final ($\mathsf{c}(\boldsymbol{g}) = \varepsilon/\varepsilon$). We have also $\mathsf{c}(\boldsymbol{r}_1) = \mathsf{c}(\boldsymbol{r}_2) = \varepsilon$, and thus $\varepsilon \in \mathcal{L}(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_1)) = \mathcal{L}(\boldsymbol{r}_1)$, $\varepsilon \in \mathcal{L}(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_2)) = \mathcal{L}(\boldsymbol{r}_2)$ and $(u, v) \in \mathcal{R}(\boldsymbol{g})$.

Now let $(u, v) \in \mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g}))$ with $u \neq \varepsilon$ or $v \neq \varepsilon$. Then there exists an accepting path $R = \langle s_{i-1}/t_{i-1}, (u_i, v_i), s_i/t_i, \rangle_{i=1}^{m}$ of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$ with $s_0/t_0 = \boldsymbol{r}_1/\boldsymbol{r}_2$, $\mathsf{c}(s_m/t_m) = \varepsilon/\varepsilon$, $w = (u_1, v_1) \cdots (u_m, v_m)$, and $((u_i, v_i), s_i/t_i) \in \mathsf{n}(s_{i-1}/t_{i-1})$ for all i. If there exists a smallest $1 \leq \ell \leq m$ with $u_\ell = \varepsilon$ (resp., $1 \leq k \leq m$ with $v_k = \varepsilon$) then $u_j = \varepsilon$ and $s_j = \varepsilon$ (resp., $v_i = \varepsilon$ and $t_i = \varepsilon$) for all $j > \ell$ (resp., $i > k$) and $u = u_1 \cdots u_{\ell-1}$ (resp., $v = v_1 \cdots v_{k-1}$). If such an ℓ (resp., k) does not exist, let $\ell = m + 1$ (resp., $k = m + 1$). We can conclude that $u \in \mathcal{L}(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_1)) = \mathcal{L}(\boldsymbol{r}_1)$ (resp., $v \in \mathcal{L}(\mathcal{A}_{\mathrm{PD}}(\boldsymbol{r}_2)) = \mathcal{L}(\boldsymbol{r}_2)$), and thus $(u, v) \in \mathcal{R}(\boldsymbol{g})$. □

Example 4. *Considering $\boldsymbol{g}_1 = a^\star bc^\star/aa^\star$, we compute $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g}_1)$. The linear* *forms are*

$$
\begin{aligned}
\mathsf{n}(a^\star bc^\star) &= \{(a, a^\star bc^\star), (b, c^\star)\},\\
\mathsf{n}(aa^\star) &= \{(a, a^\star)\},\\
\mathsf{n}(c^\star) &= \{(c, c^\star)\},\\
\mathsf{n}(a^\star bc^\star/aa^\star) &= \{((a,a), a^\star bc^\star/a^\star), ((b,a), c^\star/a^\star)\},\\
\mathsf{n}(a^\star bc^\star/a^\star) &= \{((a,a), a^\star bc^\star/a^\star), ((b,a), c^\star/a^\star), ((a,\varepsilon), a^\star bc^\star/\varepsilon), ((b,\varepsilon), c^\star/\varepsilon)\},\\
\mathsf{n}(c^\star/a^\star) &= \{((c,a), c^\star/a^\star), ((c,\varepsilon), c^\star/\varepsilon), ((\varepsilon,a), \varepsilon/a^\star)\},
\end{aligned}
$$

where $\mathsf{n}(c^\star/\varepsilon)$*,* $\mathsf{n}(a^\star bc^\star/\varepsilon)$ *and* $\mathsf{n}(\varepsilon/a^\star)$ *can be computed from the above values. From these, we obtain the following SFT:*



□

An upper bound of the number of states of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$ is obtained if one assumes that

$$\pi(\boldsymbol{r}_1/\boldsymbol{r}_2) = \pi(\boldsymbol{r}_1)\,||\,\pi(\boldsymbol{r}_2) \;\cup\; \pi(\boldsymbol{r}_1)\,||\{\varepsilon\} \;\cup\; \{\varepsilon\}\,||\,\pi(\boldsymbol{r}_2)$$

always holds, and as usual $\pi(\boldsymbol{g}+\boldsymbol{g}') = \pi(\boldsymbol{g}) \cup \pi(\boldsymbol{g}')$, $\pi(\boldsymbol{g}\boldsymbol{g}') = \pi(\boldsymbol{g})\boldsymbol{g}' \cup \pi(\boldsymbol{g}')$, and $\pi(\boldsymbol{g}^\star) = \pi(\boldsymbol{g})\boldsymbol{g}^\star$. These equalities are used in Section 5 to obtain an upper bound for the average case size of partial derivative transducers. In Section 4 we set up the analytic combinatorics framework that allows to obtain those estimates.

3.2. Standard S2D-RE *Expressions*

To represent rational relations over Σ and Δ, one can just consider 1D expressions where basic terms correspond to the generators of $\Sigma^\star \times \Delta^\star$. Those expressions are called *standard* 2D regular expressions (S2D-RE) and are a particular case of the ones considered by Konstantinidis et al. [3].

A *standard* 2D *regular expression* (S2D-RE) over $\Sigma \times \Delta$, where Σ is the *input alphabet* and Δ the *output alphabet* is an expression that is either $\emptyset$, or can be defined by the following grammar

$$
\begin{aligned}
\beta &:= \sigma/\tau \mid \varepsilon/\tau \mid \sigma/\varepsilon\\
\boldsymbol{s} &:= \beta \mid \varepsilon/\varepsilon \mid (\boldsymbol{s}+\boldsymbol{s}) \mid (\boldsymbol{s}\cdot\boldsymbol{s}) \mid \boldsymbol{s}^\star,
\end{aligned}
\tag{6}
$$

where $\sigma \in \Sigma$ and $\tau \in \Delta$. The *relation* $\mathcal{R}(\boldsymbol{s}) \subseteq \Sigma^\star \times \Delta^\star$ realised by an S2D-RE $\boldsymbol{s}$ is defined as for general expressions. A relation is *rational* if it is represented by a S2D-RE.

The definition of linear form can be adapted directly from Equation (2) by considering $\mathsf{n}(\varepsilon/\varepsilon) = \emptyset$ and $\mathsf{n}(\gamma/\gamma') = \{((\gamma, \gamma'), \varepsilon/\varepsilon)\}$, where $\gamma \in \Sigma_\varepsilon$ and $\gamma' \in \Delta_\varepsilon$, but not both simultaneously equal to ε.

Proposition 10. *For all* $\boldsymbol{s} \in$ S2D-RE*,* $\boldsymbol{s} \sim \bigcup_{((\gamma,\gamma'),\boldsymbol{s}') \in \mathsf{n}(\boldsymbol{s})} (\gamma/\gamma')\boldsymbol{s}' \cup \mathsf{c}(\boldsymbol{s})$.

Proof. Follows from Proposition 1, except for the base cases. For $s = \varepsilon/\varepsilon$ is true because $\mathsf{c}(\varepsilon/\varepsilon) = \varepsilon/\varepsilon$. For σ/τ, $\mathsf{c}(\sigma/\tau) = \emptyset$, and the result follows. □

In order to obtain an SFT equivalent to a expression $\boldsymbol{s}$, one needs to iterate $\mathsf{n}(s)$, and thus compute a subset of the set of partial derivatives w.r.t to words in $\Sigma^\star \times \Delta^\star$. We note that the support satisfies Proposition 2 except for the base cases, which are now the following: $\pi(\varepsilon/\varepsilon) = \emptyset$ and $\pi(\beta) = \{\varepsilon/\varepsilon\}$, where β is as defined in (6). It follows that $\pi(\boldsymbol{s})$ is finite. The *partial derivative transducer* of $\boldsymbol{s}$ is $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{s}) = \langle \pi(\boldsymbol{s}) \cup \{\boldsymbol{s}\}, \Sigma, \Delta, \delta_{\mathrm{PD}}, \boldsymbol{s}, F\rangle$, where $F = \{\, \boldsymbol{s}_1 \in \pi(\boldsymbol{s}) \cup \{\boldsymbol{s}\} \mid \mathsf{c}(\boldsymbol{s}_1) = \varepsilon/\varepsilon \,\}$, and $\delta_{\mathrm{PD}} = \{\, (\boldsymbol{s}_1, (\gamma, \gamma'), \boldsymbol{s}') \mid \boldsymbol{s}_1 \in \pi(\boldsymbol{s}) \cup \{\boldsymbol{s}\} \wedge ((\gamma, \gamma'), \boldsymbol{s}') \in \mathsf{n}(\boldsymbol{s}_1) \,\}$.

Proposition 11. *For all* $\boldsymbol{s} \in$ S2D-RE*,* $\mathcal{R}(\mathcal{T}_{\mathrm{PD}}(\boldsymbol{s})) = \mathcal{R}(\boldsymbol{s})$.

Proof. It is an easy extension of Proposition 4 or it can also follows from a more general proof considering expressions with user-defined labels over a finitely generated graded monoid [3]. Also see [13]. □

Any 2D-RE can be transformed into an equivalent S2D-RE. For that one needs to transform the expressions of the form $\boldsymbol{r}/\boldsymbol{r}'$ where $\boldsymbol{r}, \boldsymbol{r}'$ are not both simultaneously ε, using the following equivalences, for $\circ \in \{+, \cdot\}$:

$$\begin{aligned}
\boldsymbol{r}_1/\boldsymbol{r}_2 &\sim (\boldsymbol{r}_1/\varepsilon) \cdot (\varepsilon/\boldsymbol{r}_2),\\
(\boldsymbol{r}_1 \circ \boldsymbol{r}_2)/\varepsilon &\sim (\boldsymbol{r}_1/\varepsilon) \circ (\boldsymbol{r}_2/\varepsilon),\\
\varepsilon/(\boldsymbol{r}_1 \circ \boldsymbol{r}_2) &\sim (\varepsilon/\boldsymbol{r}_1) \circ (\varepsilon/\boldsymbol{r}_2),\\
\boldsymbol{r}_1^\star/\varepsilon &\sim (\boldsymbol{r}_1/\varepsilon)^\star,\\
\varepsilon/\boldsymbol{r}_1^\star &\sim (\varepsilon/\boldsymbol{r}_1)^\star.
\end{aligned}$$

4. The Analytic Combinatorics Framework

Given some measure of the objects of a combinatorial class, $\mathcal{A}$, for each $n \in \mathbb{N}_0$ let a_n be the sum of the values of this measure for all objects of size n. Let $A(z) = \sum_n a_n z^n$ be the corresponding generating function (*cf.* [26]). We will use the notation $[z^n]A(z)$ for a_n. The generating function $A(z)$ can be seen as a complex analytic function, and the study of its behaviour around its dominant singularity ρ, when unique, gives us access to the asymptotic form of its coefficients. In particular, if $A(z)$ is analytic in some indented disc neighbourhood of ρ, then one has the following [26, 12]:

Theorem 12. *The coefficients of the series expansion of the complex function*

$$f(z) = (1-z)^{\alpha},$$

where $\alpha \in \mathbb{C} \setminus \mathbb{N}_0$*, have the following asymptotic approximation:*

$$[z^n]f(z) = \frac{n^{-\alpha-1}}{\Gamma(-\alpha)} + o\left(n^{-\alpha-1}\right).$$

Here Γ *is Euler's gamma function.*

The combinatorial classes that we deal with in the present paper give rise to generating functions implicitly defined by algebraic curves that are quite a bit more convoluted than those previously described in the literature. We, therefore, needed to refine the method to pursue these calculations, and we will expound that, in some detail, here. Generically, from an unambiguous generating grammar, one obtains a set of polynomial equations involving the generating functions for the objects corresponding to the variables of the grammar, in particular the one whose coefficients we want to asymptotically estimate. Computing a Gröbner basis for the ideal generated by those polynomials, one gets an algebraic equation for that generating function $w = w(z)$, i.e., an equation of the form

$$G(z, w) = 0,$$

where $G(z, w)$ is a polynomial in $\mathbb{Z}[z][w]$ of which $w(z)$ is a root.

Since $w(z)$ is the generating function of a combinatorial class, thus a series with non-negative integer coefficients which is not a polynomial, it must have, by Pringsheim's Theorem (*cf.* [26], Thm IV.6), a real positive singularity, ρ, smaller than or equal to 1. In all that follows we will assume that there is no other singularity with that norm, which is the case of all generating functions dealt with in this paper, as we will see. At this singularity, ρ, two cases may occur:

Case I: $\lim_{z\to\rho} w(z) = a$, where a is a positive real number.

Case II: $\lim_{z\to\rho} w(z) = +\infty$.

In the first case the curve defined by G has a shape similar to the one depicted in Fig. 1, on the left, and

$$\frac{\partial G}{\partial w}(\rho, a) = 0. \tag{7}$$

This, together with the fact that $G(\rho, a) = 0$, shows that ρ is a root of the resultant,

$$\operatorname{res}_w(G(z, w), \frac{\partial G}{\partial w}(z, w)),$$

of $G(z, w)$ and $\frac{\partial G}{\partial w}(z, w)$ with respect to w (*cf.* [27, p. 204]).

With the help of a numerical solver and drawing the relevant part of the algebraic curve $G(z, w) = 0$, one can, by an elimination process, find out the

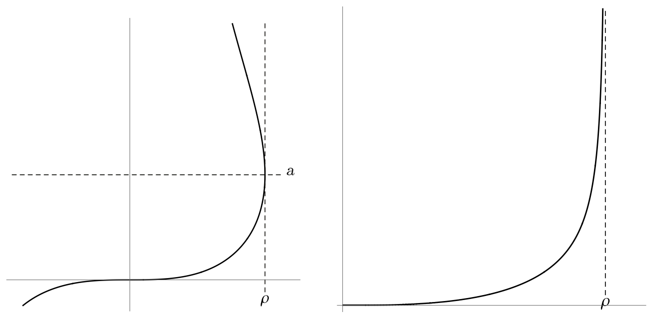


Figure 1: Generic shape of $G(z, w)$ near its dominant singularity (cases I and II).

minimum polynomial, in $\mathbb{Q}[z]$, of ρ. We will denote this polynomial by $m(z)$. Using now the

$$\mathrm{res}_z(G(z, w), \frac{\partial}{\partial w} G(z, w)),$$

one can get, in a similar fashion, an irreducible polynomial that has a as a root.

In Case II, the irreducible polynomial for ρ is a factor of the leading coefficient of $G(z, w)$, seen as a polynomial in w (*cf* [28], Th. 12.2.1).

In Case I, after making the change of variable $s = 1 - z/\rho$, one knows that $w = w(s)$ has a Puiseux series expansion at the singularity $s = 0$, i.e., there exists a slit neighbourhood of that point in which $w(s)$ has a representation as a power series with fractional powers (*cf.* [28], Chap. 12). In particular, w must have the form

$$w(s) = a - g(s)s^{\alpha}, \tag{8}$$

for some $\alpha \in \mathbb{Q}^+$, the first positive exponent of that expansion, and where $g(s)$ is such that

$$g(s) = b + h(s)s^{\beta}, \tag{9}$$

with $h(0) \neq 0$, $\beta \in \mathbb{Q}^+$, and $b \in \mathbb{R}^{\star}$.

The value of α can be obtained by looking at the Taylor expansion of $G(z, w)$ at (ρ, a),

$$G(z, w) = \sum_{i,j \geq 0} \frac{1}{i!j!} \frac{\partial^{i+j} G}{\partial z^i w^j} \bigg|_{\substack{z=\rho \\ w=a}} (z - \rho)^i (w - a)^j.$$

Noticing that $z = \rho - \rho s$, and using Equation (8), one has

$$G(\rho - \rho s, a - g(s)s^{\alpha}) = \sum_{i,j \geq 0} \frac{(-1)^{i+j}}{i!j!} \frac{\partial^{i+j} G}{\partial z^i w^j} \bigg|_{\substack{z=\rho \\ w=a}} \rho^i g(s)^j s^{i+j\alpha}. \tag{10}$$

Using that $G(z, w(z)) = 0$, $G(\rho, a) = 0$, and (7), and dividing it through by s^{α}, one gets

$$0 = \sum_{\substack{i,j \geq 0 \\ (i,j) \notin \{(0,0),(0,1)\}}} \frac{(-1)^{i+j}}{i!j!} \frac{\partial^{i+j} G}{\partial z^i w^j} \bigg|_{\substack{z=\rho \\ w=a}} \rho^i g(s)^j s^{i+(j-1)\alpha}. \tag{11}$$

One can now compute

$$p_{ij}(z) = \mathrm{res}_w \left(G(z,w), \frac{\partial^{i+j} G}{\partial z^i w^j} \right),$$

and $\gcd(p_{ij}(z), m(z))$ to see which derivatives are non-zero at ρ. Then, one can use the Newton's polygon technique to find α [29, 30, 31]. The interested reader can also see Broda et al. [11] for a more detailed explanation of the use of that tecnique in this context.

The points of Newton polygon that lead to the value of α correspond to the terms of (11) with the lowest exponent, that must cancel out together. This conduces, after setting $s = 0$, to a polynomial equation for the value b in (9).

One then uses this value in Theorem 12 to get the desired asymptotic approximation. In conclusion, for the case where $\lim_{z\to\rho} w(z) = a$, one has

$$[z^n]w(z) \sim \frac{-b}{\Gamma(-\alpha)} \rho^{-n} n^{-\alpha-1}. \tag{12}$$

In Case II, the one where $\lim_{z\to\rho} w(z) = +\infty$, making $v = \frac{1}{w}$ one concludes as above that

$$v = cs^\alpha - g(s)s^{\alpha+\beta},$$

for some $0 < \alpha < 1$, $\beta > 0$, and for some Puiseux series $g(s)$, with non-negative exponents. Denoting by m the degree of G relative to w, the polynomial satisfied by v is then

$$H(z,v) = v^m G\left(z, \frac{1}{v}\right), \tag{13}$$

which is the reciprocal polynomial of $G(z,w)$ with respect to the variable w. In this case the equation that corresponds to equation (10) is:

$$H(\rho - \rho s, cs^\alpha - g(s)s^{\alpha+\beta}) = \sum_{i,j\geq 0} \frac{(-1)^i}{i!j!} \frac{\partial^{i+j} H}{\partial z^i w^j}\bigg|_{\substack{z=\rho\\ w=0}} \rho^i (c - g(s)s^\beta)^j s^{i+j\alpha}. \tag{14}$$

Using the same procedure as above, one computes ρ, and then the value of c. Since

$$\begin{aligned} w = \frac{1}{cs^\alpha - g(s)s^{\alpha+\beta}} &= \frac{1}{c} s^{-\alpha} \frac{1}{1 - \frac{g(s)}{c} s^\beta} \\ &= \frac{1}{c} s^{-\alpha} \left(1 + \frac{g(s)}{c} s^\beta + \frac{g(s)^2}{c^2} s^{2\beta} + \cdots \right), \end{aligned}$$

one sees, using again Theorem 12, that

$$[z^n]w(z) \sim \frac{1}{c\,\Gamma(\alpha)} \rho^{-n} n^{\alpha-1}. \tag{15}$$

Summing up, we have the following.

Theorem 13. *With the notations and in the conditions above described, one has*

$$[z^n]w(z) \sim \begin{cases} \frac{-b}{\Gamma(-\alpha)}\rho^{-n}n^{-\alpha-1}, & \text{if } \lim_{z\to\rho} w(z) = a, \\ \frac{1}{c\,\Gamma(\alpha)}\rho^{-n}n^{\alpha-1}, & \text{if } \lim_{z\to\rho} w(z) = +\infty, \end{cases}$$

where b, c, ρ and α can be computed as above described.

5. Average Descriptional Complexity Results

Using the framework just described, we obtain asymptotic estimates for an upper bound of the average state complexity of partial derivative transducer for 2D expressions of size $n > 0$. Those estimates depend on the size of the alphabets Σ and Δ, which we assume both to be equal to some integer $k > 0$. Moreover we denote by RE_k the set of 1D expressions over an alphabet of size k.

5.1. Average State Complexity of $\mathcal{T}_{\mathrm{PD}}$ for 2D-RE

The generating function $G_k(z)$ associated with $\boldsymbol{g} \in$ 2D-RE is the following[5], where $R_k(z)$ is the generating function of regular expressions $\boldsymbol{r} \in \mathrm{RE}_k$ [12].

$$\begin{aligned} G_k(z) &= zR_k(z)^2 + zG_k(z) + 2zG_k(z)^2, & (16)\\ R_k(z) &= (k+1)z + zR_k(z) + 2zR_k(z)^2. & (17) \end{aligned}$$

Considering Proposition 2, let $\ell(\boldsymbol{r})$ be an upper bound of the size of the support of an expression $\boldsymbol{r} \in \mathrm{RE}_k$ [10] which is defined by

$$\begin{gathered} \ell(\varepsilon) = 0,\ \ell(\sigma) = 1,\ \ell(\boldsymbol{s}^\star) = \ell(\boldsymbol{s}), \\ \ell(\boldsymbol{s} + \boldsymbol{s}') = \ell(\boldsymbol{s} \cdot \boldsymbol{s}') = \ell(\boldsymbol{s}) + \ell(\boldsymbol{s}'). \end{gathered} \tag{18}$$

Note that $\ell(\boldsymbol{r})$ counts exactly the number of letters occurring in $\boldsymbol{r}$ and will be used in Section 5.3. An upper bound for the size of the support $\pi(\boldsymbol{g})$, $q(\boldsymbol{g})$, is defined by

$$\begin{aligned} q(\boldsymbol{r}/\boldsymbol{r}') &= \ell(\boldsymbol{r})\ell(\boldsymbol{r}') + \ell(\boldsymbol{r}) + \ell(\boldsymbol{r}'), \\ q(\boldsymbol{g} + \boldsymbol{g}') &= q(\boldsymbol{g} \cdot \boldsymbol{g}') = q(\boldsymbol{g}) + q(\boldsymbol{g}'), \\ q(\boldsymbol{g}^\star) &= q(\boldsymbol{g}). \end{aligned}$$

Thus, the generating function $Q_k(z) = \sum_{\boldsymbol{g}} q(\boldsymbol{g}) z^{|\boldsymbol{g}|}$ for an upper bound of $|\pi(\boldsymbol{g})|$ satisfies the following equation,

$$Q_k(z) = zQ_k(z) + 4zQ_k(z)G_k(z) + 2zP_k(z)R_k(z) + zP_k(z)^2, \tag{19}$$

where $P_k(z)$ is the generating function for an upper bound of the support of regular expressions in RE_k, which satisfy

$$P_k(z) = kz + zP_k(z) + 4zR_k(z)P_k(z). \tag{20}$$

[5] I.e. $[z^n]G_k(z)$ gives the number of expressions $\boldsymbol{g}$ of size n.

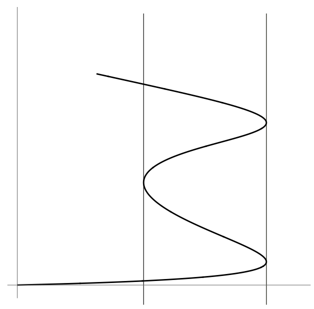


Figure 2: Possible values for (ρ_k, a_k).

From equations (17), (20), (16) and (19), using Gröbner basis, one obtains algebraic equations for $G_k(z)$ and $Q_k(z)$:

$$\mathcal{C}_G(z,w) = 16z^3w^4 + 16(z^3 - z^2)w^3 - g_2(z)w^2 + g_1(z)w + (1+k)^2z^3 = 0, \quad (21)$$

where $g_2(z) = 2z((1+4k)z^2 + 6z - 3)$ and $g_1(z) = (1-z)((3+4k)z^2 + 2z - 1)$ and

$$\mathcal{C}_Q(z,w) = p(z)^4 q_4(z)w^4 - k^2z^2p(z)^2q_2(z)w^2 + k^4z^8q_0(z)^2 = 0, \qquad (22)$$

where

$$p(z) = (8k+7)z^2 + 2z - 1,$$

$$q_4(z) = (16k^2 + 40k + 23)z^4 - 4(4k+3)z^3 + (8k+2)z^2 + 4z - 1,$$

$$q_2(z) = (200k^3 + 544k^2 + 474k + 133)z^6 - (48k^2 + 24k - 10)z^5 + (24k^2 - 44k - 41)z^4 + 28(2k+1)z^3 + (3-14k)z^2 - 6z + 1,$$

$$q_0(z) = (25k^2 + 37k + 14)z^2 + (6k+4)z - (3k+2).$$

For $G_k(z)$, we conclude to be in Case I. The irreducible polynomial that implicitly defines the singularity ρ_k of $G_k(z)$ is, computed using the resultant

$$\mathrm{res}_w(\mathcal{C}_G(z,w), \frac{\partial \mathcal{C}_G}{\partial w}(z,w)).$$

In this case we obtain two candidates for the minimal polynomial $m_G(z)$ of the singularity ρ_k, each one having only one root in $]0,1[$. Using a computer algebra system, one can show that those roots are only equal for $k = -1$. This implies, by continuity (in k), that they always keep their relative position, for all $k > -1$. Now,

$$\mathrm{res}_z(\mathcal{C}_G(z,w), \frac{\partial \mathcal{C}_G}{\partial w}(z,w))$$

factors into three irreducible polynomials, one of which has a_k as a root. These three polynomials have, among them, four positive roots, which a computer algebra system can find, as a function of k. Then, one can check which pairs (ρ'_k, a'_k), where ρ'_k is a candidate for ρ_k, and a'_k a candidate for a_k, belong to the curve $\mathcal{C}_G$, and their relative location. By a simple topological argument, one then can conclude that

$$m_G(z) = (8k+7)z^2 + 2z - 1, \rho_k = \frac{1}{1+\sqrt{8k+8}} \text{ and } a_k = \frac{\sqrt{2}-1}{2}\sqrt{k+1}.$$

One then checks that $\frac{\partial \mathcal{C}_G}{\partial z}(\rho_k, a_k)$ and $\frac{\partial^2 \mathcal{C}_G}{\partial w^2}(\rho_k, a_k)$ are both non-zero, for all k, which entails that $\alpha = \frac{1}{2}$. The value for b_k can then be computed, and one obtains

$$b_k \sim \sqrt{\frac{k}{2}}. \tag{23}$$

As for $Q_k(z)$, one sees that Case II applies, and that the minimal polynomial is either $p(z)$ or $q_4(z)$. It turns out that each of these polynomials has exactly one positive real root, ρ_k and ζ_k. One can then check that these roots coincide only for $k = -1$, and so that one of them is always bigger than the other for all positive values of k, namely ρ_k. One then can check that the curve $\mathcal{C}_Q$ crosses the vertical line $z = \zeta_k$ exactly once above the z-axis, which makes clear that the singularity for $Q_k(z)$ is ρ_k, thus the same as for $G_k(z)$. In this case, the Newton polygon analysis shows that $\alpha = 1$ and that the polynomial satisfied by c, as explained after (11), and noticing that here we make use of inversion explained in (13), is given by

$$\left.\frac{\partial^4 H}{\partial v^4}\right|_{\substack{z=\rho\\ v=0}} c^4 + 6\left.\frac{\partial^4 H}{\partial z^2 \partial v^2}\right|_{\substack{z=\rho\\ v=0}} \rho^2\, c^2 + \left.\frac{\partial^4 H}{\partial z^4}\right|_{\substack{z=\rho\\ v=0}} \rho^4 = 0.$$

This is a quadratic equation in c^2, whose discriminant can be seen to be zero. One gets

$$c_k^2 = -3\rho_k^2 \left(\left.\frac{\partial^4 H}{\partial z^2 \partial v^2}\right|_{\substack{z=\rho\\ v=0}} \right) \Bigg/ \left(\left.\frac{\partial^4 H}{\partial v^4}\right|_{\substack{z=\rho\\ v=0}} \right). \tag{24}$$

From all this, it follows that

Theorem 14. *With the notations above introduced, the ratio of the total number of states in the partial derivative transducer $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$ of expressions of size n to the total number of expressions of the same size is given by*

$$\frac{[z^n]\, Q_k(z)}{[z^n]\, G_k(z)} \sim \frac{-\Gamma(-\frac{1}{2})}{b_k c_k} n^{\frac{3}{2}}, \text{ for all } k, \text{ and } \lim_{k\to\infty} \frac{-\Gamma(-\frac{1}{2})}{b_k c_k} = \frac{\sqrt{\pi}}{8\sqrt{2}}.$$

5.2. Average State Complexity of $\mathcal{T}_{\mathrm{PD}}$ for pairs of REs

If we consider only 2D-expressions of the form $\boldsymbol{r}/\boldsymbol{r}'$,with $\boldsymbol{r}, \boldsymbol{r}' \in$ RE, the generating function for these expressions is

$$G'_k(z) = zR_k^2(z),$$

and for the support π is, following Proposition 8,

$$Q'_k(z) = 2zP_k(z)R_k(z) + zP_k^2(z).$$

From these, one can deduce the following algebraic equations for $G'_k(z)$ and $Q'_k(z)$:

$$\mathcal{C}_{G'}(z,w) = 4zw^2 + ((4k+3)z^2 + 2z - 1)w + (k+1)^2z^2 = 0, \tag{25}$$

and

$$\mathcal{C}_{Q'}(z,w) = p(z)^2w^2 + kzg'_1(z)w + k^2z^4g'_0(z) = 0, \tag{26}$$

where $p(z)$ is as above, and

$$\begin{aligned} g'_1(z) &= (80k^2 + 126k + 49)z^4 + 4(9k+7)z^3 - 2(9k+5)z^2 - 4z + 1, \\ g'_0(z) &= (25k^2 + 37k + 14)z^2 + (6k+4)z - 3k - 2. \end{aligned}$$

Let us first deal with $G'_k(z)$. We easily conclude that we are in Case I. The irreducible polynomial that implicitly defines the singularity ρ_k of $G'_k(z)$ is computed using

$$\mathrm{res}_w(\mathcal{C}_{G'}(z,w), \frac{\partial \mathcal{C}_{G'}}{\partial w}(z,w)).$$

In this case we obtain a single candidate for the minimal polynomial, $m_{G'}(z)$, of the singularity, ρ_k, namely

$$m_{G'}(z) = (8k+7)z^2 + 2z - 1,$$

and thus $\rho_k = \frac{1}{1+\sqrt{8k+8}}$. One has

$$\mathrm{res}_w(\mathcal{C}_{G'}(z,w), \frac{\partial \mathcal{C}_{G'}}{\partial w}(z,w)) = 4(7+8k)w^2 + 4(1+k)w - (1+k)^2,$$

from which one gets

$$a_k = \frac{-(1+k) + 2(1+k)\sqrt{2(1+k)}}{2(7+8k)},$$

where $a_k = G'_k(\rho_k)$. Using now the Newton's polygon method, one gets that $\alpha = \frac{1}{2}$, and

$$b_k = \sqrt{\frac{2\rho_k \frac{\partial \mathcal{C}_{G'}}{\partial z}(\rho_k, a_k)}{\frac{\partial^2 \mathcal{C}_{G'}}{\partial w^2}(\rho_k, a_k)}} \sim \frac{\sqrt{k}}{2}.$$

As for Q'_k, one sees that one is in Case II, and that the dominant singularity is the same as for G'_k. Using the methods expounded above, one gets that $\alpha = 1$, and that c_k is a zero of the equation

$$\frac{\partial^2 H}{\partial v^2}(\rho_k, 0)c_k^2 - 2\rho_k \frac{\partial^2 H}{\partial z \partial v}(\rho_k, 0)c_k + \rho_k^2 \frac{\partial^2 H}{\partial z^2}(\rho_k, 0) = 0,$$

where

$$H(z,v) = v^2 G_{Q'}(z, \frac{1}{v}).$$

It turns out that this equation has a single solution, namely

$$c_k = \frac{4}{k^2}\left(8 + 8k + (9+8k)\sqrt{2+2k}\right) \sim 32\sqrt{\frac{2}{k}}.$$

Therefore, in this case an upper bound of the average state complexity of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{r}/\boldsymbol{r}')$ is,

Theorem 15. *With the notations above introduced, one has*

$$\frac{[z^n]\, Q_k'(z)}{[z^n]\, G_k'(z)} \sim \frac{-\Gamma(-\frac{1}{2})}{b_k c_k}\, n^{\frac{3}{2}} \sim \frac{\sqrt{\pi}}{8\sqrt{2}}\, n^{\frac{3}{2}}.$$

We conclude that, as the alphabetic size grows, the upper bound of the average state complexity of $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{r}/\boldsymbol{r}')$ is exactly the same as for $\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$, for $\boldsymbol{g} \in$ 2D-RE[6].

5.3. Average Number of Letters in 2D-RE

Let $\ell(\boldsymbol{r})$ be the number of letters in $\boldsymbol{r} \in \mathrm{RE}_k$ as computed in (18), the number of letters in a expression $\boldsymbol{g} \in$ 2D-RE, $d(\boldsymbol{g})$, can be computed in the following way:

$$\begin{aligned} d(\boldsymbol{r}/\boldsymbol{r}') &= \ell(\boldsymbol{r}) + \ell(\boldsymbol{r}'), \\ d(\boldsymbol{g}+\boldsymbol{g}') &= d(\boldsymbol{g}\cdot\boldsymbol{g}') = d(\boldsymbol{g}) + d(\boldsymbol{g}'), \\ d(\boldsymbol{g}^\star) &= d(\boldsymbol{g}). \end{aligned}$$

Thus, the generating function $D(z) = \sum_{\boldsymbol{g}} d(\boldsymbol{g}) z^{|\boldsymbol{g}|}$ for the number of letters in expressions $\boldsymbol{g} \in$ 2D-RE satisfies the following

$$D_k(z) = 2zL_k(z)R_k(z) + zD_k(z) + 4zD_k(z)G_k(z), \tag{27}$$

$$L_k(z) = kz + zL_k(z) + 4zL_k(z)R_k(z), \tag{28}$$

where $L_k(z)$ is the generating function for the number of letters in $\boldsymbol{r} \in \mathrm{RE}_k$. From equations (17), (28), and (16), using Gröbner basis, one obtains the following algebraic equation for $D_k(z)$:

$$\mathcal{C}_D(z,w) = p(z)^2 q_4(z) w^4 - k^2 z^2 p(z) d_2(z) w^2 + 4k^4(k+1)^2 z^8 = 0, \tag{29}$$

where

$$d_2(z) = (16k^2 + 36k + 19)z^4 + (-8k-4)z^3 + (4k-2)z^2 + 4z - 1.$$

[6] In the conference paper [1] there was an error in the definition of $G'(z)$ that lead to a slightly different limit.

One can solve (29) to explicitly get the function $w = w(z)$, obtaining:

$$
\begin{aligned}
w(z) &= kz\sqrt{\frac{d_2(z)+(1-z)^3\sqrt{-p(z)}}{2p(z)q_4(z)}} && (30)\\
&= \frac{2\sqrt{2}\,k(k+1)z^3}{\sqrt{p(z)\left(d_2(z)-(1-z)^3\sqrt{-p(z)}\right)}}, && (31)
\end{aligned}
$$

and then one can see that $d_2(z)$ has no roots in the interval $[0, \rho_k]$, using Sturm's theorem, and therefore it is always negative. This shows that

$$d_2(z) - (1-z)^3\sqrt{-p(z)}$$

has no roots in that same interval, and hence the singularity comes necessarily from $p(z)$, being therefore the same as for $G_k(z)$, i.e.

$$\rho_k^D = \rho_k^G = \frac{1}{1+\sqrt{8k+8}}.$$

We are, thus, dealing with Case II, and an analysis entirely similar to what was done above for $Q_k(z)$ yields $\alpha = \frac{1}{2}$, and

$$\left.\frac{\partial^4 H}{\partial v^4}\right|_{\substack{z=\rho\\ v=0}} c^4 - 12\left.\frac{\partial^3 H}{\partial z\partial v^2}\right|_{\substack{z=\rho\\ v=0}} \rho c^2 + 12\left.\frac{\partial^2 H}{\partial z^2}\right|_{\substack{z=\rho\\ v=0}} \rho^2 = 0.$$

This is a quadratic equation in c^2, whose discriminant turns out to be zero. One then gets

$$c_k^2 = 6\rho_k\left(\left.\frac{\partial^3 H}{\partial z\partial v^2}\right|_{\substack{z=\rho\\ v=0}}\right)\Big/\left(\left.\frac{\partial^4 H}{\partial v^4}\right|_{\substack{z=\rho\\ v=0}}\right),$$

or

$$c_k = \frac{2}{k}\sqrt{8+8k+\sqrt{8+8k}}.$$

From all this, it follows that

Theorem 16. *With the notations above introduced, the ratio of the total number of letters in a* 2D-RE *is*

$$\lim_{n\to\infty}\frac{[z^n]\,D_k(z)}{n[z^n]\,G_k(z)} = \frac{2}{b_k c_k},\ \textit{for all } k, \textit{and}\ \lim_{k\to\infty}\frac{2}{b_k c_k} = \frac{1}{2}. \qquad (32)$$

5.4. Average State Complexity of $\mathcal{T}_{\mathrm{PD}}$ *for* S2D-RE

For the S2D-RE expressions, from the grammar (6) one sees that the generating functions for β and $\boldsymbol{s}$, respectively $B_k(z)$ and $S_k(z)$, satisfy the following equations:

$$
\begin{aligned}
B_k(z) &= (k^2+2k)z^3,\\
S_k(z) &= (k^2+2k+1)z^3 + zS_k(z) + 2zS_k(z)^2.
\end{aligned} \qquad (33)
$$

Letting $t(\boldsymbol{s})$ be the size of the support $\pi(\boldsymbol{s})$, we have by Proposition 2 and Proposition 5,

$$\begin{aligned} t(\varepsilon/\varepsilon) &= 0,\ t(\beta) = 1, \\ t(\boldsymbol{s}+\boldsymbol{s}') &= t(\boldsymbol{s}\cdot\boldsymbol{s}') = t(\boldsymbol{s}) + t(\boldsymbol{s}'), \\ t(\boldsymbol{s}^\star) &= t(\boldsymbol{s}). \end{aligned}$$

So, the generating function for π, $T_k(z) = \sum_{\boldsymbol{s}} t(\boldsymbol{s})z^{|\boldsymbol{s}|}$, satisfies the following equation:

$$T_k(z) = B_k(z) + zT_k(z) + 4zT_k(z)S_k(z). \tag{34}$$

From (33), we know that the generating function $S_k(z)$ is an algebraic function, root of

$$\mathcal{C}_S(z,w) = 2zw^2 + (z-1)w + (k+1)^2z^3 = 0.$$

Since the leading coefficient, in w, has no positive roots, we are, therefore, in Case I, as explained above. Thus, in order to find the irreducible polynomial that implicitly defines the singularity ρ_k of $S_k(z)$, we compute

$$\operatorname{res}_w(\mathcal{C}_S(z,w), \frac{\partial \mathcal{C}_S}{\partial w}(z,w)),$$

obtaining $2z\, m_S(z)$, where

$$\begin{aligned} m_S(z) &= 8(k+1)^2z^4 - (z-1)^2 \\ &= (\sqrt{8}\,(k+1)z^2 - z + 1)(\sqrt{8}\,(k+1)z^2 + z + 1). \end{aligned}$$

From this we easily get that

$$\rho_k = \frac{-1 + \sqrt{1 + 8\sqrt{2}\,(k+1)}}{4\sqrt{2}\,(k+1)}.$$

On the other hand,

$$\operatorname{res}_z(\mathcal{C}_S(z,w), \frac{\partial \mathcal{C}_S}{\partial w}(z,w)) = \left(\sqrt{2}w\,(4w+1) - k - 1\right)\left(\sqrt{2}w\,(4w+1) + k + 1\right),$$

from which one gets

$$a_k = \frac{1}{8}\left(-1 + \sqrt{8\sqrt{2}\,(k+1) + 1}\right),$$

where $a_k = S_k(\rho_k)$.

Using this, one can check that $\dfrac{\partial \mathcal{C}_S}{\partial z}(\rho_k, a_k)$ and $\dfrac{\partial^2 \mathcal{C}_S}{\partial w^2}(\rho_k, a_k)$ are both non-zero, for all k. Therefore, using Newton's polygon method, as above explained, one gets that the value of α in (8) is $\frac{1}{2}$, and that the value for b_k (as in (12)) is

$$b_k = \sqrt{\frac{a_k(8a_k+1)}{2}}. \tag{35}$$

From (33) and (34), using Buchberger's algorithm, one obtains that the generating function $T_k(z)$ satisfies

$$\mathcal{C}_T(z,w) = m_S(z)\, w^2 + k^2(k+2)^2 z^6 = 0,$$

and, clearly this falls under Case II, and $T_k(z)$ has the same singularity as $S_k(z)$. One also sees that $\alpha = \frac{1}{2}$ in the case, and proceeding as explained in the previous section, one can compute c_k (as in (15)), obtaining

$$c_k = \frac{(k+1)^2 \sqrt{k+1+\sqrt{2}\, a_k\, (32a_k^2 - 1)}}{2^{3/4} k\, (k+2)\, a_k^{\frac{5}{2}}}. \tag{36}$$

From all this, it follows that,

Theorem 17. *With the notations above introduced, the ratio of the total number of states in the partial derivative automata for a* S2D-RE *is*

$$\lim_{n\to\infty} \frac{[z^n]\, T_k(z)}{n[z^n]\, S_k(z)} = \frac{2}{b_k c_k},\ \textit{for all } k, \textit{and} \lim_{k\to\infty} \frac{2}{b_k c_k} = \frac{1}{4}. \tag{37}$$

The above ratio is the same as the one for 1D expressions [10].

6. Experimental Results

We ran some experiments by uniformly generating 2D-RE expressions and computing the size of the corresponding partial derivative transducer, using the FAdo system [32]. For the results to be statistically significant, expressions were uniformly random generated using a version of the grammar in prefix notation.

For each alphabet size $k \in \{2, 5, 20, 50\}$ and for each size of expressions $n \in \{50, 100, 200\}$, samples of 1000 regular expressions were generated. This is sufficient to ensure a 95% confidence level within a 1% error margin. For each sample we computed the average of the alphabetic size ($|\boldsymbol{g}|_{\Sigma\cup\Delta}$) and the size (number of states) of the partial derivative transducer ($\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})$). In Table 1 are some results for expressions of the form $\boldsymbol{r}/\boldsymbol{r}$ (which are the ones that lead to a larger blow-up). The results suggest that the average size of $\mathcal{T}_{\mathrm{PD}}$ is even smaller that the upper bound obtained in Section 5.2.

7. Conclusions

We considered partial derivative transducers for 2D regular expressions over pairs of 1D regular expressions. For studying the average state complexity, and given the intricacy of the resulting generating functions, we refine known methods within the analytic combinatorics framework. In Section 5, we conclude that for 2D expressions of size n, both general and restricted, asymptotically and on average, the state complexity of the partial derivative transducers is bounded from above by $O(n^{\frac{3}{2}})$. For ordinary 1D regular expressions, the number of letters

Table 1: Experimental Results for 2D-REs of the form $\boldsymbol{r}/\boldsymbol{r}$.

k	$\lvert\boldsymbol{g}\rvert$	$\lvert\boldsymbol{g}\rvert_{\Sigma\cup\Delta}$	$\lvert\mathcal{T}_{\mathrm{PD}}(\boldsymbol{g})\rvert$	$\frac{\lvert\boldsymbol{g}\rvert}{\lvert\boldsymbol{g}\rvert_{\Sigma\cup\Delta}}$	$\frac{Q'_k}{G'_k}$
	50	20.5	24.4	0.41	55.39
2	100	40.5	49.5	0.41	156.66
	200	80.5	160.4	0.40	443.11
	50	22.1	23.0	0.44	55.39
5	100	43.7	55.4	0.44	156.66
	200	95.7	109.3	0.47	443.11
	100	46.8	48.4	0.47	156.66
20	200	93.1	111.2	0.47	443.11
	100	42.5	48.1	0.48	156.66
50	200	95.8	104.7	0.48	443.11

in an expression is, asymptotically and on average, $\frac{1}{2}n$ [14, 10]. The same holds for general 2D expressions. Considering standard 2D expressions of size n, asymptotically and on average, the state complexity of the partial derivative transducers is bounded from above by $O(n/4)$. This bound coincides with the one for the partial derivative automata for 1D ordinary regular expressions.

References

[1] S. Konstantinidis, A. Machiavelo, N. Moreira, R. Reis, On the average state complexity of partial derivative transducers, in: A. Chatzigeorgiou, R. Dondi, H. Herodotou, C. A. Kapoutsis, Y. Manolopoulos, G. A. Papadopoulos, F. Sikora (Eds.), Proc. SOFSEM 2020, Vol. 12011 of LNCS, Springer, 2020, pp. 174–186. `doi:10.1007/978-3-030-38919-2\_15`.

[2] S. Konstantinidis, N. Moreira, R. Reis, J. Young, Regular expressions and transducers over alphabet-invariant and user-defined labels, in: C. Câmpeanu (Ed.), Proc. 23rd CIAA, Vol. 10977 of LNCS, Springer, 2018, pp. 1–27, Full version in: Int. J. Found. Comput. Sci. 31 (8) (2020) 983–1019. `doi:10.1142/S0129054120420010`.

[3] S. Konstantinidis, N. Moreira, J. Pires, R. Reis, Partial derivatives of regular expressions over alphabet-invariant and user-defined labels, in: M. Hospodár, G. Jirásková (Eds.), Proc. 24th CIAA, Vol. 11601 of LNCS, 2019, pp. 184–196, Full version in: Theor. Comput. Sci. 870(16) (2021) 103–120. `doi:10.1016/j.tcs.2020.12.029`.

[4] A. Demaille, Derived-term automata of multitape expressions with composition, Scien. Annals of Comput. Science 27 (2) (2017) 137–176.

[5] V. M. Antimirov, Partial derivatives of regular expressions and finite automaton constructions., Theoret. Comput. Sci. 155 (2) (1996) 291–319.

[6] S. Lombardy, J. Sakarovitch, Derivatives of rational expressions with multiplicity, Theor. Comput. Sci. 332 (1-3) (2005) 141–177. `doi:10.1016/j.tcs.2004.10.016`.

[7] R. Bastos, S. Broda, A. Machiavelo, N. Moreira, R. Reis, On the average complexity of partial derivative automata for semi-extended expressions, J. Autom. Lang. Comb. 22 (1–3) (2017) 5–28. `doi:10.25596/jalc-2017-005`.

[8] S. Broda, A. Machiavelo, N. Moreira, R. Reis, On average behaviour of regular expressions in strong star normal form, Int. J. Found. Comput. Sci. 30 (6–7) (2019) 899–920. `doi:10.1142/S0129054119400227`.

[9] S. Broda, A. Machiavelo, N. Moreira, R. Reis, Automata for regular expressions with shuffle, Inf. Comput. 259 (2) (2018) 162–173. `doi:10.1016/j.ic.2017.08.013`.

[10] S. Broda, A. Machiavelo, N. Moreira, R. Reis, On the average state complexity of partial derivative automata: an analytic combinatorics approach, Int. J. Found. Comput. Sci. 22 (7) (2011) 1593–1606. `doi:10.1142/S0129054111008908`.

[11] S. Broda, A. Machiavelo, N. Moreira, R. Reis, Analytic combinatorics and descriptional complexity of regular languages on average, ACM SIGACT News 51 (1) (2020) 38–56, SIGACT News Complexity Theory Column 104, Editor, Hemaspaandra, Lane A. `doi:10.1145/3388392.3388400`.

[12] S. Broda, A. Machiavelo, N. Moreira, R. Reis, A hitchhiker's guide to descriptional complexity through analytic combinatorics, Theoret. Comput. Sci. 528 (2014) 85–100. `doi:10.1016/j.tcs.2014.02.013`.

[13] B. G. Mirkin, On the theory of multitape automata, Cybernetics 2 (5) (1968) 9–14. `doi:10.1007/BF01073664`.

[14] C. Nicaud, On the average size of Glushkov's automata, in: A. Dediu, A.-M. Ionescu, C. M. Vide (Eds.), Proc. 3rd LATA, Vol. 5457 of LNCS, 2009, pp. 626–637.

[15] B. G. Mirkin, An algorithm for constructing a base in a language of regular expressions, Eng. Cybernetics 5 (1966) 51—57.

[16] J. Berstel, Transductions and Context-Free Languages, B.G. Teubner, Stuttgart, 1979.

[17] J. Sakarovitch, Elements of Automata Theory, CUP, 2009.

[18] M. O. Rabin, D. Scott, Finite automata and their decision problems, IBM Journal of Research and Development 3 (1959) 114–125.

[19] A. L. Rosenberg, On n-tape finite state acceptors, in: Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design, 1964, p. 6. doi:10.1109/swct.1964.17.

[20] A. Y. Makarevskii, E. D. Stotskaya, Representability in deterministic multi-tape automata, Cybernetics 5 (4) (1969) 390–399. doi:10.1007/BF01073050.

[21] A. Godlevsky, H. A. Grigoryan, T. Grigoryan, S. K. Shoukourian, Some results on regular events for multitape finite automata: A preliminary report, Bull. EATCS 133 (2021).
URL http://eatcs.org/beatcs/index.php/beatcs/article/view/645

[22] R. M. Kaplan, M. Kay, Regular Models of Phonological Rule Systems, Computational Linguistics 20 (3) (1994) 48.

[23] K. R. Beesley, L. Karttunen, Finite State Morphology, Vol. 3, CSLI Publications, 2003.

[24] A. Yli-Jyrä, K. Koskenniemi, Compiling generalized two-level rules and grammars, in: T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala (Eds.), Proc. 5th International Conference on NLP, FinTAL, Vol. 4139 of LNCS, Springer, 2006, pp. 174–185. doi:10.1007/11816508_19.

[25] Helsinki Finite-State Technology, Helsinki finite-state transducer toolkit, http://hfst.github.io (Access date:1.4.2021).

[26] P. Flajolet, R.Sedgewick, Analytic Combinatorics, CUP, 2008.

[27] S. Lang, Algebra, 3rd Edition, Vol. 211 of Grad. Texts Math., Springer, 2001.

[28] E. Hille, Analytic Function Theory, Vol. 2, Blaisdell Publishing Company, 1962.

[29] R. J. Walker, Algebraic Curves, Princeton University Press, 1950.

[30] C. T. C. Wall, Singular Points of Plane Curves, no. 63 in London Mathematical Society Student Texts, Cambridge University Press, 2004.

[31] É. Ghys, A singular mathematical promenade, ENS Éditions, 2017.

[32] Project FAdo, tools for formal languages manipulation, https://pypi.org/project/FAdo/ (Access date:1.1.2021).