

Location Automata for Regular Expressions with Shuffle and Intersection

Sabine Broda ^a, 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*

Abstract





We define the notion of location for regular expressions with shuffle by extending the notion of position in standard regular expressions. Locations allow for the definition of the sets **Follow**, **First**, and **Last** with their usual semantics. From these, we construct an automaton for regular expressions with shuffle (\mathcal{A}_{POS}), which generalises the standard position/Glushkov automaton. The sets mentioned above are also the foundation for other constructions, such as the Follow automaton, and automata based on pointed expressions. As a consequence, all these constructions can now be directly generalised to regular expressions with shuffle, as well as their known relationships. Furthermore, we show that the partial derivative automaton (\mathcal{A}_{PD}) is a right-quotient of the new position automaton, \mathcal{A}_{POS} . In a previous work, an automaton construction based on positions was studied ($\mathcal{A}_{\partial\text{pos}}$), and here we relate \mathcal{A}_{POS} and $\mathcal{A}_{\partial\text{pos}}$. We extend the construction of the prefix automaton \mathcal{A}_{pre} to the shuffle operator and show that it is not a quotient of \mathcal{A}_{POS} . The position automaton has been generalised for regular expressions with the intersection operator. Here we show that locations can be used to define the same automaton. Shuffle and intersection can be seen as two extreme cases of concurrency, namely pure interleaving and strict synchronisation. Locations provide a unified framework that will allow, not only to define position based automata constructions for these two operators, but also for other operators expressing intermediate kinds of concurrency.

Keywords: Regular Expressions, Position Automaton, Locations, Shuffle, Intersection

*This is a completely revised and expanded version of a paper presented 15th International Conference, LATA 2021, Milan, Italy, March 1–5, 2021 [1]

**This work was partially supported by CMUP, which is financed by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the project with reference UIDB/00144/2021.

*Corresponding author

Email addresses: `sabine.broda@fc.up.pt` (Sabine Broda ) ,
`antonio.machiavelo@fc.up.pt` (António Machiavelo ) , `nelma.moreira@fc.up.pt` (Nelma Moreira ) , `rogerio.reis@fc.up.pt` (Rogério Reis )

1. Introduction

Regular expressions with shuffle provide succinct representations for modelling concurrent systems [2, 3] or schema languages [4, 5]. Recently, several automata constructions for expressions with shuffle operators were considered [6, 7, 8]. For the standard interleaving shuffle operator (\sqcup), Broda et al. [6] defined the partial derivative automaton (\mathcal{A}_{PD}) and a position automaton ($\mathcal{A}_{\partial pos}$), showing that \mathcal{A}_{PD} is a right-quotient of $\mathcal{A}_{\partial pos}$. For standard regular expressions there is a one-to-one correspondence between non initial states in the position/Glushkov automaton [9] and occurrences of letters (positions) in the expression. This is no longer true for $\mathcal{A}_{\partial pos}$. Moreover, unlike most constructions of position automata, the definition of $\mathcal{A}_{\partial pos}$ did not rely on the sets **First**, **Last**, and **Follow** [6]. The former two sets characterise the positions of letters that can, respectively, begin or end words of the language; while the latter contains, for each position of a letter, the positions of letters that can follow that position in words of the language. In order to define these sets for expressions containing the shuffle operator, we introduce novel and more complex structures of positions, which we call locations. Locations are defined in such a way that, given an expression with nested shuffles, it allows to specify how far a word has advanced in each of the components (shuffles) of the expression. Each location in **First** corresponds to a position of a letter that can begin a word in the language. The positions that appear in a location in **Last** are the ones that can end a word. In the same way, the members of **Follow** represent pairs of positions of letters in which the second letter follows the first one in some word of the language. From these sets, using locations, the definition of this position automaton \mathcal{A}_{POS} is similar to the usual one: each location is the label of a state, and the incoming transitions of a state are labelled with letters corresponding to positions in that location.

This new construction is presented in Section 3, where an upper bound for the number of states of \mathcal{A}_{POS} in the worst case is given. In Section 4 it is shown that the partial derivative automaton \mathcal{A}_{PD} is a right-quotient of \mathcal{A}_{POS} . A comparison of \mathcal{A}_{POS} and $\mathcal{A}_{\partial pos}$ is considered in Section 5, where their average number of states is discussed. Restricted to expressions without shuffle, both constructions coincide with the standard position automaton. The same holds for \mathcal{A}_{PD} [10]. Thus, the proofs in Section 4 are alternatives to show that, for standard regular expressions, \mathcal{A}_{PD} is a quotient of \mathcal{A}_{POS} . In Section 6 a generalization for the construction of the prefix automaton \mathcal{A}_{Pre} [11, 12] to the shuffle operator is presented, and it is shown that \mathcal{A}_{Pre} is not a quotient of \mathcal{A}_{POS} . Some experimental results comparing the sizes of these constructions are also reported.

The sets **First**, **Last**, and **Follow** are also the base for other constructions, such as the Follow automaton [13], as well as (deterministic) automata based on pointed expressions [14, 15, 16]. As a consequence, it is now straightforward to extend those constructions to expressions with shuffle, solving a problem

stated by Asperti et al. [14]. Moreover, the known relationships between those constructions [16] extend to expressions with shuffle. The resulting taxonomy is presented in Section 7.

Language intersection can model strict synchronisation of concurrent systems. In Section 8 it is shown that locations can be used to define the position automaton for regular expressions with intersection introduced by Broda et al. [17, 1]. In this way, locations allow to obtain a uniform position based automaton construction for regular expressions with both shuffle and intersection operators, and which can be extended to other concurrency operators.

In this paper we revise and present the full proofs of many results that appeared in [1]. Subsection 5.1, Subsection 6.1, and Section 8 are new. Section 7 is substantially expanded and now includes more examples.

2. Preliminaries

The set of standard regular expressions over an alphabet Σ , denoted by RE, contains \emptyset plus all terms generated by the grammar

$$\alpha \rightarrow \varepsilon \mid \sigma \mid (\alpha + \alpha) \mid (\alpha \cdot \alpha) \mid \alpha^* \quad (\sigma \in \Sigma). \quad (1)$$

Note that most of the time the concatenation operator \cdot is omitted. The *language* associated with an expression $\alpha \in \text{RE}$ is denoted by $\mathcal{L}(\alpha)$ and is inductively defined as follows for $\alpha, \beta \in \text{RE}$: $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\varepsilon) = \{\varepsilon\}$, $\mathcal{L}(\sigma) = \{\sigma\}$, $\mathcal{L}(\alpha + \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$, $\mathcal{L}(\alpha \cdot \beta) = \mathcal{L}(\alpha)\mathcal{L}(\beta) = \{wv \mid w \in \mathcal{L}(\alpha) \wedge v \in \mathcal{L}(\beta)\}$, and $\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^* = \bigcup_{n \in \mathbb{N}} (\mathcal{L}(\alpha)^n)$. The empty word is denoted by ε . We define $\varepsilon(\alpha)$ by $\varepsilon(\alpha) = \varepsilon$ if $\varepsilon \in \mathcal{L}(\alpha)$, and $\varepsilon(\alpha) = \emptyset$ otherwise. Given a set of expressions S , the *language* associated with S is $\mathcal{L}(S) = \bigcup_{\alpha \in S} \mathcal{L}(\alpha)$. Moreover, we consider $\varepsilon S = S\varepsilon = S$ and $\emptyset S = S\emptyset = \emptyset$, for any set S of expressions (or other objects). The *alphabetic size* $|\alpha|_\Sigma$ is its number of letters. We denote the subset of Σ containing the symbols that occur in α by Σ_α . A *nondeterministic finite automaton* (NFA) is a quintuple $A = \langle Q, \Sigma, \delta, I, F \rangle$ 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 $\delta : Q \times \Sigma \rightarrow 2^Q$ is the transition function. If $|I| = 1$ and $|\delta(q, \sigma)| \leq 1$, for all $q \in Q, \sigma \in \Sigma$, A is *deterministic* (DFA). The *language* of A is denoted by $\mathcal{L}(A)$ and two automata are *equivalent* if they have the same language. Given an automaton $A = \langle Q, \Sigma, \delta, I, F \rangle$ its *reversal* is $A^R = \langle Q, \Sigma, \delta^R, F, I \rangle$, where $\delta^R(q, \sigma) = \{p \mid q \in \delta(p, \sigma)\}$, and $\mathcal{L}(A^R) = \mathcal{L}(A)^R$, which is the language obtained by reversing the words in $\mathcal{L}(A)$. Two automata $A_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1 \rangle$ and $A_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2 \rangle$ are *isomorphic*, $A_1 \simeq A_2$, if there is a bijection $\varphi : Q_1 \rightarrow Q_2$ such that $\varphi(I_1) = I_2$, $\varphi(F_1) = F_2$, and $\varphi(\delta_1(q_1, \sigma)) = \delta_2(\varphi(q_1), \sigma)$, for all $q_1 \in Q_1, \sigma \in \Sigma$. An equivalence relation \equiv defined on the set of states Q is *right-invariant* w.r.t. A if and only if $\equiv \subseteq (Q \setminus F)^2 \cup F^2$ and if $p \equiv q$, then $\forall \sigma \in \Sigma, p' \in \delta(p, \sigma), \exists q' \in \delta(q, \sigma)$ such that $p' \equiv q'$, for all $p, q \in Q$. If \equiv is a right-invariant relation on Q , the *right-quotient automaton* A/\equiv is given by $A/\equiv = \langle Q/\equiv, \Sigma, \delta/\equiv, I/\equiv, F/\equiv \rangle$, where $\delta/\equiv([p], \sigma) = \{[q] \mid q \in \delta(p, \sigma)\}$. Then, $\mathcal{L}(A/\equiv) = \mathcal{L}(A)$. An equivalence relation on Q is *left-invariant* w.r.t. A if it is right-invariant w.r.t. A^R .

The Position Automaton. Given $\alpha \in \text{RE}$, one can mark each occurrence of a letter σ with its position in α , reading it from left to right. The resulting regular expression is a *marked* regular expression $\bar{\alpha}$ with all letters occurring only once (linear) and belonging to $\Sigma_{\bar{\alpha}}$. Each *position* $i \in [1, |\alpha|_{\Sigma}]$ corresponds to the symbol σ_i in $\bar{\alpha}$, and thus to exactly one occurrence of σ in α . For instance, if $\alpha = a(bb + aba)^*b$, then $\bar{\alpha} = a_1(b_2b_3 + a_4b_5a_6)^*b_7$. The same notation is used for unmarking, $\bar{\bar{\alpha}} = \alpha$. Let $\text{Pos}(\alpha) = \{1, 2, \dots, |\alpha|_{\Sigma}\}$, and $\text{Pos}_0(\alpha) = \text{Pos}(\alpha) \cup \{0\}$. Positions were used by Glushkov [9] to define an NFA equivalent to α , usually called the *position* or *Glushkov automaton*, $\mathcal{A}_{\text{POS}}(\alpha)$. Each state of the automaton, except for the initial one, corresponds to a position, and there exists a transition from i to j by σ such that $\bar{\sigma}_j = \sigma$, if σ_i can be followed by σ_j in some word represented by $\bar{\alpha}$. The sets of positions that are used to define the position automaton for a given α and $i \in \text{Pos}(\alpha)$, are

$$\begin{aligned} \text{First}(\bar{\alpha}) &= \{i \mid (\exists w \in \Sigma_{\bar{\alpha}}^*) (\sigma_i w \in \mathcal{L}(\bar{\alpha}))\}, \\ \text{Last}(\bar{\alpha}) &= \{i \mid (\exists w \in \Sigma_{\bar{\alpha}}^*) (w \sigma_i \in \mathcal{L}(\bar{\alpha}))\}, \\ \text{Follow}(\bar{\alpha}, i) &= \{j \mid (\exists u, v \in \Sigma_{\bar{\alpha}}^*) (u \sigma_i \sigma_j v \in \mathcal{L}(\bar{\alpha}))\}. \end{aligned}$$

For the sake of readability, whenever an expression α is not marked, we take $f(\alpha) = f(\bar{\alpha})$, for any function $f \in \{\text{First}, \text{Follow}, \text{Last}, \text{Pos}\}$, as well as for other functions which we will define later (e.g. Loc , p2loc), that have marked expressions as arguments. We define the position automaton using the approach in Broda et. al [16], where the transition function is expressed as the composition of functions Select and Follow . Given a letter σ and a set of positions S , the function Select selects the subset of positions in S that correspond to letter σ . Formally, given $S \subseteq \text{Pos}(\alpha)$ and $\sigma \in \Sigma$, let

$$\text{Select}(S, \sigma) = \{i \mid i \in S \wedge \bar{\sigma}_i = \sigma\}.$$

Then, the *position automaton* for α is

$$\mathcal{A}_{\text{POS}}(\alpha) = \langle \text{Pos}_0(\alpha), \Sigma, \delta_{\text{POS}}, 0, \text{Last}_0(\alpha) \rangle,$$

where $\text{Last}_0(\alpha) = \text{Last}(\alpha) \cup \varepsilon(\alpha)\{0\}$ and $\delta_{\text{POS}}(i, \sigma) = \text{Select}(\text{Follow}(\alpha, i), \sigma)$, for $i \in \text{Pos}_0(\alpha)$ and $\sigma \in \Sigma$.

Regular Expressions with Shuffle. Given an alphabet Σ , the shuffle of two words in Σ^* is the finite set of words defined inductively as follows: $x \sqcup \varepsilon = \varepsilon \sqcup x = \{x\}$ and $\sigma x \sqcup \tau y = \{\sigma z \mid z \in x \sqcup \tau y\} \cup \{\tau z \mid z \in \sigma x \sqcup y\}$, for $x, y \in \Sigma^*$, and $\sigma, \tau \in \Sigma$. This definition is extended to languages in the natural way by $L_1 \sqcup L_2 = \bigcup_{x \in L_1, y \in L_2} x \sqcup y$. It is well known that \sqcup is a regular operator. One can, hence, extend regular expressions to include the \sqcup operator. The set of regular expressions with shuffle over Σ , $\mathcal{R}(\sqcup)$, contains all the expressions of RE generated by the grammar rules in (1) plus rule $\alpha \rightarrow (\alpha \sqcup \alpha)$. The language represented by an expression $\alpha \sqcup \beta$ is $\mathcal{L}(\alpha \sqcup \beta) = \mathcal{L}(\alpha) \sqcup \mathcal{L}(\beta)$.

3. A Location Based Position Automaton

In this section we define a new construction for a position automaton for expressions with shuffle, which is based on the sets **First**, **Last**, and **Follow**. In order to define those sets for expressions containing the shuffle operator, we need to consider more complex structures, which we call locations. Locations are defined in such a way that, given an expression with nested shuffles, it allows to specify how far a word has advanced in each of the components (shuffles) of this expression. More precisely, when we enter a shuffle, we need to know not only one position, but two, since we need to know where we are in the two subwords that are actually shuffled right now. Due to nesting of shuffles, this means that we have to store a tree of positions, which is illustrated in the following example.

Example 1. Consider $\alpha = (a^*b \sqcup cd)^* \sqcup (ac)^*$ and $\bar{\alpha} = (a_1^*b_2 \sqcup c_3d_4)^* \sqcup (a_5c_6)^*$. We have $a_1a_1b_2 \in \mathcal{L}(a_1^*b_2)$, $c_3d_4 \in \mathcal{L}(c_3d_4)$ and consequently $a_1a_1c_3d_4b_2 \in \mathcal{L}((a_1^*b_2 \sqcup c_3d_4)^*)$. Since $a_5c_6 \in \mathcal{L}((a_5c_6)^*)$, we conclude that $w = a_1a_5a_1c_3d_4c_6b_2 \in (a_1a_1c_3d_4b_2 \sqcup a_5c_6) \subseteq \mathcal{L}(\alpha)$. When processing w in an automaton, and after reading the prefix $a_1a_5a_1c_3d_4$, one has to know that in the different shuffle components the last letters read are respectively a_1 , d_4 , and a_5 . This information will be stored in the location $((1,4),5)$. On the other hand, reading the prefix a_1a_5 should lead to the location $((1,0),5)$, where 0 indicates that the right side of the first shuffle has not been entered yet.

Formally, given $\alpha \in \mathcal{R}(\sqcup)$, the set of locations $\text{Loc}(\alpha) = \text{Loc}(\bar{\alpha})$, is inductively defined on the structure of the expression $\bar{\alpha}$ as follows.

$$\begin{aligned} \text{Loc}(\varepsilon) &= \emptyset, \text{Loc}(\sigma_i) = \{i\}, \text{Loc}(\alpha^*) = \text{Loc}(\alpha), \\ \text{Loc}(\alpha_1 + \alpha_2) &= \text{Loc}(\alpha_1\alpha_2) = \text{Loc}(\alpha_1) \cup \text{Loc}(\alpha_2), \\ \text{Loc}(\alpha_1 \sqcup \alpha_2) &= \text{Loc}(\alpha_1) \times \text{Loc}(\alpha_2) \cup \text{Loc}(\alpha_1) \times \{0\} \cup \{0\} \times \text{Loc}(\alpha_2). \end{aligned} \quad (2)$$

Note that each location p in α is either a position $i \in \text{Pos}(\alpha)$, or of the form $(0, p_2)$, $(p_1, 0)$, or (p_1, p_2) , where p_1, p_2 are also locations in α . As such, each location corresponds to a complete binary tree.

The set of positions in a location p , $\text{l2pos}(p)$, is defined inductively by

$$\begin{aligned} \text{l2pos}(i) &= \{i\}, \\ \text{l2pos}((0, p)) &= \text{l2pos}((p, 0)) = \text{l2pos}(p), \\ \text{l2pos}((p_1, p_2)) &= \text{l2pos}(p_1) \cup \text{l2pos}(p_2). \end{aligned}$$

Note that for $p \in \text{Loc}(\alpha_1 \circ \alpha_2)$, where $\circ \in \{+, \cdot\}$, one has either $\text{l2pos}(p) \subseteq \text{Pos}(\alpha_1)$ or $\text{l2pos}(p) \subseteq \text{Pos}(\alpha_2)$.

Example 2. For $\alpha = (a^*b \sqcup cd)^* \sqcup (ac)^*$ and $\bar{\alpha} = (a_1^*b_2 \sqcup c_3d_4)^* \sqcup (a_5c_6)^*$,

$$\begin{aligned} \text{Loc}((a^*b \sqcup cd)^*) &= \{(1, 0), (2, 0), (0, 3), (0, 4), (1, 3), (1, 4), (2, 3), (2, 4)\} \\ \text{Loc}((ac)^*) &= \{5, 6\} \\ \text{Loc}(\alpha) &= \{((0, 3), n), ((0, 4), n), ((1, 0), n), ((2, 0), n), (0, 5), (0, 6), \\ &\quad ((1, 3), n), ((2, 3), n), ((1, 4), n), ((2, 4), n) \mid n = 0, 5, 6\}, \end{aligned}$$

with $\text{l2pos}(((2,3),0)) = \{2,3\}$, and $\text{l2pos}(((2,3),5)) = \{2,3,5\}$. For instance, the location $((2,3),5)$ corresponds to words for which the last letters read in the subexpressions a^*b , cd , and $(ac)^*$, are respectively b , c , and a . This example also illustrates that the locations of an expression often contain elements of different signature. In this case we have for instance $(0,5), ((2,3),6) \in \text{Loc}(\alpha)$. Furthermore, considering the expression αa we have $\text{Loc}(\alpha a) = \text{Loc}(\alpha) \cup \{7\}$, which contains the locations 7, $(0,5)$ and $((2,3),6)$.

In the following, we show that the function $\text{l2pos} : \text{Loc}(\alpha) \rightarrow 2^{\text{Pos}(\alpha)}$ is injective. First note that, for $I \in \text{l2pos}(\text{Loc}(\alpha)) = \{\text{l2pos}(p) \mid p \in \text{Loc}(\alpha)\}$ there exists a unique location p such that $I = \text{l2pos}(p)$. Furthermore, consider $\text{p2loc}(\alpha, I) = \text{p2loc}(\bar{\alpha}, I)$ defined by the rules below, where $\circ \in \{+, \cdot\}$.

$$\begin{aligned} \text{p2loc}(\sigma_i, \{i\}) &= i, \\ \text{p2loc}(\alpha_1 \circ \alpha_2, I) &= \begin{cases} \text{p2loc}(\alpha_1, I), & \text{if } I \subseteq \text{Pos}(\alpha_1), \\ \text{p2loc}(\alpha_2, I), & \text{if } I \subseteq \text{Pos}(\alpha_2), \end{cases} \\ \text{p2loc}(\alpha^*, I) &= \text{p2loc}(\alpha, I), \\ \text{p2loc}(\alpha_1 \sqcup \alpha_2, I) &= \begin{cases} (\text{p2loc}(\alpha_1, I), 0), & \text{if } I \subseteq \text{Pos}(\alpha_1), \\ (0, \text{p2loc}(\alpha_2, I)), & \text{if } I \subseteq \text{Pos}(\alpha_2), \\ (\text{p2loc}(\alpha_1, I_1), \text{p2loc}(\alpha_2, I_2)), & \text{if } I = I_1 \cup I_2, \\ & \emptyset \neq I_j \subseteq \text{Pos}(\alpha_j), j = 1, 2. \end{cases} \end{aligned}$$

Then, we have the following result.

Lemma 1. *Given $p \in \text{Loc}(\alpha)$, one has $\text{p2loc}(\alpha, \text{l2pos}(p)) = p$.*

Proof. By induction on the structure of α . □

As a consequence of the previous lemma it follows that the function l2pos is injective. The following proposition gives an upper bound on the size of $\text{Loc}(\alpha)$, and the next example exhibits an expression for which this upper bound is reached.

Proposition 2. *Given $\alpha \in \mathcal{R}(\sqcup)$, one has $|\text{Loc}(\alpha)| \leq 2^{|\alpha|_\Sigma} - 1$.*

Proof. It follows from Lemma 1, and in particular from the injectivity of l2pos , that the number of locations is less or equal to the number of non-empty subsets of $\text{Pos}(\alpha)$, which is precisely $2^{|\alpha|_\Sigma} - 1$. □

Example 3. *Consider $\alpha_n = a_1 \sqcup \dots \sqcup a_n$, where $n \geq 1$, $a_i \neq a_j$ for $1 \leq i \neq j \leq n$. Then, $\text{l2pos}(\text{Loc}(\alpha_n)) = 2^{\text{Pos}(\alpha_n)} \setminus \{\emptyset\}$, which is of size $2^n - 1$.*

Lemma 3. *Given $\alpha \in \mathcal{R}(\sqcup)$ and $i \in \text{Pos}(\alpha)$, the following hold:*

1. *there is $p \in \text{Loc}(\alpha)$ with $i \in \text{l2pos}(p)$;*
2. *there are words $w, w' \in \Sigma_\alpha^*$, such that $w\sigma_i w' \in \mathcal{L}(\bar{\alpha})$.*

Proof. Straightforward by structural induction on α . \square

Given $\alpha \in \mathcal{R}(\sqcup)$, the states in the position automaton will be labelled by the elements in $\text{Loc}(\alpha)$, except for the initial state labelled by 0.

The sets **First**, **Last** and **Follow** are defined extending the usual definitions, [18, 13], to the shuffle operator. For expressions without shuffle, each position i corresponds exactly to one marked letter σ_i and, consequently, in the Glushkov automaton all incoming transitions of state i are labelled by $\sigma = \overline{\sigma_i}$. This is no longer true for expressions with shuffle. In this case a location p labelling a state can have incoming transitions labelled by different letters, corresponding to the positions in $\text{l2pos}(p)$ and depending on the source state. For this reason we will include letters in the definition of **First** and **Follow**. Recall that given a set S and an expression α , $\varepsilon(\alpha)S = S$ if $\varepsilon(\alpha) = \varepsilon$, and $\varepsilon(\alpha)S = \emptyset$ otherwise.

Given $\alpha \in \mathcal{R}(\sqcup)$ the set $\text{First}(\alpha) \subseteq \Sigma_\alpha \times \text{Loc}(\alpha)$ is defined as follows.

$$\begin{aligned} \text{First}(\varepsilon) &= \emptyset, \\ \text{First}(\sigma_i) &= \{(\overline{\sigma_i}, i)\}, \\ \text{First}(\alpha_1 + \alpha_2) &= \text{First}(\alpha_1) \cup \text{First}(\alpha_2), \\ \text{First}(\alpha_1 \alpha_2) &= \text{First}(\alpha_1) \cup \varepsilon(\alpha_1) \text{First}(\alpha_2), \\ \text{First}(\alpha^*) &= \text{First}(\alpha), \\ \text{First}(\alpha_1 \sqcup \alpha_2) &= \{(\sigma, (p, 0)) \mid (\sigma, p) \in \text{First}(\alpha_1)\} \cup \{(\sigma, (0, p)) \mid (\sigma, p) \in \text{First}(\alpha_2)\}. \end{aligned} \tag{3}$$

Fact 1. For every $(\sigma, p) \in \text{First}(\alpha)$ the location p contains exactly one non-null component $i \in \text{Pos}(\alpha)$, i.e., $\text{l2pos}(p) = \{i\}$. Furthermore, $\overline{\sigma_i} = \sigma$.

Lemma 4. Given $\alpha \in \mathcal{R}(\sqcup)$, one has $(\sigma, p) \in \text{First}(\alpha)$ with $\text{l2pos}(p) = \{i\}$, if and only if there is some $w \in \Sigma_{\overline{\alpha}}^*$, such that $\sigma_i w \in \mathcal{L}(\overline{\alpha})$ and $\overline{\sigma_i} = \sigma$.

Proof. The proof is by structural induction on the marked expression $\overline{\alpha}$. For ε and marked singletons the result is obvious. For union, concatenation and Kleene star, the proof is similar to the one for standard expressions. Consider the case of an expression $\alpha_1 \sqcup \alpha_2$.

Let $(\sigma, (p, 0))$, with $(\sigma, p) \in \text{First}(\alpha_1)$, $\text{l2pos}((p, 0)) = \text{l2pos}(p) = \{i\}$, and $\overline{\sigma_i} = \sigma$. By the induction hypothesis, there is some $w \in \Sigma_{\overline{\alpha}}^*$, such that $\sigma_i w \in \mathcal{L}(\alpha_1)$. Consider any word $w' \in \mathcal{L}(\alpha_2) \neq \emptyset$. Then, $\sigma_i w w' \in \mathcal{L}(\alpha_1) \sqcup \mathcal{L}(\alpha_2) = \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. The case of $(\sigma, (0, p))$, with $(\sigma, p) \in \text{First}(\alpha_2)$, is analogous. For the other direction, consider a word $\sigma_i w \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. By definition, either there is some $\sigma_i w_1 \in \mathcal{L}(\alpha_1)$ and some $w_2 \in \mathcal{L}(\alpha_2)$ such that $w \in w_1 \sqcup w_2$, or there is some $w_1 \in \mathcal{L}(\alpha_1)$ and some $\sigma_i w_2 \in \mathcal{L}(\alpha_2)$ such that $w \in w_1 \sqcup w_2$. In the first case, by induction, there exists $(\sigma, p) \in \text{First}(\alpha_1)$ with $\text{l2pos}(p) = \{i\}$ and $\overline{\sigma_i} = \sigma$. Consequently, $(\sigma, (p, 0)) \in \text{First}(\alpha_1 \sqcup \alpha_2)$. The other case is similar. \square

As usual, the equations for **Last** are the same as for **First**, except for concatenation and shuffle. Note that for **Last** we do not need the letter, which is therefore omitted in the definition.

$$\begin{aligned}
\text{Last}(\sigma_i) &= \{i\}, \\
\text{Last}(\alpha_1 \alpha_2) &= \text{Last}(\alpha_2) \cup \varepsilon(\alpha_2) \text{Last}(\alpha_1), \\
\text{Last}(\alpha_1 \sqcup \alpha_2) &= \text{Last}(\alpha_1) \times \text{Last}(\alpha_2) \\
&\quad \cup \varepsilon(\alpha_1)(\{0\} \times \text{Last}(\alpha_2)) \cup \varepsilon(\alpha_2)(\text{Last}(\alpha_1) \times \{0\}).
\end{aligned} \tag{4}$$

Lemma 5. *Given $\alpha \in \mathcal{R}(\sqcup)$ and $i \in \text{Pos}(\alpha)$, there is a location $p \in \text{Last}(\alpha)$ with $i \in \text{l2pos}(p)$ if and only if there is some $w \in \Sigma_{\bar{\alpha}}^*$, such that $w\sigma_i \in \mathcal{L}(\bar{\alpha})$.*

Proof. The proof is by structural induction on $\bar{\alpha}$. We need only to consider the case of an expression $\alpha_1 \sqcup \alpha_2$. Let $(p_1, p_2) \in \text{Last}(\alpha_1) \times \text{Last}(\alpha_2)$ and $i \in \text{l2pos}(p_1)$. By the induction hypothesis, there is some $w_1 \in \Sigma_{\alpha_1}^*$, such that $w_1\sigma_i \in \mathcal{L}(\alpha_1)$. For any $w_2 \in \mathcal{L}(\alpha_2) \neq \emptyset$, $w_2 w_1 \sigma_i \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. Next, consider a location $(0, p) \in \varepsilon(\alpha_1)(\{0\} \times \text{Last}(\alpha_2))$ and $i \in \text{l2pos}((0, p)) = \text{l2pos}(p)$. By induction, there is some $w_2 \in \Sigma_{\alpha_2}^*$, such that $w_2\sigma_i \in \mathcal{L}(\alpha_2)$. On the other hand $\varepsilon \in \mathcal{L}(\alpha_1)$. Thus, $w_2\sigma_i \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. The remaining cases are analogous.

For the other direction, consider $w\sigma_i \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. By definition, there is some $w_1\sigma_i \in \mathcal{L}(\alpha_1)$ and some $w_2 \in \mathcal{L}(\alpha_2)$ such that $w \in w_1 \sqcup w_2$ (or vice-versa). By induction, there exists a location $p_1 \in \text{Last}(\alpha_1)$ with $i \in \text{l2pos}(p_1)$. If $w_2 = w'_2\sigma_j$, by induction there is some $p_2 \in \text{Last}(\alpha_2)$ with $j \in \text{l2pos}(p_2)$. Thus, $(p_1, p_2) \in \text{Last}(\alpha_1) \times \text{Last}(\alpha_2)$ and $i \in \text{l2pos}(p_1) \subseteq \text{l2pos}((p_1, p_2))$. If $w_2 = \varepsilon$, then $(p_1, 0) \in \varepsilon(\alpha_2)(\text{Last}(\alpha_1) \times \{0\})$ and $i \in \text{l2pos}(p_1) = \text{l2pos}((p_1, 0))$. \square

Given $\alpha \in \mathcal{R}(\sqcup)$, we define $\text{Loc}_0(\alpha) = \text{Loc}(\alpha) \cup \{0\}$ and $\text{Last}_0(\alpha) = \text{Last}(\alpha) \cup \varepsilon(\alpha)\{0\}$. Finally, we define $\text{Follow}(\alpha, p) \subseteq \Sigma_{\alpha} \times \text{Loc}(\alpha)$ by setting $\text{Follow}(\alpha, 0) = \text{First}(\alpha)$, and for $p \in \text{Loc}(\alpha)$,

$$\begin{aligned}
\text{Follow}(\varepsilon, p) &= \text{Follow}(\sigma_i, p) = \emptyset, \\
\text{Follow}(\alpha_1 + \alpha_2, p) &= \begin{cases} \text{Follow}(\alpha_1, p), & \text{if } p \in \text{Loc}(\alpha_1), \\ \text{Follow}(\alpha_2, p), & \text{if } p \in \text{Loc}(\alpha_2), \end{cases} \\
\text{Follow}(\alpha_1 \alpha_2, p) &= \begin{cases} \text{Follow}(\alpha_1, p), & \text{if } p \notin \text{Last}(\alpha_1), \\ \text{Follow}(\alpha_1, p) \cup \text{First}(\alpha_2), & \text{if } p \in \text{Last}(\alpha_1), \\ \text{Follow}(\alpha_2, p), & \text{if } p \in \text{Loc}(\alpha_2), \end{cases} \tag{5} \\
\text{Follow}(\alpha_1^*, p) &= \begin{cases} \text{Follow}(\alpha_1, p), & \text{if } p \notin \text{Last}(\alpha_1), \\ \text{Follow}(\alpha_1, p) \cup \text{First}(\alpha_1), & \text{otherwise,} \end{cases} \\
\text{Follow}(\alpha_1 \sqcup \alpha_2, p) &= \{ (\sigma, (p'_1, p_2)) \mid (\sigma, p'_1) \in \text{Follow}(\alpha_1, p_1) \} \\
&\quad \cup \{ (\sigma, (p_1, p'_2)) \mid (\sigma, p'_2) \in \text{Follow}(\alpha_2, p_2) \} \\
&\quad \text{if } p = (p_1, p_2).
\end{aligned}$$

Furthermore, given $S \in 2^{\text{Loc}_0(\alpha)}$ set $\text{Follow}(\alpha, S) = \bigcup_{p \in S} \text{Follow}(\alpha, p)$. The following example shows why letters are necessary in the definition of Follow .

Example 4. For $\alpha = a^* \sqcup b^*$ and $\bar{\alpha} = a_1^* \sqcup b_2^*$, $\text{Last}(\alpha) = \{(1, 0), (0, 2), (1, 2)\}$, and

$$\begin{aligned}\text{Follow}(\alpha, 0) &= \text{First}(\alpha) = \{(a, (1, 0)), (b, (0, 2))\}, \\ \text{Follow}(\alpha, (1, 0)) &= \{(a, (1, 0)), (b, (1, 2))\}, \\ \text{Follow}(\alpha, (0, 2)) &= \{(a, (1, 2)), (b, (0, 2))\}, \\ \text{Follow}(\alpha, (1, 2)) &= \{(a, (1, 2)), (b, (1, 2))\}.\end{aligned}$$

Lemma 6. Given $\alpha \in \mathcal{R}(\sqcup)$ the following hold.

1. If there are locations $p, q \in \text{Loc}(\alpha)$ with $(\sigma, q) \in \text{Follow}(\alpha, p)$, then there are $w, w' \in \Sigma_{\bar{\alpha}}^*$ and $i, j \in \text{Pos}(\alpha)$, such that $w\sigma_i\sigma_jw' \in \mathcal{L}(\bar{\alpha})$ with $i \in \text{l2pos}(p)$, $j \in \text{l2pos}(q)$, and $\sigma = \bar{\sigma}_j$.
2. If there are $w, w' \in \Sigma_{\bar{\alpha}}^*$ and $i, j \in \text{Pos}(\alpha)$ such that $w\sigma_i\sigma_jw' \in \mathcal{L}(\bar{\alpha})$, then there are $p, q \in \text{Loc}(\alpha)$ with $i \in \text{l2pos}(p)$ and $j \in \text{l2pos}(q)$, such that $(\bar{\sigma}_j, q) \in \text{Follow}(\alpha, p)$.

Proof. The proof is by structural induction on the marked expression $\bar{\alpha}$. We consider the cases of concatenation and shuffle. For 1. consider $(\sigma, q) \in \text{Follow}(\alpha_1\alpha_2, p)$. Then $(\sigma, q) \in \text{Follow}(\alpha_1, p)$, $(\sigma, q) \in \text{Follow}(\alpha_2, p)$, or $(\sigma, q) \in \text{First}(\alpha_2)$ and $p \in \text{Last}(\alpha_1)$. The first two cases follow from the induction hypothesis. For the last case, we have by Lemmas 4 and 5, that there are $w, w' \in \Sigma_{\bar{\alpha}}^*$, $i \in \text{l2pos}(p)$, $j \in \text{l2pos}(q)$ with $\sigma = \bar{\sigma}_j$, such that $w\sigma_i \in \mathcal{L}(\alpha_1)$ and $\sigma_jw' \in \mathcal{L}(\alpha_2)$. We conclude that $w\sigma_i\sigma_jw' \in \mathcal{L}(\alpha_1\alpha_2)$. For 2. we write $w\sigma_i\sigma_jw' = w_1w_2\sigma_i\sigma_jw'_1w'_2$ and distinguish between three cases, where respectively,

- $w_1w_2\sigma_i\sigma_jw'_1 \in \mathcal{L}(\alpha_1)$ and $w'_2 \in \mathcal{L}(\alpha_2)$,
- $w_1w_2\sigma_i \in \mathcal{L}(\alpha_1)$ and $\sigma_jw'_1w'_2 \in \mathcal{L}(\alpha_2)$, or
- $w_1 \in \mathcal{L}(\alpha_1)$ and $w_2\sigma_i\sigma_jw'_1w'_2 \in \mathcal{L}(\alpha_2)$.

In the first and the last case the result follows from the induction hypothesis. For the remaining case, we know by Lemma 5, that there is $p \in \text{Last}(\alpha_1)$ and $i \in \text{l2pos}(p)$. On the other hand it follows from Lemma 4 and Fact 1 that there is $(\sigma, q) \in \text{First}(\alpha_2)$ with $\text{l2pos}(q) = \{j\}$ and $\bar{\sigma}_j = \sigma$. Now, it is sufficient to apply the definition of Follow.

Now, for 1. let $(\sigma, q) \in \text{Follow}(\alpha_1 \sqcup \alpha_2, (p_1, p_2))$, where $q = (p'_1, p_2)$ with $(\sigma, p'_1) \in \text{Follow}(\alpha_1, p_1)$ (or $q = (p_1, p'_2)$ with $(\sigma, p'_2) \in \text{Follow}(\alpha_2, p_2)$).

From $(\sigma, p'_1) \in \text{Follow}(\alpha_1, p_1)$ we consider two cases. If $p_1 \neq 0$, then by induction there exists $i \in \text{l2pos}(p_1)$ and $j \in \text{l2pos}(p'_1)$ with $\sigma = \bar{\sigma}_j$ such that $w\sigma_i\sigma_jw' \in \mathcal{L}(\alpha_1)$. Now, consider any word $w_2 \in \mathcal{L}(\alpha_2)$. Then, $w\sigma_i\sigma_jw'_1w'_2 \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. If $p_1 = 0$, then $(\sigma, p'_1) \in \text{First}(\alpha_1)$ and $\text{l2pos}(p'_1) = \{j\}$ for some $j \in \text{Pos}(\alpha_1)$. Furthermore, there is a word w_1 such that $\sigma_jw_1 \in \mathcal{L}(\alpha_1)$, by Lemma 4. Moreover, $0 \neq p_2 \in \text{Loc}(\alpha_2)$. Now, consider any $i \in \text{l2pos}(p_2) = \text{l2pos}((0, p_2))$. It follows from Lemma 3, that there are words w_2 and w'_2 such that $w_2\sigma_iw'_2 \in \mathcal{L}(\alpha_2)$. Consequently, $w_2\sigma_i\sigma_jw_1w'_2 \in \sigma_jw_1 \sqcup w_2\sigma_iw'_2 \subseteq \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. If $q =$

(p_1, p'_2) the proof is analogous. For 2. we consider two cases (the remaining cases are variants of these two).

First, let $x\sigma_i\sigma_jy \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$, for some $w_1\sigma_iw'_1 \in \mathcal{L}(\alpha_1)$ and $w_2\sigma_jw'_2 \in \mathcal{L}(\alpha_2)$, where $x \in w_1 \sqcup w_2$ and $y \in w'_1 \sqcup w'_2$. By Lemma 3 there is $p_1 \in \text{Loc}(\alpha_1)$ with $i \in \text{l2pos}(p_1)$. We consider two cases for w_2 .

If $w_2 = w''_2\sigma_k$, i.e., $w''_2\sigma_k\sigma_jw'_2 \in \mathcal{L}(\alpha_2)$, then it follows by induction that there are locations $p_2, p'_2 \in \text{Loc}(\alpha_2)$, such that $k \in \text{l2pos}(p_2)$, $j \in \text{l2pos}(p'_2)$, $\sigma = \overline{\sigma_j}$, and $(\sigma, p'_2) \in \text{Follow}(\alpha_2, p_2)$. Moreover, $i \in \text{l2pos}(p_1) \subseteq \text{l2pos}((p_1, p_2))$, and by the definition of the Follow set, $(\sigma, (p_1, p'_2)) \in \text{Follow}(\alpha_1 \sqcup \alpha_2, (p_1, p_2))$.

If $w_2 = \varepsilon$, then there exists a location $p_2 \in \text{Loc}(\alpha_2)$ such that $\text{l2pos}(p_2) = \{j\}$, $(\sigma, p_2) \in \text{First}(\alpha_2) = \text{Follow}(\alpha_2, 0)$, and $\sigma = \overline{\sigma_j}$. We conclude that $(\sigma, (p_1, p_2)) \in \text{Follow}(\alpha_1 \sqcup \alpha_2, (p_1, 0))$, while $i \in \text{l2pos}(p_1) \subseteq \text{l2pos}((p_1, 0))$.

Second, let $x\sigma_i\sigma_jy \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$, for some $w_1\sigma_i\sigma_jw'_1 \in \mathcal{L}(\alpha_1)$ and $w_2w'_2 \in \mathcal{L}(\alpha_2)$, where $x \in w_1 \sqcup w_2$ and $y \in w'_1 \sqcup w'_2$. It follows by induction that there are locations $p_1, p'_1 \in \text{Loc}(\alpha_1)$, such that $i \in \text{l2pos}(p_1)$, $j \in \text{l2pos}(p'_1)$, $\sigma = \overline{\sigma_j}$, and $(\sigma, p'_1) \in \text{Follow}(\alpha_1, p_1)$. Consider any location $p_2 \in \text{Loc}_0(\alpha_2)$. Then, $(p_1, p_2), (p'_1, p_2) \in \text{Loc}(\alpha_1 \sqcup \alpha_2)$. Furthermore, $i \in \text{l2pos}((p_1, p_2))$ and $(\sigma, (p'_1, p_2)) \in \text{Follow}(\alpha_1 \sqcup \alpha_2, (p_1, p_2))$. \square

For a set $S \subseteq \Sigma_\alpha \times \text{Loc}(\alpha)$ and $\sigma \in \Sigma$, let $\text{Select}(S, \sigma) = \{p \mid (\sigma, p) \in S\}$. The *position automaton* for $\alpha \in \mathcal{R}(\sqcup)$ is

$$\mathcal{A}_{\text{POS}}(\alpha) = \langle \text{Loc}_0(\alpha), \Sigma, \delta_{\text{POS}}, 0, \text{Last}_0(\alpha) \rangle,$$

where $\delta_{\text{POS}}(p, \sigma) = \text{Select}(\text{Follow}(\alpha, p), \sigma)$, for $p \in \text{Loc}_0(\alpha), \sigma \in \Sigma$.

The correctness of this construction follows from the previous four lemmas.

Proposition 7. $\mathcal{L}(\mathcal{A}_{\text{POS}}(\alpha)) = \mathcal{L}(\alpha)$.

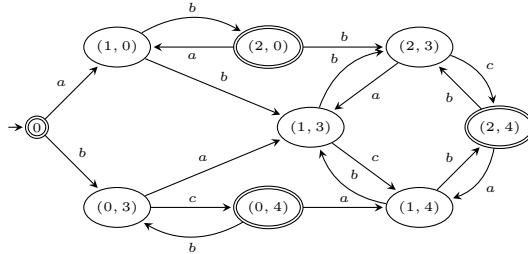
Example 5. For $\alpha = (ab)^* \sqcup (bc)^*$ with $\overline{\alpha} = (a_1b_2)^* \sqcup (b_3c_4)^*$,

$$\text{Loc}_0(\alpha) = \{0, (0, 3), (0, 4), (1, 0), (2, 0), (1, 3), (1, 4), (2, 3), (2, 4)\},$$

$$\text{First}(\alpha) = \{(a, (1, 0)), (b, (0, 3))\},$$

$$\text{Last}_0(\alpha) = \{0, (0, 4), (2, 0), (2, 4)\}.$$

The position automaton $\mathcal{A}_{\text{POS}}(\alpha)$ is depicted below.



Note that, for an expression α without shuffle we have $\text{Loc}(\alpha) = \text{Pos}(\alpha)$ and $\mathcal{A}_{\text{POS}}(\alpha)$ is exactly the standard position automaton. In fact, the usual *Follow* set for a position j is equal to $\{i \mid (\overline{\sigma}_i, i) \in \text{Follow}(\alpha, j)\}$.

4. $\mathcal{A}_{PD}(\alpha)$ as a Quotient of $\mathcal{A}_{POS}(\alpha)$

In this section we show that the partial derivative automaton $\mathcal{A}_{PD}(\alpha)$ for expressions $\alpha \in \mathcal{R}(\sqcup)$ [6] is a quotient of the position automaton as defined in the previous section. This generalises a well known result for regular expressions to expressions in $\mathcal{R}(\sqcup)$. We give some intuition on this, considering as an example the regular expression $\alpha = (a+b)c$ with $\bar{\alpha} = (a_1+b_2)c_3$. Besides of the initial state 0, the position automaton $\mathcal{A}_{POS}(\alpha)$ will have three further states, labelled respectively with 1, 2, and 3. These *contain* essentially the information that the last letter in an prefix leading to them are: a in the case of 1, b in the case of 2, and c in the case of 3. The information *contained* in a state of the partial derivative automaton $\mathcal{A}_{PD}(\alpha)$ is of different nature. In fact, each state is labelled by a regular expression (a partial derivative). This expression defines the set of words that complement (as a suffix) any prefix, that leads to this state, to a word of $\mathcal{L}(\alpha)$. In our example prefixes a and b have the same set of possible complements, namely $\{c\}$. Hence, in this particular case, $\mathcal{A}_{PD}(\alpha)$ can be obtained from $\mathcal{A}_{POS}(\alpha)$ by merging the two states labelled with 1 and with 2 into a state labelled with c .

We recall that the definition of the *set of partial derivatives of $\alpha \in \mathcal{R}(\sqcup)$ w.r.t. a letter $\sigma \in \Sigma$* , denoted by $\partial_\sigma(\alpha)$, is a set of expressions inductively defined by

$$\begin{aligned} \partial_\sigma(\emptyset) &= \partial_\sigma(\varepsilon) = \emptyset, & \partial_\sigma(\alpha + \beta) &= \partial_\sigma(\alpha) \cup \partial_\sigma(\beta), \\ \partial_\sigma(\sigma') &= \begin{cases} \{\varepsilon\} & \text{if } \sigma = \sigma', \\ \emptyset & \text{otherwise,} \end{cases} & \partial_\sigma(\alpha\beta) &= \partial_\sigma(\alpha)\beta \cup \varepsilon(\alpha)\partial_\sigma(\beta), \\ \partial_\sigma(\alpha^*) &= \partial_\sigma(\alpha)\alpha^*, & \partial_\sigma(\alpha \sqcup \beta) &= \partial_\sigma(\alpha) \sqcup \{\beta\} \cup \{\alpha\} \sqcup \partial_\sigma(\beta). \end{aligned}$$

where, for any $S, T \subseteq \mathcal{R}(\sqcup) \setminus \{\emptyset\}$, we define $S \sqcup T = \{\alpha \sqcup \beta \mid \alpha \in S \wedge \beta \in T\}$, $S\emptyset = \emptyset S = \emptyset$, $S\varepsilon = \{\varepsilon\}S = S$, and, if $\alpha' \neq \emptyset, \varepsilon$,

$$S\alpha' = \{\alpha\alpha' \mid \alpha \in S \wedge \alpha \neq \varepsilon\} \cup \{\alpha' \mid \exists \varepsilon \in S\}.$$

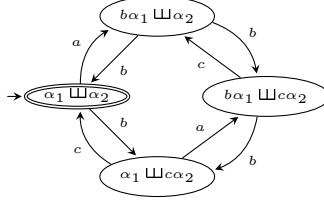
As usual, the set of partial derivatives of $\alpha \in \mathcal{R}(\sqcup)$ w.r.t. a word $w \in \Sigma^*$ is inductively defined by $\partial_\varepsilon(\alpha) = \{\alpha\}$ and $\partial_{w\sigma}(\alpha) = \partial_\sigma(\partial_w(\alpha))$, where, given a set $S \subseteq \mathcal{R}(\sqcup)$, $\partial_\sigma(S) = \bigcup_{\alpha \in S} \partial_\sigma(\alpha)$. Moreover, $\mathcal{L}(\partial_w(\alpha)) = \{w_1 \mid ww_1 \in \mathcal{L}(\alpha)\}$.

Let $\partial(\alpha) = \bigcup_{w \in \Sigma^*} \partial_w(\alpha)$, and $\partial^+(\alpha) = \bigcup_{w \in \Sigma^+} \partial_w(\alpha)$. The partial derivative automaton of $\alpha \in \mathcal{R}(\sqcup)$ is

$$\mathcal{A}_{PD}(\alpha) = \langle \partial(\alpha), \Sigma, \{\alpha\}, \delta_{PD}, F_{PD} \rangle,$$

with $F_{PD} = \{\beta \in \partial(\alpha) \mid \varepsilon(\beta) = \varepsilon\}$ and $\delta_{PD}(\beta, \sigma) = \partial_\sigma(\beta)$, for $\beta \in \partial(\alpha)$, $\sigma \in \Sigma$.

Example 6. The partial derivative automaton for $\alpha_1 \sqcup \alpha_2$, where $\alpha_1 = (ab)^*$ and $\alpha_2 = (bc)^*$, from Example 5, is depicted below.



We note that both $\partial^+(\alpha)$ and $\text{Loc}_0(\alpha)$ are at most of size $2^{|\alpha|_\Sigma}$, cf. [6]. Given an expression α , one can naturally apply any automaton construction \mathcal{A} to the marked expression $\overline{\alpha}$, where transitions are labelled with marked letters. We denote by $\overline{\mathcal{A}(\overline{\alpha})}$ the automaton obtained from $\mathcal{A}(\overline{\alpha})$ by unmarking the labels of transitions, but without changing the labels of the states. Champarnaud and Ziadi [19] proved that, for standard regular expressions, \mathcal{A}_{PD} is a quotient of the position automaton \mathcal{A}_{POS} . It was shown that, given a position i , there exists some expression, denoted by $c(\overline{\alpha}, i)$ and called *c-continuation*, such that for all $w \in \Sigma_{\overline{\alpha}}^*$, either $\partial_{w\sigma_i}(\overline{\alpha}) = \emptyset$ or $\partial_{w\sigma_i}(\overline{\alpha}) = \{c(\overline{\alpha}, i)\}$. This naturally induces a right-invariant relation on the set of positions, where $i \equiv_o j$ if $c(\overline{\alpha}, i) = c(\overline{\alpha}, j)$, and such that $\mathcal{A}_{\text{POS}}(\alpha)/\equiv_o \simeq \overline{\mathcal{A}_{\text{PD}}(\overline{\alpha})}$. For expressions in $\mathcal{R}(\sqcup)$ it is no longer true that given a position i there exists a unique expression $c(\overline{\alpha}, i)$ satisfying the conditions described above. The following is an example of this.

Example 7. Consider $\alpha = (a^* + b)^* \sqcup (c^* + d)^*$ and $\overline{\alpha} = \alpha_1 \sqcup \alpha_2$, where $\alpha_1 = (a_1^* + b_2)^*$ and $\alpha_2 = (c_3^* + d_4)^*$. For the letter a_1 we have $\partial_{a_1}(\overline{\alpha}) = \{a_1^* \alpha_1 \sqcup \alpha_2\}$, while $\partial_{c_3 a_1}(\overline{\alpha}) = \{a_1^* \alpha_1 \sqcup c_3^* \alpha_2\}$.

However, for expressions with shuffle we can associate a unique expression $c(\overline{\alpha}, p)$ to each location p , which will be used to show that, also in this case, \mathcal{A}_{PD} is a quotient of \mathcal{A}_{POS} . Let $c(\overline{\alpha}, 0) = \overline{\alpha}$. The *c-continuation* $c(\overline{\alpha}, p)$ of $\overline{\alpha}$ w.r.t. a location p is defined as:

$$\begin{aligned} c(\sigma_i, i) &= \varepsilon, \quad c(\alpha^*, p) = c(\alpha, p)\alpha^*, \\ c(\alpha_1 + \alpha_2, p) &= \begin{cases} c(\alpha_1, p), & \text{if } p \in \text{Loc}(\alpha_1), \\ c(\alpha_2, p), & \text{if } p \in \text{Loc}(\alpha_2), \end{cases} \\ c(\alpha_1 \alpha_2, p) &= \begin{cases} c(\alpha_1, p)\alpha_2, & \text{if } p \in \text{Loc}(\alpha_1), \\ c(\alpha_2, p), & \text{if } p \in \text{Loc}(\alpha_2), \end{cases} \\ c(\alpha_1 \sqcup \alpha_2, (p_1, p_2)) &= c(\alpha_1, p_1) \sqcup c(\alpha_2, p_2). \end{aligned}$$

Example 8. Consider again $\alpha = (ab)^* \sqcup (bc)^*$ from Example 5. For the elements in $\text{Loc}_0(\alpha)$ we have $c(\overline{\alpha}, 0) = c(\overline{\alpha}, (2, 0)) = c(\overline{\alpha}, (2, 4)) = c(\overline{\alpha}, (0, 4)) = \overline{\alpha}$, $c(\overline{\alpha}, (0, 3)) = c(\overline{\alpha}, (2, 3)) = (a_1 b_2)^* \sqcup c_4 (b_3 c_4)^*$, $c(\overline{\alpha}, (1, 0)) = c(\overline{\alpha}, (1, 4)) = b_2 (a_1 b_2)^* \sqcup (b_3 c_4)^*$, and $c(\overline{\alpha}, (1, 3)) = b_2 (a_1 b_2)^* \sqcup c_4 (b_3 c_4)^*$. The partial derivative automaton of the expression given above is obtained by merging the states in the $\mathcal{A}_{\text{POS}}(\alpha)$ labelled with locations that have the same c-continuation.

To show that $\mathcal{A}_{\text{PD}}(\overline{\alpha})$ is a quotient of $\mathcal{A}_{\text{POS}}(\overline{\alpha})$, we first prove that the set of all c-continuations is precisely $\partial^+(\overline{\alpha})$ (Lemma 8). Furthermore, p is a final

state in $\mathcal{A}_{\text{POS}}(\bar{\alpha})$ if and only if $c(\bar{\alpha}, p)$ is a final state in $\mathcal{A}_{\text{PD}}(\bar{\alpha})$ (Lemma 9). Finally, in Proposition 10 we relate $\partial_{\sigma_i}(c(\bar{\alpha}, p))$ with $\text{Follow}(\bar{\alpha}, p)$. Due to their length, the proofs can be found in the appendix.

Lemma 8. *Let $\alpha \in \mathcal{R}(\sqcup)$. Then, $\partial^+(\bar{\alpha}) = \{ c(\bar{\alpha}, p) \mid p \in \text{Loc}(\alpha) \}$.*

Lemma 9. *For $\alpha \in \mathcal{R}(\sqcup)$ and $p \in \text{Loc}(\alpha)$, $\varepsilon(c(\bar{\alpha}, p)) = \varepsilon \iff p \in \text{Last}(\alpha)$.*

Proof. By structural induction on $\bar{\alpha}$. \square

The next proposition relates derivatives of $c(\bar{\alpha}, p)$ with $\text{Follow}(\alpha, p)$.

Proposition 10. *For $\alpha \in \mathcal{R}(\sqcup)$, $p \in \text{Loc}_0(\alpha)$, and $\sigma_i \in \Sigma_{\bar{\alpha}}$, one has*

$$\beta \in \partial_{\sigma_i}(c(\bar{\alpha}, p)) \iff \exists q \in \text{Loc}(\alpha) : \beta = c(\bar{\alpha}, q) \wedge i \in \text{l2pos}(q) \wedge (\bar{\sigma}_i, q) \in \text{Follow}(\alpha, p).$$

Now, the equivalence relation \equiv_o on $\text{Loc}_0(\alpha)$, that defines $\mathcal{A}_{\text{PD}}(\bar{\alpha})$ as a quotient of $\mathcal{A}_{\text{POS}}(\bar{\alpha})$, is defined by $p \equiv_o q$ if $c(\bar{\alpha}, p) = c(\bar{\alpha}, q)$.

Lemma 11. *The relation \equiv_o is right-invariant w.r.t. $\mathcal{A}_{\text{POS}}(\alpha)$.*

Proof. Consider $p, q \in \text{Loc}_0(\alpha)$ such that $p \equiv_o q$, i.e., $c(\bar{\alpha}, p) = c(\bar{\alpha}, q)$. By Lemma 9 we have $p \in \text{Last}(\alpha)$ if and only if $q \in \text{Last}(\alpha)$. Let $(\bar{\sigma}_i, p') \in \text{Follow}(\alpha, p)$ with $i \in \text{l2pos}(p')$. By Proposition 10, one gets

$$c(\bar{\alpha}, p') \in \partial_{\sigma_i}(c(\bar{\alpha}, p)) = \partial_{\sigma_i}(c(\bar{\alpha}, q)),$$

and also that there is $q' \in \text{Loc}(\alpha)$ such that $(\bar{\sigma}_i, q') \in \text{Follow}(\alpha, q)$ and $c(\bar{\alpha}, q') = c(\bar{\alpha}, p')$, i.e., $p' \equiv_o q'$. \square

Example 9. *Consider again $\alpha = (ab)^* \sqcup (bc)^*$ from Example 5. Recall that $c(\bar{\alpha}, (2, 0)) = c(\bar{\alpha}, (0, 4)) = \bar{\alpha}$, i.e., $(2, 0) \equiv_o (0, 4)$. Furthermore,*

$$\begin{aligned} (a_1 b_2)^* \sqcup c_4(b_3 c_4)^* &\in \partial_{b_3}(\bar{\alpha}), \\ (b, (2, 3)) &\in \text{Follow}(\alpha, (2, 0)), \\ (b, (0, 3)) &\in \text{Follow}(\alpha, (0, 4)), \text{ and } (2, 3) \equiv_o (0, 3). \end{aligned}$$

Proposition 12. $\mathcal{A}_{\text{POS}}(\alpha)/\equiv_o \simeq \overline{\mathcal{A}_{\text{PD}}(\bar{\alpha})}$.

Proof. We show that the function $\varphi_c : \text{Loc}_0(\alpha)/\equiv_o \longrightarrow \partial^+(\alpha)$, defined by $\varphi_c([p]) = c(\bar{\alpha}, p)$, is an isomorphism. Injectivity follows from Lemma 11 and surjectivity from Lemma 8. For the initial state we have $\varphi_c([0]) = c(\bar{\alpha}, 0) = \bar{\alpha}$. Furthermore, by Lemma 9, $[p]$ is a final state in $\mathcal{A}_{\text{POS}}(\alpha)/\equiv_o$ if and only if $\varphi_c([p])$ is a final state in $\partial^+(\alpha)$. Finally,

$$\begin{aligned} \varphi_c(\delta_{\text{POS}/\equiv_o}([p], \sigma)) &= \varphi_c(\{ [q] \mid (\sigma, q) \in \text{Follow}(\alpha, p) \}) \\ &= \{ c(\bar{\alpha}, q) \mid (\sigma, q) \in \text{Follow}(\alpha, p) \} \\ &= \bigcup_{\bar{\sigma}_i = \sigma} \partial_{\sigma_i}(c(\bar{\alpha}, p)) = \delta_{\text{PD}}(\varphi_c([p]), \sigma). \end{aligned}$$

\square

Broda et al. [6] showed that $\mathcal{A}_{\text{PD}}(\alpha)$ is a quotient of $\overline{\mathcal{A}_{\text{PD}}(\overline{\alpha})}$ by the right-invariant equivalence relation \equiv_2 defined on the states of $\overline{\mathcal{A}_{\text{PD}}(\overline{\alpha})}$ by $\beta_1 \equiv_2 \beta_2$ if $\overline{\beta_1} = \overline{\beta_2}$. Let \equiv_c be the relation $\equiv_2 \circ \equiv_o$. Thus, we have the following result.

Proposition 13. $\mathcal{A}_{\text{POS}}(\alpha)/\equiv_c \simeq (\overline{\mathcal{A}_{\text{PD}}(\overline{\alpha})})/\equiv_2 \simeq \mathcal{A}_{\text{PD}}(\alpha)$.

Example 10. It follows from the c-continuations computed in Example 8 for $\alpha = (ab)^* \sqcup (bc)^*$, that there are no $\beta_1 \neq \beta_2 \in \partial^+(\overline{\alpha})$ such that $\beta_1 \equiv_2 \beta_2$. Consequently, in this particular case, $\overline{\mathcal{A}_{\text{PD}}(\overline{\alpha})} \simeq \mathcal{A}_{\text{PD}}(\alpha)$.

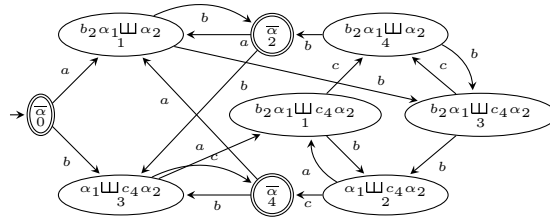
5. $\mathcal{A}_{\text{POS}}(\alpha)$ vs. $\mathcal{A}_{\partial\text{pos}}(\alpha)$

In this section, we relate the position automaton defined in this paper with the one presented by Broda et al. [6]. The states of $\mathcal{A}_{\partial\text{pos}}$ are labelled by pairs (γ, i) , where i is a position of a letter in the original expression, and $\gamma \in \mathcal{R}(\sqcup)$ describes the right-language of the state. Given $\alpha \in \mathcal{R}(\sqcup)$, the automaton obtained by that construction will be denoted by $\mathcal{A}_{\partial\text{pos}}(\alpha)$, and is defined by

$$\mathcal{A}_{\partial\text{pos}}(\alpha) = \langle S_{\partial\text{pos}}^0(\alpha), \Sigma, \{(\overline{\alpha}, 0)\}, \delta_{\partial\text{pos}}, F_{\partial\text{pos}} \rangle,$$

where $S_{\partial\text{pos}}^0(\alpha) = \{(\overline{\alpha}, 0)\} \cup \{(\gamma, i) \mid \gamma \in \partial_{\sigma_i}(\partial(\overline{\alpha})), \sigma_i \in \Sigma_{\overline{\alpha}}\}$, $F_{\partial\text{pos}} = \{(\gamma, i) \in S_{\partial\text{pos}}^0(\alpha) \mid \varepsilon(\gamma) = \varepsilon\}$ and $\delta_{\partial\text{pos}}((\gamma, i), \sigma) = \{(\beta, j) \mid \beta \in \partial_{\sigma_j}(\gamma), \sigma = \overline{\sigma_j}\}$.

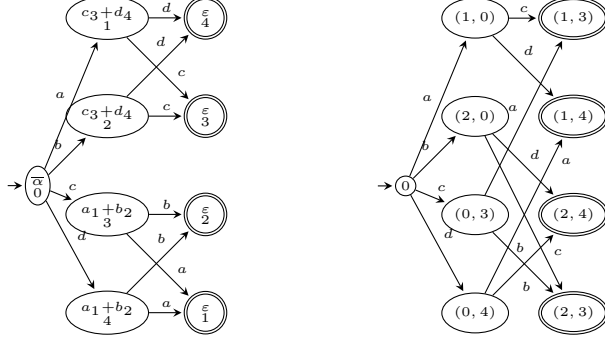
Example 11. Consider the expression $\alpha = (ab)^* \sqcup (bc)^*$ from Example 5, with $\overline{\alpha} = \alpha_1 \sqcup \alpha_2$, where $\alpha_1 = (a_1 b_2)^*$ and $\alpha_2 = (b_3 c_4)^*$. The $\mathcal{A}_{\partial\text{pos}}(\alpha)$ is depicted below.



Merging the states whose labels contain the same expression, we obtain $\overline{\mathcal{A}_{\text{PD}}(\overline{\alpha})}$, which in this case coincides with $\mathcal{A}_{\text{PD}}(\alpha)$, displayed in Example 6.

For α in the previous example neither $\mathcal{A}_{\text{POS}}(\alpha)$ is a right-quotient of $\mathcal{A}_{\partial\text{pos}}(\alpha)$, nor vice-versa. This can also be seen in the following example.

Example 12. Consider $\alpha = (a + b) \sqcup (c + d)$ with $\overline{\alpha} = (a_1 + b_2) \sqcup (c_3 + d_4)$. Both, $\mathcal{A}_{\partial\text{pos}}(\alpha)$ and $\mathcal{A}_{\text{POS}}(\alpha)$, depicted below, have nine states. However, words ac and bc lead in $\mathcal{A}_{\partial\text{pos}}(\alpha)$ to the same final state, and in $\mathcal{A}_{\text{POS}}(\alpha)$ to different final states. This shows that the automata are not isomorphic, hence neither of them is a quotient of the other.



It was shown [6] that $\overline{\mathcal{A}_{PD}(\overline{\alpha})}$ is a quotient of $\mathcal{A}_{\partial pos}(\alpha)$ by the right-invariant equivalence relation \equiv_1 defined on the set of states in $\mathcal{A}_{\partial pos}(\alpha)$ by $(\beta_1, i) \equiv_1 (\beta_2, j)$ if $\beta_1 = \beta_2$. Thus, we obtain the following relation between $\mathcal{A}_{POS}(\alpha)$ and $\mathcal{A}_{\partial pos}(\alpha)$.

Corollary 1. $\mathcal{A}_{\partial pos}(\alpha)/\equiv_1 \simeq \mathcal{A}_{POS}(\alpha)/\equiv_o \simeq \overline{\mathcal{A}_{PD}(\overline{\alpha})}$.

The average number of states of \mathcal{A}_{PD} , which is $(\frac{4}{3} + o(1))^{|\alpha|_\Sigma}$, was estimated using an upper bound $p(\alpha)$ for the number of elements in $\partial(\alpha)$ (see [6]). The value of $p(\alpha)$ is precisely $|\text{Loc}(\alpha)|$, obtained by the definition 2. Thus, we conclude that the average number of states of \mathcal{A}_{POS} is the same. However, an analogous analysis gives an upper bound for the average number of states for $\mathcal{A}_{\partial pos}$ of $(\frac{5}{3} + o(1))^{|\alpha|_\Sigma}$ (see [6]).

5.1. \mathcal{A}_{POS} , $\mathcal{A}_{\partial pos}$, and Standard Regular Expressions

Clearly, all results established in Section 4 and 5 hold when considering standard regular expressions, i.e., expressions without the shuffle operator. In this section we derive known results [19, 13] using our approach for this particular subcase.

Let τ denote a standard regular expression. Then, it is easy to see that $\text{Loc}(\tau) = \text{Pos}(\tau)$. Lemmas 4, 5, and 6 ensure that the inductive definitions of the sets **First**, **Last**, and **Follow** correspond precisely to the usual semantic interpretation of these sets, as defined in Section 2. In particular, this implies that the definition of \mathcal{A}_{POS} corresponds to the standard one. The same is true for the notion of c -continuation $c(\overline{\alpha}, i)$ of a position i . The fact that \mathcal{A}_{PD} is a quotient of \mathcal{A}_{POS} follows from Lemmas 8, 9, and Proposition 10, which in the present case read as follows, respectively.

- $\partial^+(\overline{\tau}) = \{c(\overline{\tau}, i) \mid i \in \text{Pos}(\tau)\};$
- $\varepsilon(c(\overline{\tau}, i)) = \varepsilon \iff i \in \text{Last}(\tau);$
- $c(\overline{\tau}, i) \in \partial_{\sigma_i}(c(\overline{\tau}, j)) \iff (\overline{\sigma_i}, i) \in \text{Follow}(\tau, j).$

Finally, note that, for standard regular expressions, $\mathcal{A}_{\partial pos}$ also coincides (up to isomorphism) with the standard position automaton. This is due to the fact that, whenever $\gamma, \gamma' \in \partial_{\sigma_i}(\partial(\overline{\tau}))$ then $\gamma = \gamma' = c(\overline{\tau}, i)$.

6. $\mathcal{A}_{\text{Pre}}(\alpha)$ and $\mathcal{A}_{\text{Pos}}(\alpha)$

A conversion from regular expressions to automata, which has been recently studied, is the prefix-automaton \mathcal{A}_{Pre} [11, 12, 16]. For standard regular expressions \mathcal{A}_{Pre} is a left-quotient of the \mathcal{A}_{Pos} , and for linear expressions $\bar{\alpha}$ one has $\mathcal{A}_{\text{Pos}}(\bar{\alpha}) \simeq \mathcal{A}_{\text{Pre}}(\bar{\alpha})$. Being a left-quotient also implies that the determinisation of \mathcal{A}_{Pre} coincides with the determinisation of \mathcal{A}_{Pos} [16]. Below we define an extension of the \mathcal{A}_{Pre} construction for expressions in $\mathcal{R}(\sqcup)$. However, the relationship with the position automaton doesn't hold any more, neither for \mathcal{A}_{Pos} , nor for $\mathcal{A}_{\partial\text{pos}}$. Every state in \mathcal{A}_{Pre} is labelled either with ε or with an expression of the form $\alpha\sigma$, describing the left-language of that state. To obtain those expressions, one uses a function R that, given an expression α , computes a set of normalised expressions of the form $\alpha'\sigma$. For $\alpha \in \mathcal{R}(\sqcup)$, the set $R(\alpha)$ is given by

$$\begin{aligned} R(\emptyset) &= R(\varepsilon) = \emptyset, \\ R(\sigma) &= \{\sigma\}, & R(\alpha^*) &= \alpha^* R(\alpha), \\ R(\alpha_1 + \alpha_2) &= R(\alpha_1) \cup R(\alpha_2), & R(\alpha_1 \alpha_2) &= \alpha R(\alpha_2) \cup \varepsilon(\alpha_2) R(\alpha_1), \\ R(\alpha_1 \sqcup \alpha_2) &= \{(\alpha'_1 \sqcup \alpha_2)\sigma \mid \alpha'_1\sigma \in R(\alpha_1)\} \cup \{(\alpha_1 \sqcup \alpha'_2)\sigma \mid \alpha'_2\sigma \in R(\alpha_2)\}. \end{aligned} \tag{6}$$

One can see that $R_\varepsilon(\alpha) = R(\alpha) \cup \varepsilon(\alpha)$ is such that $\mathcal{L}(R_\varepsilon(\alpha)) = \mathcal{L}(\alpha)$. Thus, it is the set of final states of $\mathcal{A}_{\text{Pre}}(\alpha)$. Then, the remaining construction of the automaton is done backwards. For each state of the form $\alpha'\sigma$ the set $R_\varepsilon(\alpha')$ is computed and a transition by σ is added from each element $\alpha'' \in R_\varepsilon(\alpha')$ to $\alpha'\sigma$. The state labelled by ε is the initial state of $\mathcal{A}_{\text{Pre}}(\alpha)$. Formally, consider the function $\mathbf{p}_w(\alpha)$ for words $w \in \Sigma^*$ defined as follows: $\mathbf{p}_\varepsilon(\alpha) = R_\varepsilon(\alpha)$, and

$$\mathbf{p}_{\sigma w}(\alpha) = \bigcup_{\alpha'\sigma \in \mathbf{p}_w(\alpha)} R_\varepsilon(\alpha').$$

We have that $\mathcal{L}(\mathbf{p}_w(\alpha)) = \{x \mid xw \in \mathcal{L}(\alpha)\}$. The *prefix* automaton of α is

$$\mathcal{A}_{\text{Pre}}(\alpha) = \langle \text{Pre}(\alpha), \Sigma, \delta_{\text{Pre}}, \varepsilon, R_\varepsilon(\alpha) \rangle,$$

where $\text{Pre}(\alpha) = \bigcup_{w \in \Sigma^*} \mathbf{p}_w(\alpha)$, and

$$\delta_{\text{Pre}} = \{(\alpha'', \sigma, \alpha'\sigma) \mid \alpha'\sigma \in \text{Pre}(\alpha), \alpha'' \in R_\varepsilon(\alpha'), \sigma \in \Sigma\},$$

that is, for all $\alpha'\sigma \in \text{Pre}(\alpha)$, $\delta_{\text{Pre}}^R(\alpha'\sigma, \sigma) = R_\varepsilon(\alpha')$.

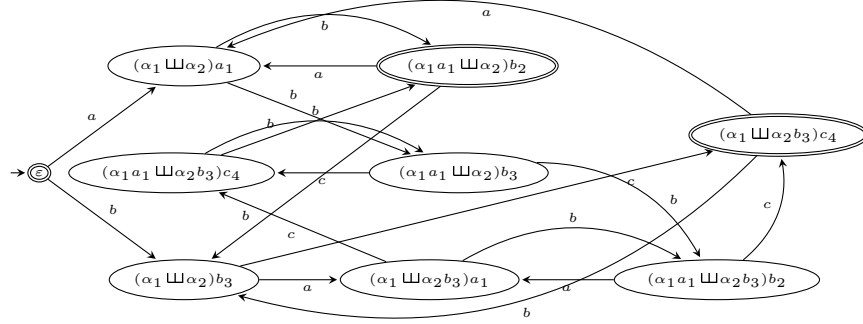
Proposition 14. $\mathcal{L}(\mathcal{A}_{\text{Pre}}(\alpha)) = \mathcal{L}(\alpha)$.

Proof. Based on the construction of \mathcal{A}_{Pre} for standard regular expressions one only needs to prove that $\mathcal{L}(R_\varepsilon(\alpha_1 \sqcup \alpha_2)) = \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. First $\varepsilon \in \mathcal{L}(R_\varepsilon(\alpha_1 \sqcup \alpha_2))$ if and only if $\varepsilon \in \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. Let $x \in \mathcal{L}(R_\varepsilon(\alpha_1 \sqcup \alpha_2))$ and suppose that $x \in \mathcal{L}((\alpha'_1 \sqcup \alpha_2)\sigma)$ with $\alpha'_1\sigma \in R(\alpha_1)$ (the other case is analogous). Then $x = x'\sigma$ and $x' \in w_1 \sqcup w_2$, where $w_1 \in \mathcal{L}(\alpha'_1)$ and $w_2 \in \mathcal{L}(\alpha_2)$. But then

we have $w_1\sigma \in \mathcal{L}(\mathcal{R}(\alpha_1)) \subseteq \mathcal{L}(\alpha_1)$ and $x \in \mathcal{L}(\alpha_1) \sqcup \mathcal{L}(\alpha_2) = \mathcal{L}(\alpha_1 \sqcup \alpha_2)$. If $x \in \mathcal{L}(\alpha_1 \sqcup \alpha_2) \setminus \{\varepsilon\}$ then $x \in w_1 \sqcup w_2$, with $w_1 \in \mathcal{L}(\alpha_1)$ and $w_2 \in \mathcal{L}(\alpha_2)$. Let $x = x'\sigma$, $w_2 = w'_2\sigma$ and $w_2 \in \mathcal{L}(\alpha'_2\sigma)$, for some $\alpha'_2\sigma \in \mathcal{R}(\alpha_2)$. Then $x \in \mathcal{L}((\alpha_1 \sqcup \alpha'_2)\sigma) \subseteq \mathcal{L}(\mathcal{R}_\varepsilon(\alpha_1 \sqcup \alpha_2))$. For $w_1 = w'_1\sigma$ the proof is similar. \square

For expressions with shuffle, \mathcal{A}_{Pre} is neither a quotient of \mathcal{A}_{POS} , nor of $\mathcal{A}_{\partial\text{pos}}$. The following example shows, that for expressions with shuffle \mathcal{A}_{Pre} is not a quotient of \mathcal{A}_{POS} .

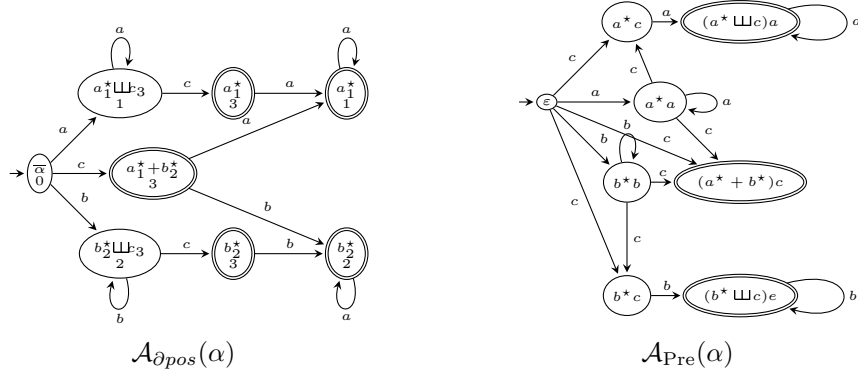
Example 13. Consider the expression $\alpha = (ab)^* \sqcup (bc)^*$ with $\bar{\alpha} = \alpha_1 \sqcup \alpha_2$, where $\alpha_1 = (a_1b_2)^*$ and $\alpha_2 = (b_3c_4)^*$ (of Example 5). The automaton $\mathcal{A}_{\text{Pre}}(\bar{\alpha})$, displayed below, does not coincide with $\mathcal{A}_{\text{POS}}(\alpha)$, contrary to the case for regular expressions without shuffle [16].



The automaton $\mathcal{A}_{\text{Pre}}(\alpha)$ is obtained from $\overline{\mathcal{A}_{\text{Pre}}(\bar{\alpha})}$ by merging states that after unmarking are labelled with identical expressions. One can verify that in this case $\mathcal{A}_{\text{Pre}}(\alpha)$ is not a quotient of $\mathcal{A}_{\text{POS}}(\alpha)$. Moreover, the determinisation of \mathcal{A}_{Pre} does not coincide with the determinisation of \mathcal{A}_{POS} . Also, the determinisation of $\mathcal{A}_{\text{Pre}}(\alpha)$ does not coincide with the determinisation of $\mathcal{A}_{\text{POS}}(\alpha)$.

The following example shows that, in general, $\mathcal{A}_{\text{Pre}}(\alpha)$ is neither a quotient of $\mathcal{A}_{\partial\text{pos}}(\alpha)$.

Example 14. For $\alpha = (a^* + b^*) \sqcup c$ the automata $\mathcal{A}_{\partial\text{pos}}(\alpha)$ and $\mathcal{A}_{\text{Pre}}(\alpha)$ are depicted below.



6.1. Experimental Results

Table 1: Experimental results.

k	n	$ \alpha _{\Sigma}$	$ Q_{\text{POS}} $	$ Q_{\text{PD}} $	$ Q_{\text{Pre}} $	$ \delta_{\text{POS}} $	$ \delta_{\text{PD}} $	$ \delta_{\text{Pre}} $	$\frac{ Q_{\text{PD}} }{ Q_{\text{POS}} }$	$\frac{ \delta_{\text{PD}} }{ \delta_{\text{POS}} }$
2	10	3.13	5.71	4.02	5.33	6.28	10.18	8.51	0.70	0.62
	20	6.01	16.73	9.89	15.11	25.84	50.39	40.68	0.59	0.51
	30	8.85	43.15	21.07	36.69	75.11	180.96	136.83	0.49	0.42
	40	11.72	101.65	42.13	80.46	188.73	532.59	374.72	0.41	0.35
	50	14.59	250.87	85.20	177.69	455.14	1606.65	988.14	0.34	0.28
5	10	4.02	7.82	5.41	8.57	15.08	9.61	15.51	0.69	0.64
	20	7.84	28.38	16.42	34.79	88.81	47.33	101.45	0.58	0.53
	30	11.58	91.74	47.06	118.45	393.64	188.81	477.92	0.51	0.48
	40	15.27	281.40	109.41	352.17	1595.98	559.48	1861.45	0.39	0.35
	50	19.04	790.81	252.47		5345.74	1537.58		0.32	0.29
10	10	4.47	9.03	6.24	10.77	17.86	11.66	20.25	0.69	0.65
	20	8.76	37.75	22.09	55.32	119.51	66.81	166.57	0.59	0.56
	30	12.97	130.96	63.03	204.80	566.82	259.10	843.73	0.48	0.46
	40	7.14	463.53	181.01		2636.58	961.48		0.39	0.36
	50	21.34	1491.69	493.65		10273.77	3197.12		0.33	0.31

For standard regular expressions the average sizes of \mathcal{A}_{POS} , \mathcal{A}_{PD} , and \mathcal{A}_{Pre} have been studied both experimentally using uniform random generated expressions, and asymptotically using the framework of analytic combinatorics [20]. It was shown that, asymptotically, the size of the \mathcal{A}_{PD} is half the size of \mathcal{A}_{POS} , and \mathcal{A}_{Pre} is almost the same size of \mathcal{A}_{POS} . Both \mathcal{A}_{POS} and \mathcal{A}_{PD} can be computed in time $O(n^2)$, but to compute \mathcal{A}_{PD} , in general, the \mathcal{A}_{POS} must be first computed [21].

In order to compare, on average, the sizes of \mathcal{A}_{POS} , \mathcal{A}_{PD} , and \mathcal{A}_{Pre} for expressions with shuffle we performed some experiments, using the Fado package [22]. Regular expressions $\alpha \in \mathcal{R}(\sqcup)$ were uniformly random generated using a version of the grammar (1) (with the shuffle operator) in prefix notation [23]. For each expression size, n , and alphabet size, k , samples of 10000 expressions were generated. This is sufficient to ensure a 95% confidence level within a 1% error margin [24]. Due to the exponential blow-up of the size of the automata, only small values of n and k were used. Table 1 presents some average results for $n \in \{10, 20, 30, 40, 50\}$ and $k \in \{2, 5, 10\}$.

Column three represents the average alphabetic size of the expressions. Columns four up to nine give the average number of states and the average

number of transitions for each construction. The last two columns present the ratios of the size of states and of the size of transitions, respectively, between \mathcal{A}_{PD} and \mathcal{A}_{POS} . As expected, the size of the \mathcal{A}_{PD} is never larger than the size of \mathcal{A}_{POS} , but it is not clear if those ratios tend to $\frac{1}{2}$. Runtime of each construction is exponential in the size of the expression. However, in the current implementation, \mathcal{A}_{PD} construction [?] seems the faster. For instance for $k = 5$ and $n = 30$ the runtime per expression for \mathcal{A}_{POS} , \mathcal{A}_{PD} , and \mathcal{A}_{Pre} were, respectively, 0.0022, 0.0019 and 1.117 seconds in a 2.7 GHz Intel Core I7. Some values are missing for \mathcal{A}_{Pre} as their computation would take too much time.

7. Relation with other Constructions

Although \mathcal{A}_{POS} and \mathcal{A}_{Pre} are incomparable, the relationship between \mathcal{A}_{Pre} and \mathcal{A}_{PD} established in Broda et al. [16] still holds for the set $\mathcal{R}(\sqcup)$. To show that, it is enough to consider the dual reversal of \mathcal{A}_{Pre} , i.e., $\mathcal{A}_{\overleftarrow{Pre}}(\alpha) \simeq \mathcal{A}_{Pre}(\alpha^R)^R$. Defining $L(\alpha) = R(\alpha^R)^R$ and $\overleftarrow{p}_w(\alpha)$ as $p_w(\alpha)$, but using L instead of R , we have

$$\mathcal{A}_{\overleftarrow{Pre}}(\alpha) = \langle \overleftarrow{Pre}(\alpha), \Sigma, \delta_{\overleftarrow{Pre}}, L_\varepsilon(\alpha), \varepsilon \rangle,$$

where $\overleftarrow{Pre}(\alpha) = \bigcup_{w \in \Sigma^*} \overleftarrow{p}_w(\alpha)$ and $\delta_{\overleftarrow{Pre}}(\alpha', \sigma) = L_\varepsilon(\alpha'')$ if $\alpha' = \sigma\alpha''$, and $\delta_{\overleftarrow{Pre}}(\alpha', \sigma) = \emptyset$ otherwise, for $\sigma \in \Sigma$. The following lemma establishes the relationship between L and partial derivatives for $\alpha \in \mathcal{R}(\sqcup)$.

Lemma 15. $L_\varepsilon(\alpha) = \bigcup_{\sigma \in \Sigma} \sigma \partial_\sigma(\alpha) \cup \varepsilon(\alpha)$.

Proof. We only need to prove the case $\alpha = \alpha_1 \sqcup \alpha_2$, which is obvious considering the definition of $\partial_\sigma(\alpha)$ and the fact that

$$L(\alpha_1 \sqcup \alpha_2) = \{ \sigma(\alpha'_1 \sqcup \alpha_2) \mid \sigma\alpha'_1 \in L(\alpha_1) \} \cup \{ \sigma(\alpha_1 \sqcup \alpha'_2) \mid \sigma\alpha'_2 \in L(\alpha_2) \}. \quad \square$$

With Lemma 15 one can prove that the determinisation of $\mathcal{A}_{\overleftarrow{Pre}}$ is isomorphic to a quotient of the determinisation of \mathcal{A}_{PD} by the right-invariant relation (\equiv_{L_ε}) , defined as follows

$$X \equiv_{L_\varepsilon} X' \Leftrightarrow L_\varepsilon(X) = L_\varepsilon(X'),$$

where $X, X' \subseteq PD(\alpha)$, and $L_\varepsilon(X)$ denotes $\bigcup_{\alpha' \in X} L_\varepsilon(\alpha')$ [16, Prop. 19]. The same holds if one considers Brzozowski derivatives [25] extended with shuffle and the correspondent deterministic automaton (B in Figure 1).

Broda et al. [16] established relations between different conversions from regular expressions to equivalent finite automata, using the notion of position, the sets **Follow** and **Select**, and operations such as quotients, determinisation and reversal. These constructions are the Follow automaton (\mathcal{A}_F) [13], the *Aut Point* automaton (\mathcal{A}_{MB}) [14, 15], the McNaughton-Yamada automaton (\mathcal{A}_{MY} , \mathcal{A}_{MA}) [26, 15], as well as some dual constructions using a double reversal. Considering locations instead of positions and the definitions of **Follow** and **Select** given in this paper, these constructions are now automatically defined for expressions in $\mathcal{R}(\sqcup)$. Moreover, all the relationships established between them extend to expressions with shuffle. Those relationships are depicted in Figure 1.

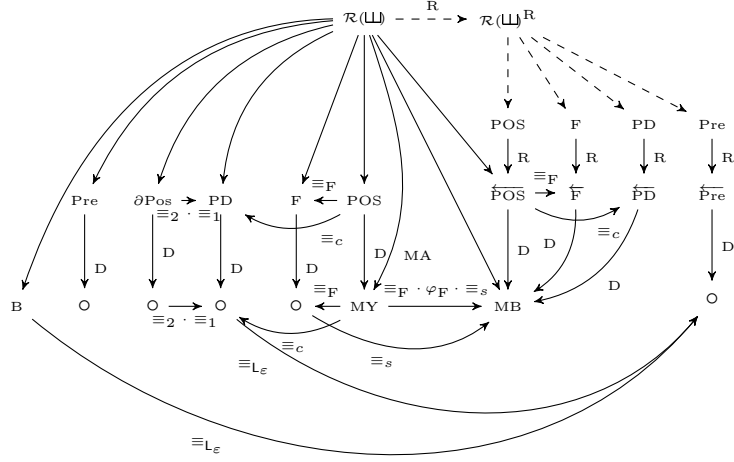


Figure 1: Taxonomy of conversions for regular expressions with shuffle to finite automata. Nodes correspond to models and nodes labeled by \overleftarrow{A} denote a double reverse construction $A(\alpha^R)^R$. Edges correspond to operations or conversions between models. The edges labelled by R correspond to the reversal operation, and the ones labelled by D to determinisation. The remaining labelled edges correspond to quotients where the labels identify the defining relation (see [16] for details).

In contrast to the situation for RE, for $\mathcal{R}(\sqcup)$ we cannot ensure that $D(\mathcal{A}_{\overleftarrow{\text{Pre}}})$ is always the smallest DFA among the ones present in that figure, as it is incomparable (for instance) with \mathcal{A}_{MB} . For convenience, in the next paragraph we recall the definitions of \mathcal{A}_{MB} and \mathcal{A}_{MA} .

Asperti *et al.* [14] introduced the notion of pointed regular expression in order to obtain a compact representation of a set of positions. A point is used to mark a position to be visited when reading a letter instead of a position reached after reading the letter, as is the case for \mathcal{A}_{POS} . The resulting construction was called *mark before*, \mathcal{A}_{MB} , by Nipkow and Traytel. In the framework developed in [16], this means that δ_{MB} is a composition of Follow with Select. Formally, given $\alpha \in \mathcal{R}(\sqcup)$, let

$$\mathcal{A}_{\text{MB}}(\alpha) = \langle Q_{\text{MB}}, \Sigma, \delta_{\text{MB}}, (\text{Follow}(\alpha, 0), \varepsilon(0)), F_{\text{MB}} \rangle,$$

where $Q_{\text{MB}} \subseteq 2^{\text{Pos}(\alpha)} \times \{\emptyset, \varepsilon\}$, and for $(S, c) \in Q_{\text{MB}}$ and $\sigma \in \Sigma$,

$$\delta_{\text{MB}}((S, c), \sigma) = (\text{Follow}(\alpha, \text{Select}(S, \sigma)), \varepsilon(\text{Select}(S, \sigma))),$$

and $F_{\text{MB}} = \{(S, c) \mid c = \varepsilon\}$. In Q_{MB} we consider only the states that are accessible from the initial state by δ_{MB} . As mentioned before, this construction contrasts with the one of \mathcal{A}_{POS} and of its determinisation, the McNaughton-Yamada DFA, that can be defined as

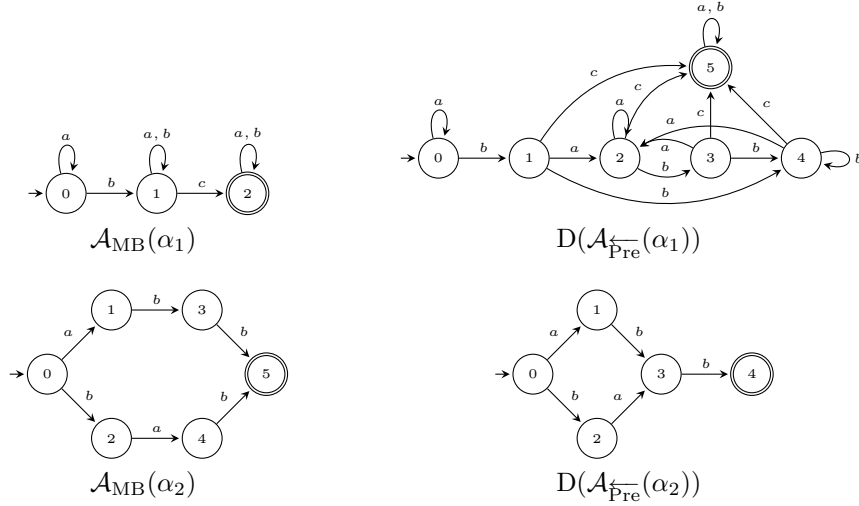
$$\mathcal{A}_{\text{MY}}(\alpha) = D(\mathcal{A}_{\text{POS}}(\alpha)) = \langle Q_{\text{MY}}, \Sigma, \delta_{\text{MY}}, \{0\}, F_{\text{MY}} \rangle,$$

where $Q_{MY} \subseteq 2^{\text{Pos}_0(\alpha)}$, $F_{MY} = \{S \in Q_{MY} \mid \varepsilon(S) = \varepsilon\}$ and for $S \in 2^{\text{Pos}_0(\alpha)}$, $\sigma \in \Sigma$,

$$\delta_{MY}(S, \sigma) = \text{Select}(\text{Follow}(\alpha, S), \sigma).$$

Because of the behaviour of the transition function δ_{MY} of $\mathcal{A}_{MY}(\alpha)$, Nipkow and Traytel called this construction *mark after* ($\mathcal{A}_{MA}(\alpha)$).

Example 15. *The following examples show that $D(\mathcal{A}_{Pre}(\alpha_1))$ and \mathcal{A}_{MB} are incomparable. Considering $\alpha_1 = (a + b^*)^* \sqcup (bc)^*$, $\mathcal{A}_{MB}(\alpha_1)$ has three states (and it is minimal) and $D(\mathcal{A}_{Pre}(\alpha_1))$ has eight states. While for $\alpha_2 = b \sqcup ab$, the $\mathcal{A}_{MB}(\alpha_2)$ has seven states and $D(\mathcal{A}_{Pre}(\alpha_2))$ has six states (and it is minimal). All four automata are represented below, where the dead states have been omitted.*



8. Location Automaton for Regular Expressions with Intersection

In this section we consider regular expressions with the intersection operator. The set $\mathcal{R}(\cap)$ of *regular expressions with intersection* over Σ is obtained by extending grammar (1) with the \cap operator (instead of \sqcup), where $\mathcal{L}(\alpha \cap \beta) = \mathcal{L}(\alpha) \cap \mathcal{L}(\beta)$. Note that intersection corresponds to strict synchronisation of concurrent events. As such, among the different kinds of concurrency operators, it is the opposite extreme of the shuffle operator, which corresponds to pure interleaving. For expressions with intersection, a position automaton was defined by Broda et al. [17, 27]. In this section, we show that using locations one can construct an automaton for regular expressions with intersection, that is isomorphic to the position automaton in [17, 27]. For $i \in \text{Pos}(\alpha)$, let $\ell(i) = \sigma$ for $\bar{\sigma}_i = \sigma$ and let $\ell(I) = \sigma$ if for all $i \in I \subseteq \text{Pos}(\alpha)$ one has $\ell(i) = \sigma$.

To gain some intuition on the (new) definitions of sets **First**, **Follow** and **Last** we consider the following example from [17].

Example 16. Let $\alpha = (ba^*b+a) \cap (aa+b)^*$ with $\bar{\alpha} = (b_1a_2^*b_3+a_4) \cap (a_5a_6+b_7)^* = \alpha_1 \cap \alpha_2$. We have $\text{First}(\alpha_1) = \{1, 4\}$ and $\text{First}(\alpha_2) = \{5, 7\}$. The intersection operator requires that one proceeds with the same letter simultaneously in both expressions. As such, the set of locations reachable from the initial state 0 is $\{(4, 5), (1, 7)\}$. The location $(4, 5)$ is reached with an a , which is the letter corresponding to both positions 4 and 5, i.e. $\ell(4) = \ell(5) = a$. Moreover, location $(5, 7)$ can be reached by $b = \ell(5) = \ell(7)$. In fact, for any location p all the positions in $\text{l2pos}(p)$ correspond to the same letter, i.e. $\ell(\text{l2pos}(p)) = \{\sigma\}$. Therefore the letter σ can be omitted in the definitions of the sets **First** and **Follow**.

For $\alpha \in \mathcal{R}(\cap)$ the positions that appear in a location correspond all to the same letter. Thus, we extend the function ℓ to locations p , that satisfy the condition $\ell(\text{l2pos}(p)) = \{\sigma\}$, by $\ell(p) = \sigma$. Let X and Y be two sets of locations satisfying that condition. We define

$$X \otimes Y = \{ (p_1, p_2) \mid \ell(p_1) = \ell(p_2) \wedge p_1 \in X \wedge p_2 \in Y \}.$$

Then, the set of locations for the intersection of two expressions is

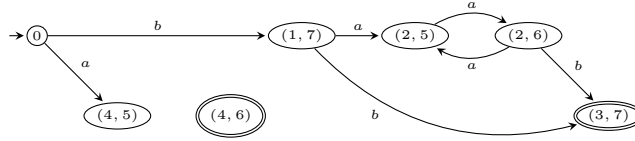
$$\text{Loc}(\alpha_1 \cap \alpha_2) = \text{Loc}(\alpha_1) \otimes \text{Loc}(\alpha_2).$$

Furthermore, for $\alpha \in \mathcal{R}(\cap)$ let $\text{First}(\alpha)$, $\text{Last}(\alpha)$, and $\text{Follow}(\alpha, p) \subseteq \text{Loc}(\alpha)$, for $p \in \text{Loc}(\alpha)$, be

$$\begin{aligned} \text{First}(\sigma_i) &= \{i\}, \\ \text{First}(\alpha_1 \cap \alpha_2) &= \text{First}(\alpha_1) \otimes \text{First}(\alpha_2), \\ \text{Last}(\alpha_1 \cap \alpha_2) &= \text{Last}(\alpha_1) \otimes \text{Last}(\alpha_2), \\ \text{Follow}(\alpha_1 \cap \alpha_2, (p_1, p_2)) &= \text{Follow}(\alpha_1, p_1) \otimes \text{Follow}(\alpha_2, p_2), \end{aligned} \tag{7}$$

where for $\circ \in \{+, \cdot, *\}$, each $f \in \{\text{First}, \text{Last}, \text{Follow}\}$ is defined as in Section 3.

Example 17. Consider $\alpha = (ba^*b + a) \cap (aa + b)^*$ from the previous example with $\bar{\alpha} = (b_1a_2^*b_3+a_4) \cap (a_5a_6+b_7)^*$. Then, $\text{First}(\alpha) = \{(1, 7), (4, 5)\}$, $\text{Last}(\alpha) = \{(3, 7), (4, 6)\}$, and $\text{Follow}(\alpha, (1, 7)) = \{(2, 5), (3, 7)\}$, $\text{Follow}(\alpha, (2, 5)) = \{(2, 6)\}$, $\text{Follow}(\alpha, (2, 6)) = \{(2, 5), (3, 7)\}$. The automaton $\mathcal{A}_{\text{POS}}(\alpha)$ is represented below.



In [17, 27] the states of the position automaton for $\alpha \in \mathcal{R}(\cap)$ are labelled by subsets $I \subseteq \text{Pos}(\alpha)$, where all positions correspond to the same letter. We recall that construction and show that the definitions above lead to an isomorphic automaton. The correctness of the location based construction follows as a consequence.

Let $\text{Ind}(\alpha)$ be the set of all non-empty subsets $I \subseteq \text{Pos}(\alpha)$, such that $\ell(I) = \sigma$ for some $\sigma \in \Sigma$. For $S_1, S_2 \subseteq \text{Ind}(\alpha)$, we consider

$$S_1 \otimes S_2 = \{ I_1 \cup I_2 \mid \ell(I_1) = \ell(I_2) \wedge I_1 \in S_1, I_2 \in S_2 \}.$$

The sets $\text{First}'(\alpha)$, $\text{Last}'(\alpha)$, and $\text{Follow}'(\alpha, I) \subseteq \text{Ind}(\alpha)$, for $I \in \text{Ind}(\alpha)$, are defined as in (3)-(5) for $\circ \in \{+, \cdot, *\}$ and for the base case and intersection, as follows

$$\begin{aligned} \text{First}'(\sigma_i) &= \text{Last}'(\sigma_i) = \{\{i\}\}, \\ \text{First}'(\alpha_1 \cap \alpha_2) &= \text{First}'(\alpha_1) \otimes \text{First}'(\alpha_2), \\ \text{Last}'(\alpha_1 \cap \alpha_2) &= \text{Last}'(\alpha_1) \otimes \text{Last}'(\alpha_2), \end{aligned}$$

and

$$\text{Follow}'(\alpha_1 \cap \alpha_2, I) = \text{Follow}'(\alpha_1, I_1) \otimes \text{Follow}'(\alpha_2, I_2),$$

if $I = I_1 \cup I_2$, $I_1 \in \text{Ind}(\alpha_1)$ and $I_2 \in \text{Ind}(\alpha_2)$; and $\text{Follow}'(\alpha_1 \cap \alpha_2, I) = \emptyset$, otherwise.

Finally, for $S \subseteq \text{Ind}(\alpha)$ and $\sigma \in \Sigma$ one has $\text{Select}(S, \sigma) = \{ I \in S \mid \ell(I) = \sigma \}$. With these definitions, the position automaton from [17, 27] is defined by

$$\mathcal{A}_{\text{POS}}^\cap(\alpha) = \langle \text{Ind}(\alpha) \cup \{\{0\}\}, \Sigma, \delta_{\text{POS}}, 0, \text{Last}'_0(\alpha) \rangle, \quad (8)$$

where $\text{Last}'_0(\alpha)$ is defined as before, $\delta_{\text{POS}}(I, \sigma) = \text{Select}(\text{Follow}'(\alpha, I), \sigma)$ and $\text{Follow}'(\alpha, \{0\}) = \text{First}'(\alpha)$ [27]. In the case of expressions containing intersection, and due to the fact that some subexpressions may describe the empty language, the construction of this automaton may include useless states, i.e., states with an empty right language.

Now, we establish the relation between locations and elements of $\text{Ind}(\alpha)$. It is easy to see that for every $p \in \text{First}(\alpha)$ (resp. $p \in \text{Last}(\alpha)$ or $p \in \text{Follow}(\alpha, p')$) one has $\text{l2pos}(p) \in \text{Ind}(\alpha)$. Consequently, $\text{l2pos}(\text{First}(\alpha)) \subseteq \text{Ind}(\alpha)$, and the same holds for Last and Follow . We extend the definition of the function p2loc to intersections by

$$\text{p2loc}(\alpha_1 \cap \alpha_2, I) = (\text{p2loc}(\alpha_1, I_1), \text{p2loc}(\alpha_2, I_2)),$$

if $I = I_1 \cup I_2 \in \text{Ind}(\alpha_1 \cap \alpha_2)$, $\emptyset \neq I_j \subseteq \text{Pos}(\alpha_j)$, for $j = 1, 2$.

The following lemma establishes a one-to-one correspondence between $f(\alpha)$ and $f'(\alpha)$, for $f \in \{\text{First}, \text{Last}, \text{Follow}\}$.

Lemma 16. *For all $\alpha \in \mathcal{R}(\cap)$ we have*

1. $\text{First}'(\alpha) = \text{l2pos}(\text{First}(\alpha));$
2. $\text{Last}'(\alpha) = \text{l2pos}(\text{Last}(\alpha));$
3. *For $I, I' \in \text{Ind}(\alpha)$ such that $I' \in \text{Follow}'(\alpha, I)$, $\text{p2loc}(\alpha, I') \in \text{Follow}(\text{p2loc}(\alpha, I));$*
4. *For $p, p' \in \text{Loc}(\alpha)$ such that $p' \in \text{Follow}(\alpha, p)$, $\text{l2pos}(p') \in \text{Follow}'(\alpha, \text{l2pos}(p)).$*

Proof. For 1. and 2. we need to prove that for $f \in \{\text{First}, \text{Last}\}$ we have

$$\begin{aligned} \forall p \in f(\alpha) \exists I \in f'(\alpha) \text{ such that } I = \text{l2pos}(p), \\ \forall I \in f'(\alpha) \exists p \in f(\alpha) \text{ such that } I = \text{l2pos}(p). \end{aligned}$$

For $\sigma_i \in \Sigma_{\bar{\alpha}}$, $i \in \text{First}(\sigma_i)$ if and only if $\text{l2pos}(i) = \{i\} \in \text{First}'(\sigma_i)$. The same holds for **Last**. Suppose that the result is valid for $\alpha_1, \alpha_2 \in \mathcal{R}(\cap)$. Then it is also valid for $\alpha_1 \circ \alpha_2$ for $\circ \in \{+, \cdot\}$ and for α_1^* . Let $p \in \text{First}(\alpha_1 \cap \alpha_2)$ with $p = (p_1, p_2)$, $p_j \in \text{First}(\alpha_j)$, $\ell(p_1) = \ell(p_2)$, and $\text{l2pos}(p_j) \in \text{First}'(\alpha_j)$, for $j \in \{1, 2\}$. Then, $\text{l2pos}(p_1) \cup \text{l2pos}(p_2) = \text{l2pos}((p_1, p_2)) = \text{l2pos}(p) \in \text{First}'(\alpha_1 \cap \alpha_2)$. On the other hand, let $I \in \text{First}'(\alpha_1 \cap \alpha_2)$ such that $I = I_1 \cup I_2$, $\ell(I) = \ell(I_1) = \ell(I_2)$, and $I_j \in \text{First}'(\alpha_j)$ for $j \in \{1, 2\}$. Then there exists $p_j \in \text{Loc}(\alpha_j)$ such that $I_j = \text{l2pos}(p_j) \in \text{First}'(\alpha_j)$ and $p_j \in \text{First}(\alpha_j)$, for $j \in \{1, 2\}$. We conclude that $I = \text{l2pos}(p_1) \cup \text{l2pos}(p_2) = \text{l2pos}((p_1, p_2))$, and $(p_1, p_2) \in \text{First}(\alpha_1 \cap \alpha_2)$. A similar proof holds for **Last**.

Next we consider the **Follow** function. We have $\text{Follow}'(\sigma_i) = \text{Follow}(\sigma_i) = \emptyset$, for $\sigma_i \in \Sigma_{\bar{\alpha}}$. Suppose that 3. and 4. hold for $\alpha_1, \alpha_2 \in \mathcal{R}(\cap)$. Then, the results are also true for $\alpha_1 \circ \alpha_2$ for $\circ \in \{+, \cdot\}$ and for α_1^* . We illustrate this fact considering 3. in the case where \circ is the concatenation operator and $\text{Follow}'(\alpha_1 \alpha_2, I) = \text{Follow}'(\alpha_1, I) \cup \text{First}'(\alpha_2)$, because $I \in \text{Last}'(\alpha_1)$. Let $I' \in \text{Follow}'(\alpha_1 \alpha_2, I)$, and $\ell(I') = \sigma$. Then, by 2. we have $\text{p2loc}(\alpha_1, I) \in \text{Last}(\alpha_1)$, and by the definition of **p2loc** it follows that $\text{p2loc}(I, \alpha_1) = \text{p2loc}(\alpha_1 \alpha_2, I)$. If $I' \in \text{First}'(\alpha_2)$ then $\text{p2loc}(\alpha_2, I') \in \text{First}(\alpha_2)$. Thus, we have $\text{p2loc}(\alpha_2, I') = \text{p2loc}(\alpha_1 \alpha_2, I')$ and $\text{p2loc}(\alpha_1 \alpha_2, I') \in \text{First}(\alpha_2) \subseteq \text{Follow}(\alpha_1 \alpha_2, \text{p2loc}(\alpha_1, I)) = \text{Follow}(\alpha_1 \alpha_2, \text{p2loc}(\alpha_1 \alpha_2, I))$. Finally, if $I' \in \text{Follow}'(\alpha_1, I)$ it follows from the induction hypothesis that

$$\begin{aligned} \text{p2loc}(\alpha_1, I') &= \text{p2loc}(\alpha_1 \alpha_2, I') \in \text{Follow}(\alpha_1, \text{p2loc}(\alpha_1 \alpha_2, I)) \\ &\subseteq \text{Follow}(\alpha_1 \alpha_2, \text{p2loc}(\alpha_1 \alpha_2, I)). \end{aligned}$$

If $I' \in \text{Follow}'(\alpha_1, I)$, we have $\text{p2loc}(\alpha_1, I') \in \text{Follow}(\alpha_1, \text{p2loc}(\alpha_1, I))$, by the induction. But $\text{p2loc}(\alpha_1, I) = \text{p2loc}(\alpha_1 \alpha_2, I)$ and $\text{p2loc}(\alpha_1, I') = \text{p2loc}(\alpha_1 \alpha_2, I')$, thus $\text{p2loc}(\alpha_1, I') \in \text{Follow}(\alpha_1 \alpha_2, \text{p2loc}(\alpha_1 \alpha_2, I))$.

Now, we consider the operator \cap and the case 3.. Let $I' \in \text{Follow}'(\alpha_1 \cap \alpha_2, I)$. Then $I = I_1 \cup I_2$, $I' = I'_1 \cup I'_2$, where $I_j, I'_j \in \text{Ind}(\alpha_j)$ and $I'_j \in \text{Follow}'(\alpha_j, I_j)$, for $j = 1, 2$. By the induction, we have $\text{p2loc}(\alpha_j, I'_j) \in \text{Follow}(\alpha_j, \text{p2loc}(\alpha_j, I_j))$, for $j = 1, 2$. Then

$$\begin{aligned} (\text{p2loc}(\alpha_1, I'_1), \text{p2loc}(\alpha_2, I'_2)) &\in \text{Follow}(\alpha_1 \cap \alpha_2, (\text{p2loc}(\alpha_1, I_1), \text{p2loc}(\alpha_2, I_2))) \iff \\ \text{p2loc}(\alpha_1 \cap \alpha_2, I') &\in \text{Follow}(\alpha_1 \cap \alpha_2, \text{p2loc}(\alpha_1 \cap \alpha_2, I)). \end{aligned}$$

The case 4. is proved in a similar manner. \square

From this lemma it is clear that the automaton defined in Equation (8) can be constructed using locations and the sets defined in Equation (7).

Corollary 2. For $\alpha \in \mathcal{R}(\cap)$, one has $\mathcal{A}_{\text{POS}}(\alpha) \simeq \mathcal{A}_{\text{POS}}^{\cap}(\alpha)$.

9. Conclusion

In this paper, locations were used to extend the position automaton construction to regular expressions with operations such as shuffle or intersection. Although we have presented the position automaton for $\mathcal{R}(\sqcup)$ and $\mathcal{R}(\cap)$ separately, one can easily consider a uniform construction for regular expressions extended with both \sqcup and \cap , i.e., $\mathcal{R}(\sqcup, \cap)$. For that, it is sufficient to consider the pair $(\ell(p), p)$, instead of only a location p , in the definition of the sets **First** and **Follow**, for expressions with intersection. The resulting construction allows to define automata for $\mathcal{R}(\sqcup, \cap)$, which is already implemented in the **FAdo** system. The extension of location based automata to other concurrency operators [7] is currently under research. In practical applications, certain classes of regular expressions with shuffle are used, such as deterministic and chain (e.g. [28]). It would be interesting to study the descriptive complexity of the automata here studied for those classes.

Acknowledgments

We thank the anonymous reviewers for their comments that helped to improve previous versions of this paper.

References

- [1] S. Broda, A. Machiavelo, N. Moreira, R. Reis, Location based automata for expressions with shuffle, in: A. Leporati, C. Martín-Vide, D. Shapira, C. Zandron (Eds.), Proc. 15th LATA 2021, Vol. 12638 of LNCS, Springer, 2021, pp. 43–54. doi:10.1007/978-3-030-68195-1_4.
- [2] V. Garg, M. Ragunath, Concurrent regular expressions and their relationship to Petri nets, Theoret. Comput. Sci. 96 (2) (1992) 285–304.
- [3] A. J. Mayer, L. J. Stockmeyer, Word problems-this time with interleaving, Inf. Comput. 115 (2) (1994) 293–311. doi:10.1006/inco.1994.1098.
- [4] E. Bárcenas, P. Genevès, N. Layaïda, A. Schmitt, Query reasoning on trees with types, interleaving, and counting, in: T. Walsh (Ed.), Proc. of the 22nd IJCAI, IJCAI/AAAI, 2011, pp. 718–723. doi:10.5591/978-1-57735-516-8/IJCAI11-127.
- [5] W. Gelade, W. Martens, F. Neven, Optimizing schema languages for XML: numerical constraints and interleaving, in: T. Schwentick, D. Suciu (Eds.), Proc. 11th ICDT, Vol. 4353 of LNCS, Springer, 2007, pp. 269–283. doi:10.1007/11965893_19.
- [6] 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.

- [7] M. Sulzmann, P. Thiemann, Derivatives for regular shuffle expressions, in: A. Dediu, E. Formenti, C. Martín-Vide, B. Truthe (Eds.), 9th LATA, Vol. 8977 of LNCS, Springer, 2015, pp. 275–286. doi:10.1007/978-3-319-15579-1.
- [8] B. D. Estrade, A. L. Perkins, J. M. Harris, Explicitly parallel regular expressions, in: J. Ni, J. Dongarra (Eds.), 1st IMSCCS, IEEE, 2006, pp. 402–409. doi:10.1109/IMSCCS.2006.60.
- [9] V. M. Glushkov, The abstract theory of automata, *Russ. Math. Surv.* 16 (1961) 1–53.
- [10] V. M. Antimirov, Partial derivatives of regular expressions and finite automaton constructions., *Theoret. Comput. Sci.* 155 (2) (1996) 291–319. doi:10.1016/0304-3975(95)00182-4.
- [11] H. Yamamoto, A new finite automaton construction for regular expressions, in: S. Bensch, R. Freund, F. Otto (Eds.), 6th NCMA, Vol. 304 of books@ocg.at, Österreichische Computer Gesellschaft, 2014, pp. 249–264.
- [12] E. Maia, N. Moreira, R. Reis, Prefix and right-partial derivative automata, in: M. Soskova, V. Mitrana (Eds.), 11th CiE, Vol. 9136 of LNCS, Springer, 2015, pp. 258–267. doi:10.1007/978-3-319-20028-6.
- [13] L. Ilie, S. Yu, Follow automata, *Inf. Comput.* 186 (1) (2003) 140–162. doi:10.1016/S0890-5401(03)00090-7.
- [14] A. Asperti, C. S. Coen, E. Tassi, Regular expressions, au point, *CoRR abs/1010.2604* (2010).
- [15] T. Nipkow, D. Traytel, Unified decision procedures for regular expression equivalence, in: G. Klein, R. Gamboa (Eds.), 5th ITP, Vol. 8558 of LNCS, Springer, 2014, pp. 450–466. doi:10.1007/978-3-319-08970-6-29.
- [16] S. Broda, M. Holzer, E. Maia, N. Moreira, R. Reis, A mesh of automata, *Inf. Comput.* 265 (2019) 94–111. doi:10.1016/j.ic.2019.01.003.
- [17] S. Broda, A. Machiavelo, N. Moreira, R. Reis, Position automata for semi-extended expressions, *J. Autom. Lang. Comb.* 23 (1–3) (2018) 39–65. doi:10.25596/jalc-2018-039.
- [18] A. Brüggemann-Klein, Regular expressions into finite automata, *Theoret. Comput. Sci.* 48 (1993) 197–213.
- [19] J. M. Champarnaud, D. Ziadi, Canonical derivatives, partial derivatives and finite automaton constructions, *Theoret. Comput. Sci.* 289 (2002) 137–163. doi:10.1016/S0304-3975(01)00267-5.

- [20] S. Broda, A. Machiavelo, N. Moreira, R. Reis, Analytic combinatorics and descriptive 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.
- [21] A. Khorsi, F. Ouardi, D. Ziadi, Fast equation automaton computation, J. Discrete Algorithms 6 (3) (2008) 433–448. doi:10.1016/j.jda.2007.10.003.
- [22] Project FAdo, tools for formal languages manipulation, <https://pypi.org/project/FAdo/> (Access date:1/1/2022).
- [23] H. G. Mairson, Generating words in a context-free language uniformly at random, Inf. Process. Lett. 49 (1994) 95–99.
- [24] W. G. Cochran, Sampling Techniques, 3rd Edition, John Wiley and Sons, 1977.
- [25] J. Brzozowski, Derivatives of regular expressions, J. ACM 11 (4) (1964) 481–494.
- [26] R. McNaughton, H. Yamada, Regular expressions and state graphs for automata, IEEE Trans. Comput. 9 (1960) 39–47.
- [27] S. Broda, E. Maia, N. Moreira, R. Reis, The prefix automaton, J. Autom. Lang. Comb. 26 (1-2) (2021) 17–53. doi:10.25596/jalc-2021-017.
- [28] F. Peng, H. Chen, X. Mou, Deterministic regular expressions with interleaving, in: M. Leucker, C. Rueda, F. D. Valencia (Eds.), Proc. 12th ICTAC, Vol. 9399 of LNCS, Springer, 2015, pp. 203–220. doi:10.1007/978-3-319-25150-9_13.

Appendix A. Some Proofs Omitted in the Main Text

The following lemma is used in the proof of Lemma 8, which states that the set of partial derivatives coincides with \mathbf{c} -continuations for marked expressions.

Lemma 17. *For $\alpha_1, \alpha_2 \in \mathcal{R}(\sqcup)$ and $w \neq \varepsilon$ the following hold.*

1. $\partial_w(\alpha_1)\alpha_2 \subseteq \partial_w(\alpha_1\alpha_2);$
2. $\partial_w(\alpha_1\alpha_2) \subseteq \partial_w(\alpha_1)\alpha_2 \cup \bigcup_{\substack{w=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_2);$
3. $\partial_w(\alpha_2) \subseteq \partial^+(\alpha_1\alpha_2);$
4. $\partial_w(\alpha_1)\alpha_1^* \subseteq \partial_w(\alpha_1^*);$
5. $\partial_w(\alpha_1^*) \subseteq \bigcup_{\substack{w=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_1)\alpha_1^*;$
6. $\partial_w(\alpha_1 \sqcup \alpha_2) = \bigcup_{w \in w_1 \sqcup w_2} \partial_{w_1}(\alpha_1) \sqcup \partial_{w_2}(\alpha_2).$

- Proof.* 1. By induction on the length of w . For σ we have $\partial_\sigma(\alpha_1)\alpha_2 \subseteq \partial_\sigma(\alpha_1)\alpha_2 \cup \varepsilon(\alpha_1)\partial_\sigma(\alpha_2) = \partial_\sigma(\alpha_1\alpha_2)$. For $w = w'\sigma$ we have $\partial_{w'\sigma}(\alpha_1)\alpha_2 = \partial_\sigma(\partial_{w'}(\alpha_1))\alpha_2 \subseteq \partial_\sigma(\partial_{w'}(\alpha_1)\alpha_2) \subseteq \partial_\sigma(\partial_{w'}(\alpha_1\alpha_2)) = \partial_{w'\sigma}(\alpha_1\alpha_2)$.
2. By induction on the length of w . For σ we have $\partial_\sigma(\alpha_1\alpha_2) = \partial_\sigma(\alpha_1)\alpha_2 \cup \varepsilon(\alpha_1)\partial_\sigma(\alpha_2) \subseteq \partial_\sigma(\alpha_1)\alpha_2 \cup \partial_\sigma(\alpha_2)$. For $w = w'\sigma$ we have $\partial_{w'\sigma}(\alpha_1\alpha_2) = \partial_\sigma(\partial_{w'}(\alpha_1\alpha_2)) \subseteq \partial_\sigma(\partial_{w'}(\alpha_1)\alpha_2 \cup \bigcup_{\substack{w'=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_2)) = \partial_\sigma(\partial_{w'}(\alpha_1)\alpha_2) \cup \bigcup_{\substack{w'=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2\sigma}(\alpha_2) \subseteq \partial_w(\alpha_1)\alpha_2 \cup \bigcup_{\substack{w=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_2)$.
3. First note that for every $\alpha \in \mathcal{R}(\sqcup)$ ($\alpha \neq \emptyset$) we have $\mathcal{L}(\alpha) \neq \emptyset$. For α_1 this implies that there is a word $w' \in \mathcal{L}(\alpha_1)$ and an expression $\alpha' \in \partial_{w'}(\alpha_1)$ such that $\varepsilon(\alpha') = \varepsilon$. Thus, $\partial_w(\alpha_2) \subseteq \partial_w(\alpha'\alpha_2) \subseteq \partial_w(\partial_{w'}(\alpha_1)\alpha_2) \subseteq \partial_w(\partial_{w'}(\alpha_1\alpha_2)) = \partial_{w'\sigma}(\alpha_1\alpha_2) \subseteq \partial^+(\alpha_1\alpha_2)$.
4. By induction on the length of w . For σ we have $\partial_\sigma(\alpha_1)\alpha_1^* = \partial_\sigma(\alpha_1^*)$. For $w = w'\sigma$ we have $\partial_{w'\sigma}(\alpha_1)\alpha_1^* = \partial_\sigma(\partial_{w'}(\alpha_1))\alpha_1^* \subseteq \partial_\sigma(\partial_{w'}(\alpha_1)\alpha_1^*) \subseteq \partial_\sigma(\partial_{w'}(\alpha_1^*)) = \partial_{w'\sigma}(\alpha_1^*)$.
5. By induction on the length of w . For σ we have $\partial_\sigma(\alpha_1^*) = \partial_\sigma(\alpha_1)\alpha_1^*$. For $w = w'\sigma$ we have $\partial_{w'\sigma}(\alpha_1^*) = \partial_\sigma(\partial_{w'}(\alpha_1^*)) \subseteq \partial_\sigma(\bigcup_{\substack{w'=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_1)\alpha_1^*) = \bigcup_{\substack{w'=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_\sigma(\partial_{w_2}(\alpha_1)\alpha_1^*) \subseteq \bigcup_{\substack{w'=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2\sigma}(\alpha_1)\alpha_1^* \cup \partial_\sigma(\alpha_1)\alpha_1^* = \bigcup_{\substack{w=w_1w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_1)\alpha_1^*$.
6. By induction on the length of w . If $w = \sigma$ possible values for w_1 and w_2 are $w_1 = \sigma$ and $w_2 = \varepsilon$, or vice-versa. But $\partial_\sigma(\alpha_1 \sqcup \alpha_2) = \partial_\sigma(\alpha_1) \sqcup \partial_\varepsilon(\alpha_2) \cup \partial_\varepsilon(\alpha_1) \sqcup \partial_\sigma(\alpha_2)$. If $w = w'\sigma$, then $\partial_{w'\sigma}(\alpha_1 \sqcup \alpha_2) = \partial_\sigma(\partial_{w'}(\alpha_1 \sqcup \alpha_2)) = \partial_\sigma(\bigcup_{w' \in w'_1 \sqcup w'_2} (\partial_{w'_1}(\alpha_1) \sqcup \partial_{w'_2}(\alpha_2))) = \bigcup_{w' \in w'_1 \sqcup w'_2} (\partial_{w'_1\sigma}(\alpha_1) \sqcup \partial_{w'_2\sigma}(\alpha_2)) \cup \partial_{w'_1}(\alpha_1) \sqcup \partial_{w'_2\sigma}(\alpha_2) = \bigcup_{w' \in w'_1 \sqcup w'_2} (\partial_{w'_1\sigma}(\alpha_1) \sqcup \partial_{w'_2\sigma}(\alpha_2)) = \bigcup_{w' \in w'_1 \sqcup w'_2} (\partial_{w'_1\sigma}(\alpha_1) \sqcup \partial_{w'_2\sigma}(\alpha_2))$.

□

Lemma 8. *Let $\alpha \in \mathcal{R}(\sqcup)$. Then, $\partial^+(\bar{\alpha}) = \{c(\bar{\alpha}, p) \mid p \in \text{Loc}(\alpha)\}$.*

Proof. First, we prove by structural induction on $\bar{\alpha}$ that $p \in \text{Loc}(\alpha) = \text{Loc}(\bar{\alpha})$ implies that $c(\bar{\alpha}, p) \in \partial^+(\bar{\alpha})$. During the proof we just write α and suppose that α is already marked. The result is trivially true for $\alpha = \emptyset$. For $p = i \in \text{Loc}(\sigma_i)$ we have

$$c(\sigma_i, i) = \varepsilon \in \partial_{\sigma_i}(\sigma_i).$$

Next consider an expression $\alpha_1 + \alpha_2$ and $p \in \text{Loc}(\alpha_1)$ (the case of $p \in \text{Loc}(\alpha_2)$ is analogous). We have

$$c(\alpha_1 + \alpha_2, p) = c(\alpha_1, p) \in \partial^+(\alpha_1).$$

Thus, there is a word $w \neq \varepsilon$ such that

$$c(\alpha_1, p) \in \partial_w(\alpha_1) \subseteq \partial_w(\alpha_1 + \alpha_2) \subseteq \partial^+(\alpha_1 + \alpha_2).$$

Next we consider an expression $\alpha_1\alpha_2$ and $p \in \text{Loc}(\alpha_1)$. Then, $c(\alpha_1\alpha_2, p) = c(\alpha_1, p)\alpha_2$, where by induction $c(\alpha_1, p) \in \partial^+(\alpha_1)$. Thus, there is a word $w \neq \varepsilon$ such that $c(\alpha_1, p) \in \partial_w(\alpha_1)$. By Lemma 17,

$$c(\alpha_1, p)\alpha_2 \in \partial_w(\alpha_1)\alpha_2 \subseteq \partial_w(\alpha_1\alpha_2) \subseteq \partial^+(\alpha_1\alpha_2).$$

Next, consider $\alpha_1\alpha_2$ and $p \in \text{Loc}(\alpha_2)$. Then, $c(\alpha_1\alpha_2, p) = c(\alpha_2, p) \in \partial^+(\alpha_2)$. There is a word $w \neq \varepsilon$ such that $c(\alpha_2, p) \in \partial_w(\alpha_2)$. But, by Lemma 17, $\partial_w(\alpha_2) \subseteq \partial^+(\alpha_1\alpha_2)$.

Now, consider an expression of the form α_1^* and $p \in \text{Loc}(\alpha_1)$. We have $c(\alpha_1^*, p) = c(\alpha_1, p)\alpha_1^*$. By the induction hypothesis $c(\alpha_1, p) \in \partial^+(\alpha_1)$. Thus, there is some word $w \neq \varepsilon$ such that $c(\alpha_1, p) \in \partial_w(\alpha_1)$. It follows from Lemma 17 that

$$c(\alpha_1, p)\alpha_1^* \in \partial_w(\alpha_1)\alpha_1^* \subseteq \partial_w(\alpha_1^*) \subseteq \partial^+(\alpha_1^*).$$

Finally, consider an expression of the form $\alpha_1 \sqcup \alpha_2$. If $p = (p_1, 0)$ with $p_1 \in \text{Loc}(\alpha_1)$, then by induction $c(\alpha_1, p_1) \in \partial^+(\alpha_1)$, i.e., there is a word $w \neq \varepsilon$ such that $c(\alpha_1, p_1) \in \partial_w(\alpha_1)$. Thus, by Lemma 17,

$$\begin{aligned} c(\alpha_1 \sqcup \alpha_2, (p_1, 0)) &= c(\alpha_1, p_1) \sqcup \alpha_2 \in \partial_w(\alpha_1) \sqcup \alpha_2 = \partial_w(\alpha_1) \sqcup \partial_\varepsilon(\alpha_2) \\ &\subseteq \partial_w(\alpha_1 \sqcup \alpha_2) \subseteq \partial^+(\alpha_1 \sqcup \alpha_2). \end{aligned}$$

The case of $p = (0, p_2)$ with $p_2 \in \text{Loc}(\alpha_2)$ is analogous. If $p = (p_1, p_2)$ with $p_1 \in \text{Loc}(\alpha_1)$ and $p_2 \in \text{Loc}(\alpha_2)$, then by induction $c(\alpha_1, p_1) \in \partial^+(\alpha_1)$ and $c(\alpha_2, p_2) \in \partial^+(\alpha_2)$, i.e., there are words $w_1, w_2 \neq \varepsilon$ such that $c(\alpha_1, p_1) \in \partial_{w_1}(\alpha_1)$ and $c(\alpha_2, p_2) \in \partial_{w_2}(\alpha_2)$. Thus, by Lemma 17,

$$\begin{aligned} c(\alpha_1 \sqcup \alpha_2, (p_1, p_2)) &= c(\alpha_1, p_1) \sqcup c(\alpha_2, p_2) \in \partial_{w_1}(\alpha_1) \sqcup \partial_{w_2}(\alpha_2) \\ &\subseteq \partial_{w_1 w_2}(\alpha_1 \sqcup \alpha_2) \subseteq \partial^+(\alpha_1 \sqcup \alpha_2). \end{aligned}$$

In the second part of the proof we show by structural induction on α that $\beta \in \partial^+(\alpha)$ implies that there is $p \in \text{Loc}(\alpha)$ such that $\beta = c(\alpha, p)$.

The result is trivially true for $\alpha = \emptyset$. For $\alpha = \sigma_i$, we have $\partial^+(\sigma_i) = \{\varepsilon\}$, $\text{Loc}(\sigma_i) = \{i\}$ and $c(\sigma_i, i) = \varepsilon$.

Next, consider an expression $\alpha_1 + \alpha_2$ and $\beta \in \partial^+(\alpha_1 + \alpha_2) = \partial^+(\alpha_1) \cup \partial^+(\alpha_2)$. If $\beta \in \partial^+(\alpha_1)$, then there is $p \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1 + \alpha_2)$ such that $\beta = c(\alpha_1, p) = c(\alpha_1 + \alpha_2, p)$.

Next we consider an expression $\alpha_1\alpha_2$ and $\beta \in \partial^+(\alpha_1\alpha_2)$. There is a word $w \neq \varepsilon$ such that

$$\beta \in \partial_w(\alpha_1\alpha_2) \subseteq \partial_w(\alpha_1)\alpha_2 \cup \bigcup_{\substack{w=w_1 w_2 \\ w_2 \neq \varepsilon}} \partial_{w_2}(\alpha_2).$$

If $\beta \in \partial_w(\alpha_1)\alpha_2$, then $\beta = \beta_1\alpha_2$ with $\beta_1 \in \partial_w(\alpha_1)$. By induction, there is $p \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1\alpha_2)$ such that $\beta_1 = c(\alpha_1, p)$. Thus,

$$\beta = \beta_1\alpha_2 = c(\alpha_1, p)\alpha_2 = c(\alpha_1\alpha_2, p).$$

If $\beta \in \partial_{w_2}(\alpha_2)$ for some suffix $w_2 \neq \varepsilon$ of w , then there is some $p \in \text{Loc}(\alpha_2) \subseteq \text{Loc}(\alpha_1\alpha_2)$ such that $\beta = c(\alpha_2, p) = c(\alpha_1\alpha_2, p)$.

Now, consider an expression of the form α_1^* and $\beta \in \partial_w(\alpha_1^*) \subseteq \partial^+(\alpha_1^*)$. By Lemma 17 there is some suffix $w_2 \neq \varepsilon$ of w such that $\beta \in \partial_{w_2}(\alpha_1)\alpha_1^*$. Thus, $\beta = \beta_1\alpha_1^*$ with $\beta_1 \in \partial_{w_2}(\alpha_1)$. By induction, there is $p \in \text{Loc}(\alpha_1) = \text{Loc}(\alpha_1^*)$ such that $\beta = c(\alpha_1, p)\alpha_1^* = c(\alpha_1^*, p)$.

Finally, consider an expression of the form $\alpha_1 \sqcup \alpha_2$ and $\beta \in \partial_w(\alpha_1 \sqcup \alpha_2)$, for some $w \neq \varepsilon$. Then, $\beta = \beta_1 \sqcup \beta_2$, where $\beta_1 \in \partial_{w_1}(\alpha_1)$ and $\beta_2 \in \partial_{w_2}(\alpha_2)$, for some w_1, w_2 such that $w \in w_1 \sqcup w_2$. If both $w_1, w_2 \neq \varepsilon$, then there is $p_i \in \text{Loc}(\alpha_i)$ such that $\beta_i = c(\alpha_i, p_i)$ for $i = 1, 2$. Thus,

$$\beta = \beta_1 \sqcup \beta_2 = c(\alpha_1, p_1) \sqcup c(\alpha_2, p_2) = c(\alpha_1 \sqcup \alpha_2, (p_1, p_2))$$

and $(p_1, p_2) \in \text{Loc}(\alpha_1 \sqcup \alpha_2)$. If $w = w_1$ and $w_2 = \varepsilon$, then $\beta_1 \in \partial_w(\alpha_1)$ and $\beta_2 = \alpha_2$. By induction, there is $p_1 \in \text{Loc}(\alpha_1)$ such that $\beta_1 = c(\alpha_1, p_1)$. Consequently, $(p_1, 0) \in \text{Loc}(\alpha_1 \sqcup \alpha_2)$ and

$$\beta = \beta_1 \sqcup \alpha_2 = c(\alpha_1, p_1) \sqcup \alpha_2 = c(\alpha_1 \sqcup \alpha_2, (p_1, 0)).$$

□

Proposition 10. For $\alpha \in \mathcal{R}(\sqcup)$, $p \in \text{Loc}_0(\alpha)$, and $\sigma_i \in \Sigma_{\bar{\alpha}}$, one has

$$\beta \in \partial_{\sigma_i}(c(\bar{\alpha}, p)) \iff \exists q \in \text{Loc}(\alpha) : \beta = c(\bar{\alpha}, q) \wedge i \in \text{l2pos}(q) \wedge (\bar{\sigma}_i, q) \in \text{Follow}(\alpha, p).$$

Proof. The proof is by structural induction on $\bar{\alpha}$.

(\Rightarrow) The result is trivially true for ε , and also for σ_i and $p = i$, for which $\partial_{\sigma_i}(c(\sigma_i, i)) = \partial_{\sigma_i}(\varepsilon) = \emptyset$. If $p = 0$, we have $\partial_{\sigma_i}(c(\sigma_i, 0)) = \partial_{\sigma_i}(\sigma_i) = \{\varepsilon\}$, i.e., $\beta = \varepsilon = c(\sigma_i, i)$. Also, $i \in \text{Loc}(\sigma_i)$, $(\sigma, i) \in \text{Follow}(\sigma_i, 0)$, and $\sigma = \bar{\sigma}_i$.

Now, we consider an expression of the form $\alpha_1 + \alpha_2$ and $p \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1 + \alpha_2)$. If $\beta \in \partial_{\sigma_i}(c(\alpha_1 + \alpha_2, p)) = \partial_{\sigma_i}(c(\alpha_1, p))$, then by the induction hypothesis there exists $q \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1 + \alpha_2)$ such that $\beta = c(\alpha_1, q) = c(\alpha_1 + \alpha_2, q)$, $i \in \text{l2pos}(q)$, and $(\bar{\sigma}_i, q) \in \text{Follow}(\alpha_1, p) = \text{Follow}(\alpha_1 + \alpha_2, p)$. The case of $p \in \text{Loc}(\alpha_2)$ is analogous. Finally, for $p = 0$ we have that

$$\beta \in \partial_{\sigma_i}(\alpha_1 + \alpha_2) = \partial_{\sigma_i}(\alpha_1) \cup \partial_{\sigma_i}(\alpha_2) = \partial_{\sigma_i}(c(\alpha_1, 0)) \cup \partial_{\sigma_i}(c(\alpha_2, 0)).$$

Suppose that $\beta \in \partial_{\sigma_i}(c(\alpha_1, 0))$. Then, by induction there exists $q \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1 + \alpha_2)$ such that $\beta = c(\alpha_1, q)$, $i \in \text{l2pos}(q)$, and

$$(\bar{\sigma}_i, q) \in \text{Follow}(\alpha_1, 0) = \text{First}(\alpha_1) \subseteq \text{First}(\alpha_1 + \alpha_2) = \text{Follow}(\alpha_1 + \alpha_2, 0).$$

For the remaining operators the proof for the subcase of $p = 0$ is similar to the others and will be omitted.

Next, we consider an expression of the form $\alpha_1\alpha_2$ and $p \in \text{Loc}(\alpha_1\alpha_2) = \text{Loc}(\alpha_1) \cup \text{Loc}(\alpha_2)$. First, suppose that $p \in \text{Loc}(\alpha_1)$ and

$$\beta \in \partial_{\sigma_i}(c(\alpha_1\alpha_2, p)) = \partial_{\sigma_i}(c(\alpha_1, p)\alpha_2) = \partial_{\sigma_i}(c(\alpha_1, p))\alpha_2 \cup \varepsilon(c(\alpha_1, p))\partial_{\sigma_i}(\alpha_2).$$

If $\beta \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p))\alpha_2$, then $\beta = \beta_1\alpha_2$ with $\beta_1 \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p))$. By induction, there is some $q \in \text{Loc}(\alpha_1)$ and $i \in \text{l2pos}(q)$ such that

$$\beta_1 = \mathbf{c}(\alpha_1, q), \text{ and } (\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1, p).$$

Thus, there is some $q \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1\alpha_2)$ such that $i \in \text{l2pos}(q)$ and

$$\beta_1\alpha_2 = \mathbf{c}(\alpha_1, q)\alpha_2 = \mathbf{c}(\alpha_1\alpha_2, q) \text{ and } (\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1, p) \subseteq \text{Follow}(\alpha_1\alpha_2, p).$$

If $\beta \in \varepsilon(\mathbf{c}(\alpha_1, p))\partial_{\sigma_i}(\alpha_2) = \partial_{\sigma_i}(\alpha_2)$, then $p \in \text{Last}(\alpha_1)$ by Lemma 9. It follows by induction from $\beta \in \partial_{\sigma_i}(\alpha_2) = \partial_{\sigma_i}(\mathbf{c}(\alpha_2, 0))$ that there is some $q \in \text{Loc}(\alpha_2) \subseteq \text{Loc}(\alpha_1\alpha_2)$ with $i \in \text{l2pos}(q)$, $\beta = \mathbf{c}(\alpha_2, q)$ and $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_2, 0) = \text{First}(\alpha_2)$. Thus, $\beta = \mathbf{c}(\alpha_2, q) = \mathbf{c}(\alpha_1\alpha_2, q)$ and $(\overline{\sigma_i}, q) \in \text{First}(\alpha_2) \subseteq \text{Follow}(\alpha_1\alpha_2, p)$ (this last inclusion follows from $p \in \text{Last}(\alpha_1)$).

Next, suppose that $p \in \text{Loc}(\alpha_2)$ and $\beta \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1\alpha_2, p)) = \partial_{\sigma_i}(\mathbf{c}(\alpha_2, p))$. By induction there exists $q \in \text{Loc}(\alpha_2) \subseteq \text{Loc}(\alpha_1\alpha_2)$ such that $i \in \text{l2pos}(q)$, $\beta = \mathbf{c}(\alpha_2, q) = \mathbf{c}(\alpha_1\alpha_2, q)$ and $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_2, p) = \text{Follow}(\alpha_1\alpha_2, p)$.

Now consider an expression α_1^* and that

$$\beta \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1^*, p)) = \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p)\alpha_1^*) = \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p))\alpha_1^* \cup \varepsilon(\mathbf{c}(\alpha_1, p))\partial_{\sigma_i}(\alpha_1)\alpha_1^*.$$

If $\beta = \beta_1\alpha_1^*$ and $\beta_1 \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p))$, then by induction there is some $q \in \text{Loc}(\alpha_1) = \text{Loc}(\alpha_1^*)$ such that $i \in \text{l2pos}(q)$, $\beta_1 = \mathbf{c}(\alpha_1, q)$ and $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1, p)$. Consequently, $\beta = \beta_1\alpha_1^* = \mathbf{c}(\alpha_1, q)\alpha_1^* = \mathbf{c}(\alpha_1^*, q)$ and

$$(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1, p) \subseteq \text{Follow}(\alpha_1^*, p).$$

Now, suppose that $\beta \in \varepsilon(\mathbf{c}(\alpha_1, p))\partial_{\sigma_i}(\alpha_1)\alpha_1^*$. Then, $p \in \text{Last}(\alpha_1)$ and $\beta = \beta_1\alpha_1^*$ with $\beta_1 \in \partial_{\sigma_i}(\alpha_1) = \partial_{\sigma_i}(\mathbf{c}(\alpha_1, 0))$. By induction there exists $q \in \text{Loc}(\alpha_1) = \text{Loc}(\alpha_1^*)$ with $i \in \text{l2pos}(q)$, $\beta_1 = \mathbf{c}(\alpha_1, q)$ and, consequently, $\beta = \mathbf{c}(\alpha_1^*, q)$, and $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1, 0) = \text{First}(\alpha_1) \subseteq \text{Follow}(\alpha_1^*, p)$.

Finally, consider an expression of the form $\alpha_1 \sqcup \alpha_2$ and $p = (p_1, p_2)$, such that

$$\begin{aligned} \beta \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1 \sqcup \alpha_2, (p_1, p_2))) &= \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p_1) \sqcup \mathbf{c}(\alpha_2, p_2)) \\ &= \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p_1)) \sqcup \mathbf{c}(\alpha_2, p_2) \cup \mathbf{c}(\alpha_1, p_1) \sqcup \partial_{\sigma_i}(\mathbf{c}(\alpha_2, p_2)). \end{aligned}$$

If $\beta \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p_1)) \sqcup \mathbf{c}(\alpha_2, p_2)$, then $\beta = \beta_1 \sqcup \mathbf{c}(\alpha_2, p_2)$ with $\beta_1 \in \partial_{\sigma_i}(\mathbf{c}(\alpha_1, p_1))$. By induction, there exists $q_1 \in \text{Loc}(\alpha_1)$ such that $i \in \text{l2pos}(q_1)$, $\beta_1 = \mathbf{c}(\alpha_1, q_1)$ and $(\overline{\sigma_i}, q_1) \in \text{Follow}(\alpha_1, p_1)$. We have $(q_1, p_2) \in \text{Loc}(\alpha_1 \sqcup \alpha_2)$,

$$\beta = \mathbf{c}(\alpha_1, q_1) \sqcup \mathbf{c}(\alpha_2, p_2) = \mathbf{c}(\alpha_1 \sqcup \alpha_2, (q_1, p_2)),$$

and $(\overline{\sigma_i}, (q_1, p_2)) \in \text{Follow}(\alpha_1 \sqcup \alpha_2, (p_1, p_2))$. The case of $\beta \in \mathbf{c}(\alpha_1, p_1) \sqcup \partial_{\sigma_i}(\mathbf{c}(\alpha_2, p_2))$ is analogous.

(\Leftarrow) The result is trivially true for ε and for σ_i and $p = i$, for which $\text{Follow}(\sigma_i, i) = \emptyset$. For σ_i and $p = 0$, we have $\text{Follow}(\sigma_i, 0) = \text{First}(\sigma_i) = \{(\overline{\sigma_i}, i)\}$. Thus,

$$\beta = \mathbf{c}(\sigma_i, i) = \varepsilon \in \partial_{\sigma_i}(\sigma_i) = \partial_{\sigma_i}(\mathbf{c}(\sigma_i, 0)).$$

Next consider $\alpha_1 + \alpha_2$ and suppose that $p \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1 + \alpha_2)$. Suppose there is some $q \in \text{Loc}(\alpha_1 + \alpha_2) = \text{Loc}(\alpha_1) \cup \text{Loc}(\alpha_2)$, such that $i \in \text{I2pos}(q)$, $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1 + \alpha_2, p) = \text{Follow}(\alpha_1, p)$ and $\beta = c(\alpha_1 + \alpha_2, q)$. We conclude that $q \in \text{Loc}(\alpha_1)$ and consequently $\beta = c(\alpha_1 + \alpha_2, q) = c(\alpha_1, p)$. Then, by induction $\beta \in \partial_{\sigma_i}(c(\alpha_1, p)) = \partial_{\sigma_i}(c(\alpha_1 + \alpha_2, p))$. Again, the case of $p = 0$ is straightforward and will be omitted here and also for the other operators.

Now, consider $\alpha_1 \alpha_2$ with $p \in \text{Loc}(\alpha_1) \subseteq \text{Loc}(\alpha_1 \alpha_2)$. Suppose there is some $q \in \text{Loc}(\alpha_1 \alpha_2) = \text{Loc}(\alpha_1) \cup \text{Loc}(\alpha_2)$, such that $i \in \text{I2pos}(q)$, $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1 \alpha_2, p)$. If $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1, p)$, then $q \in \text{Loc}(\alpha_1)$ and $\beta = c(\alpha_1 \alpha_2, q) = c(\alpha_1, q) \alpha_2 = \beta_1 \alpha_2$. By induction, $\beta_1 \in \partial_{\sigma_i}(c(\alpha_1, p))$. Thus,

$$\beta = \beta_1 \alpha_2 \in \partial_{\sigma_i}(c(\alpha_1, p)) \alpha_2 \subseteq \partial_{\sigma_i}(c(\alpha_1, p) \alpha_2).$$

Otherwise, $p \in \text{Last}(\alpha_1)$ and $(\overline{\sigma_i}, q) \in \text{First}(\alpha_2) = \text{Follow}(\alpha_2, 0)$. Thus, $q \in \text{Loc}(\alpha_2)$, $i \in \text{I2pos}(q)$, and $\beta = c(\alpha_2, q)$. By induction,

$$\beta \in \partial_{\sigma_i}(c(\alpha_2, 0)) = \partial_{\sigma_i}(\alpha_2) \subseteq \partial_{\sigma_i}(c(\alpha_1, p) \alpha_2) = \partial_{\sigma_i}(c(\alpha_1 \alpha_2, p)),$$

where the inclusion follows from Lemma 9. The remaining case of $p \in \text{Loc}(\alpha_2)$ follows directly from the definitions and induction.

The proof for α_1^* follows the structure of the case for $\alpha_1 \alpha_2$ with $p \in \text{Loc}(\alpha_1)$, considering both $p \in \text{Last}(\alpha_1)$ and $p \notin \text{Last}(\alpha_1)$.

Finally, consider an expression of the form $\alpha_1 \sqcup \alpha_2$ and $p = (p_1, p_2) \in \text{Loc}(\alpha_1 \sqcup \alpha_2)$, i.e. $\alpha_i \in \text{Loc}_0(\alpha_i)$ for $i = 1, 2$. Furthermore, suppose that there exists $q \in \text{Loc}(\alpha_1 \sqcup \alpha_2)$, such that $i \in \text{I2pos}(q)$, $(\overline{\sigma_i}, q) \in \text{Follow}(\alpha_1 \sqcup \alpha_2, (p_1, p_2))$. Let $q = (p_1, q_2)$ (the case of $q = (q_1, p_2)$ is identical), $i \in \text{I2pos}(q_2)$, $(\overline{\sigma_i}, q_2) \in \text{Follow}(\alpha_2, p_2)$, and

$$\beta = c(\alpha_1 \sqcup \alpha_2, (p_1, q_2)) = c(\alpha_1, p_1) \sqcup c(\alpha_2, q_2).$$

By induction, $\beta_2 = c(\alpha_2, q_2) \in \partial_{\sigma_i}(c(\alpha_2, p_2))$. Thus,

$$\begin{aligned} \beta &= c(\alpha_1, p_1) \sqcup \beta_2 \in c(\alpha_1, p_1) \sqcup \partial_{\sigma_i}(c(\alpha_2, p_2)) \\ &\subseteq \partial_{\sigma_i}(c(\alpha_1, p_1) \sqcup c(\alpha_2, p_2)) = \partial_{\sigma_i}(c(\alpha_1 \sqcup \alpha_2, p)). \end{aligned}$$

□