# A Mesh of Automata

Sabine Broda[a], Markus Holzer[b], Eva Maia[a], Nelma Moreira[a,*], Rogério Reis[a]

[a] *CMUP & DCC, Faculdade de Ciências da Universidade do Porto*
*Rua do Campo Alegre, 4169-007 Porto, Portugal*
[b] *Institut für Informatik, Universität Giessen,*
*Arndtstr. 2, 35392 Giessen, Germany*

**Abstract**

We contribute new relations to the taxonomy of different conversions from regular expressions to equivalent finite automata. In particular, we are interested in transformations that construct automata such as, the follow automaton, the partial derivative automaton, the prefix automaton, the automata based on pointed expressions recently introduced and studied, and last but not least the position, or Glushkov automaton ($\mathcal{A}_{\mathrm{POS}}$), and their double reversed construction counterparts. We deepen the understanding of these constructions and show that with the artefacts used to construct the Glushkov automaton one is able to capture most of them. As a byproduct we define a *dual* version $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ of the position automaton which plays a similar role as $\mathcal{A}_{\mathrm{POS}}$ but now for the reverse expression. Moreover, it turns out that the prefix automaton $\mathcal{A}_{\mathrm{Pre}}$ is central to reverse expressions, because the determinisation of the double reversal of $\mathcal{A}_{\mathrm{Pre}}$ (first reverse the expression, construct the automaton $\mathcal{A}_{\mathrm{Pre}}$, and then reverse the automaton) can be represented as a quotient of any of the considered deterministic automata that we consider in this investigation. This shows that although the conversion of regular expressions and reversal of regular expressions to finite automata seems quite similar, there are significant differences.

*Keywords:* Regular Expressions, Finite Automata, Regular Languages

[☆] This is a completely revised and expanded version of a paper presented at the 21st International Conference on Developments in Language Theory (DLT) held in Liège, Belgium, August 7–11, 2017, [5].

[☆☆] This work was partially supported by CMUP (UID/MAT/00144/2019), which is funded by FCT (Portugal) with national (MEC) and European structural funds through the programs FEDER, under the partnership agreement PT2020.

[*] Corresponding author

*Email addresses:* sbb@dcc.fc.up.pt (Sabine Broda), holzer@informatik.uni-giessen.de (Markus Holzer), emaia@dcc.fc.up.pt (Eva Maia), nam@dcc.fc.up.pt (Nelma Moreira), rvr@dcc.fc.up.pt (Rogério Reis)

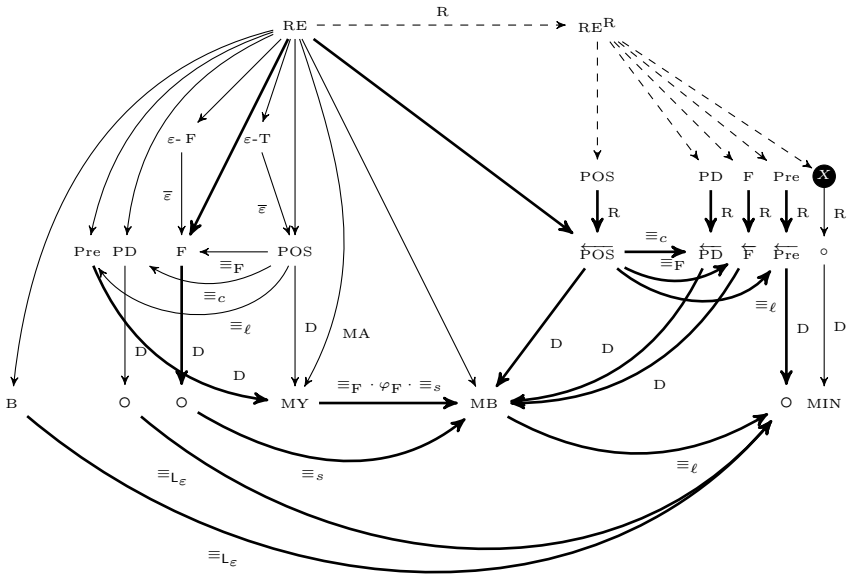



Figure 1: Taxonomy of conversions from regular expressions to finite automata. Bold arrows correspond to relations studied in this work. Dashed arrows correspond to constructions for the reversed language. Here $X$ may be any construction that yields a DFA. For the readability of the diagram we omit the arrows between RE and $\overleftarrow{\mathrm{F}}$, $\overleftarrow{\mathrm{PD}}$ and $\overleftarrow{\mathrm{Pre}}$, respectively. Brzozowski [9] showed that for a trim NFA $\mathcal{A}$, the automaton $\mathrm{D}(\mathcal{A})$ is minimal if $\mathcal{A}^{\mathrm{R}}$ is deterministic.

## 1. Introduction

It is well known that regular expressions define exactly the same languages as deterministic or nondeterministic finite automata. The conversion between these representations has been intensively studied for more than half a century—see, e.g., Gruber and Holzer [15] for a recent survey on this subject w.r.t. descriptional complexity. There are a few classical algorithms and variants thereof for converting finite automata into equivalent regular expressions and as shown in [23] all these approaches are more or less reformulations of the same underlying algorithmic idea, and they yield (almost) the same regular expressions. For the converse transformation, that is, the conversion of regular expressions into equivalent finite automata, the situation is much more diverse, since the algorithmic underlying ideas already are different. Nevertheless, for some of the algorithms the constructed automata can still be related to each other by determinisation and/or quotients w.r.t. equivalence relations. See Figure 1 for a taxonomy of conversions from regular expression to finite automata, where arrows, that are displayed in bold correspond to new contributions in [5] and in this paper. The two top nodes correspond to regular expressions. Each other node corresponds to a particular automaton, up to isomorphism, and edges between

two nodes represent transformation algorithms, such as epsilon elimination ($\overline{\varepsilon}$) determinisation (D), reversal (R), quotient by some equivalence relation, or a specific construction. Different nodes represent objects for which there is some witness that distinguishes them. Provenance of results that are not original is well indicated. An example how to read the taxonomy is, for instance, the result of Ilie and Yu [16] that shows that for a regular expression $\alpha$ the follow automaton $\mathcal{A}_{\mathrm{F}}(\alpha)$ is isomorphic ($\simeq$) to the quotient of the position or Glushkov [14] automaton $\mathcal{A}_{\mathrm{POS}}(\alpha)$ w.r.t. the relation $\equiv_{\mathrm{F}}$, that is, $\mathcal{A}_{\mathrm{F}}(\alpha) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha)/\equiv_{\mathrm{F}}$.

We briefly summarise our contributions: besides the above mentioned follow automaton $\mathcal{A}_{\mathrm{F}}$ we also consider the partial derivative automaton $\mathcal{A}_{\mathrm{PD}}$ of Mirkin [20] and Antimirov [2], the prefix automaton $\mathcal{A}_{\mathrm{Pre}}$ of Yamamoto [25], and contructions based on a recent approach of Asperti et al. [3] and by Nipkow and Traytel [21] by pointed expressions that lead to the mark after and mark before automata $\mathcal{A}_{\mathrm{MA}}$ and $\mathcal{A}_{\mathrm{MB}}$, respectively. Pointed expressions are an alternative representation of sets of positions, which are used in the construction of $\mathcal{A}_{\mathrm{POS}}$. For the follow automaton $\mathcal{A}_{\mathrm{F}}$ we show that it can be directly computed from the expression by labelling states not with positions but with their Follow sets and their finality. We also prove that the quotient of the determinised follow automaton w.r.t. a right-invariant relation $\equiv_s$, which is a generalisation of the $\equiv_{\mathrm{F}}$-relation, leads to the mark before automaton $\mathcal{A}_{\mathrm{MB}}$. It is known that $\mathcal{A}_{\mathrm{MA}}$ is isomorphic to the McNaughton-Yamada automaton $\mathcal{A}_{\mathrm{MY}}$. After observing that the determinisation of a quotient of an automaton, by a left-invariant relation, is isomorphic to the determinisation of the original automaton, we conclude that the determinisation of the prefix automaton $\mathcal{A}_{\mathrm{Pre}}$ is isomorphic to $\mathcal{A}_{\mathrm{MY}}$. From $\mathcal{A}_{\mathrm{MA}}$ to $\mathcal{A}_{\mathrm{MB}}$ we present a homomorphism, showing that $\mathcal{A}_{\mathrm{MA}}$ cannot be smaller than $\mathcal{A}_{\mathrm{MB}}$—compare with [21].

When considering pointed expressions with only one point marking we obtain the position automaton in case of the mark after interpretation, while the other interpretation leads us to a *dual* version $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ of the position automaton. We show that the double reverse construction $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}$ is isomorphic to $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)$ and that the determinisation of $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)$ yields $\mathcal{A}_{\mathrm{MB}}(\alpha)$. Our study provides evidence that $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ plays a similar role as $\mathcal{A}_{\mathrm{POS}}$, but for the reverse expression $\alpha^{\mathrm{R}}$ instead of the expression $\alpha$. This is supported by the fact that $\mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$ is isomorphic to $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$ and $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$. It is worth mentioning that the double reverse automata $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}}$ and its determinisation $\mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$ get out of the line since the latter automaton turns out to be *not* isomorphic to $\mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$. The deterministic automaton $\mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$ is of its own interest, because it can be represented as the quotient among the studied deterministic ones, such as, Brzozowski's automaton $\mathcal{A}_{\mathrm{B}}$, the determinisation of the partial derivative automaton $\mathcal{A}_{\mathrm{PD}}$, the determinisation of the follow automaton $\mathcal{A}_{\mathrm{F}}$, the McNaughton-Yamada automaton $\mathcal{A}_{\mathrm{MY}}$ that is isomorphic to pointed expression automaton $\mathcal{A}_{\mathrm{MA}}$, and the pointed expression automaton $\mathcal{A}_{\mathrm{MB}}$. This shows, that although the taxonomy of "ordinary" conversions and double reversal conversions is quite similar, there are subtle differences that break the symmetry and the automaton $\mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$ plays a central role in

that symmetry break. The resulting taxonomy is also an improvement regarding the results in [3, 21]. Most proofs of our results are based on Glushkov's position concept which turns out to be highly valuable and can be used to describe automata constructions that look different at first sight, not only for the implementation use but also from the theoretical perspective.

## 2. Preliminaries

In this section we review some basic definitions about regular expressions and finite automata and fix notation.

### *2.1. Regular Expressions*

Given an alphabet (finite set of *letters* or *alphabet symbols*) $\Sigma$, the set RE of *regular expressions*, $\alpha$, over $\Sigma$ is defined by the following grammar:

$$\alpha \quad := \quad \emptyset \mid \varepsilon \mid \sigma_1 \mid \sigma_2 \mid \cdots \mid \sigma_k \mid (\alpha + \alpha) \mid (\alpha \cdot \alpha) \mid \alpha^\star, \tag{1}$$

where the operator $\cdot$ (concatenation) is often omitted. The *language* associated to $\alpha$ is denoted by $\mathcal{L}(\alpha)$ and defined as usual. The *size* $|\alpha|$ of $\alpha \in \mathrm{RE}$ is the number of symbols in $\alpha$ (disregarding parentheses). The *alphabetic size* $|\alpha|_\Sigma$ is its number of letters. We denote the subset of $\Sigma$ containing the symbols that occur in $\alpha$ by $\Sigma_\alpha$. 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 to $S$ 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 (including a singleton).

### *2.2. Finite Automata*

A *nondeterministic finite automaton* (NFA) is a five-tuple $A = \langle Q, \Sigma, \delta, I, F \rangle$ where $Q$ is a finite set of states, $\Sigma$ is a finite alphabet, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, and $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ is the transition function. We consider the size of an NFA as its number of states. An NFA that has transitions labelled with $\varepsilon$ is an $\varepsilon$-NFA. In this paper, excepted when explicitly mentioned, we will consider NFAs without $\varepsilon$ transitions. The transition function can be extended to words and to sets of states in the natural way. When $I = \{q_0\}$, we use $I = q_0$. We define the *finality* function $\varepsilon$ on $Q$ by $\varepsilon(q) = \varepsilon$ if $q \in F$ and $\varepsilon(q) = \emptyset$, otherwise. For $S \subseteq Q$ we have $\varepsilon(S) = \varepsilon$ iff there is some state $q \in S$ with $\varepsilon(q) = \varepsilon$, and $\varepsilon(S) = \emptyset$ otherwise. An NFA accepting a non-empty language is *trim* if every state is accessible from an initial state and every state leads to a final state.

Given a state $q \in Q$, the *right language* of $q$ is

$$\mathcal{L}_q(A) = \{\, w \in \Sigma^\star \mid \delta(q, w) \cap F \neq \emptyset \,\},$$

and the *left language* is

$$\overleftarrow{\mathcal{L}}_q(A) = \{\, w \in \Sigma^\star \mid q \in \delta(I, w) \,\}.$$

The *language accepted* by $A$ is $\mathcal{L}(A) = \bigcup_{q \in I} \mathcal{L}_q(A)$. Two automata are *equivalent* if they accept the same language. 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, we write $A_1 \simeq A_2$, if there exists a bijection $\varphi : Q_1 \longrightarrow Q_2$ 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 NFA is *deterministic* (DFA) if $|\delta(q, \sigma)| \leq 1$, for all $(q, \sigma) \in Q \times \Sigma$, and $|I| = 1$. In this case, we simply write $\delta(p, \sigma) = q$ instead of $\delta(p, \sigma) = \{q\}$. We can convert an NFA $A$ into an equivalent DFA $\mathrm{D}(A)$ by the *determinisation* operation D, using the well-known subset construction, where only subsets reachable from the initial subset of $\mathrm{D}(A)$ are used. Formally,

$$\mathrm{D}(A) = \langle Q_{\mathrm{D}}, \Sigma, \delta_{\mathrm{D}}, I_{\mathrm{D}}, F_{\mathrm{D}} \rangle,$$

where $Q_{\mathrm{D}} \subseteq 2^Q$, $I_{\mathrm{D}} = I$, $\delta_{\mathrm{D}}(S, \sigma) = \bigcup_{q \in S} \delta(q, \sigma)$ for $S \subseteq Q$, $\sigma \in \Sigma$, and $F_{\mathrm{D}} = \{\, S \in Q_{\mathrm{D}} \mid S \cap F \neq \emptyset \,\}$. Note that $S \in F_{\mathrm{D}}$ if and only if $\varepsilon(S) = \varepsilon$.

An equivalence relation $\equiv$ on $Q$ is *right invariant* w.r.t. an NFA $A$ if and only if:

- $\equiv \subseteq (Q - F)^2 \cup F^2$ and
- $\forall p, q \in Q, \sigma \in \Sigma,$ if $p \equiv q$, then $\forall p' \in \delta(p, \sigma)\ \exists q' \in \delta(q, \sigma)$ such that $p' \equiv q'$.

Given a set of states $S \subseteq Q$, we denote $S/\equiv\ = \{\, [q] \mid q \in S \,\}$. Note that $p \equiv q$ implies $\delta(p, \sigma)/\equiv\ = \delta(q, \sigma)/\equiv$, for $p, q \in Q$ and $\sigma \in \Sigma$. Furthermore, if $A$ is deterministic, then $p \equiv q$ implies $\delta(p, \sigma) \equiv \delta(q, \sigma)$. If $\equiv$ is a right-invariant relation on $Q$, the *quotient automaton* $A/\equiv$ is given by

$$A/\equiv\ = \langle Q/\equiv, \Sigma, \delta/\equiv, I/\equiv, F/\equiv \rangle,$$

where $\delta/\equiv([p], \sigma) = \{\, [q] \mid q \in \delta(p, \sigma) \,\} = \delta(p, \sigma)/\equiv$. It is easy to see that $\mathcal{L}(A/\equiv) = \mathcal{L}(A)$. Given a right-invariant relation $\equiv$ w.r.t. an NFA $A$, we can consider the natural extension of $\equiv$ w.r.t. $\mathrm{D}(A)$, where for $X, Y \subseteq 2^Q$ we have $X \mathrel{\hat{\equiv}} Y$ if and only if $X/\equiv\ = Y/\equiv$. The following lemma relates determinisation with these right-invariant relations.

**Lemma 1.** *$\mathrm{D}(A/\equiv) \simeq \mathrm{D}(A)/\mathrel{\hat{\equiv}}$, if $\equiv$ is a right-invariant relation w.r.t. $A$ and $\mathrel{\hat{\equiv}}$ is the natural extension of $\equiv$ to $\mathrm{D}(A)$.*

*Proof.* By definition, we have $\mathrm{D}(A/\equiv) = \langle (Q/\equiv)_{\mathrm{D}}, \Sigma, (\delta/\equiv)_{\mathrm{D}}, I/\equiv, (F/\equiv)_{\mathrm{D}} \rangle$. Each $X \in 2^{Q/\equiv}$ can be represented by $S/\equiv$, for some $S \subseteq Q$, e.g. $S = \{\, p \mid [p] \in X \,\}$. Then, for $\sigma \in \Sigma$ we have

$$(\delta/\equiv)_{\mathrm{D}}(S/\equiv, \sigma) = \bigcup_{[p] \in S/\equiv} \delta/\equiv([p], \sigma) \quad = \quad \bigcup_{p \in S} \delta(p, \sigma)/\equiv.$$

On the other hand, $\mathrm{D}(A)/\hat{\equiv} = \langle 2^Q/\hat{\equiv}, \Sigma, \delta_\mathrm{D}/\hat{\equiv}, [I], F_\mathrm{D}/\hat{\equiv}\rangle$. Consider the bijection $\varphi : 2^Q/\hat{\equiv} \longrightarrow 2^{Q/\equiv}$ defined by $\varphi([S]) = S/\equiv$. First we note that $\varphi([I]) = I/\equiv$ and $\varphi(F_\mathrm{D}/\hat{\equiv}) = (F/\equiv)_\mathrm{D}$ because we have,

$$\begin{aligned}
\varphi(F_\mathrm{D}/\hat{\equiv}) &= \{\varphi([X]) \mid X \in F_\mathrm{D}\} \\
&= \{X/\equiv \mid X \cap F \neq \emptyset\} \\
&= \{X/\equiv \mid X/\equiv \cap F/\equiv \neq \emptyset\} \\
&= (F/\equiv)_\mathrm{D}.
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\varphi(\delta_\mathrm{D}/\hat{\equiv}([X],\sigma)) &= \varphi([\delta_\mathrm{D}(X,\sigma)]) \\
&= \delta_\mathrm{D}(X,\sigma)/\equiv \\
&= \{[q] \mid q \in \delta_\mathrm{D}(X,\sigma)\} \\
&= \{[q] \mid q \in \bigcup_{p\in X} \delta(p,\sigma)\} \\
&= \bigcup_{p\in X} \delta(p,\sigma)/\equiv \\
&= \bigcup_{[p]\in X/\equiv} \delta/\equiv([p],\sigma) \\
&= (\delta/\equiv)_\mathrm{D}(X/\equiv,\sigma) \\
&= (\delta/\equiv)_\mathrm{D}(\varphi(X),\sigma).
\end{aligned}$$

□

In the rest of the paper we will denote $\hat{\equiv}$ by $\equiv$, whenever $\equiv$ is a right-invariant relation w.r.t. an NFA.

Given a language $L$ the reversal of $L$, $L^\mathrm{R}$, is the language obtained by reversing all the words in $L$. The reversal of a regular expression $\alpha$ is denoted by $\alpha^\mathrm{R}$, and is inductively defined by: $\alpha^\mathrm{R} = \alpha$ for $\alpha \in \Sigma \cup \{\varepsilon, \emptyset\}$, $(\alpha+\beta)^\mathrm{R} = \beta^\mathrm{R} + \alpha^\mathrm{R}$, $(\alpha\beta)^\mathrm{R} = \beta^\mathrm{R}\alpha^\mathrm{R}$ and $(\alpha^\star)^\mathrm{R} = (\alpha^\mathrm{R})^\star$. The reversal $\alpha^\mathrm{R}$ describes $\mathcal{L}(\alpha)^\mathrm{R}$. In the same way, given an automaton $A = \langle Q, \Sigma, \delta, I, F\rangle$ its reversal is $A^\mathrm{R} = \langle Q, \Sigma, \delta^\mathrm{R}, F, I\rangle$, where $\delta^\mathrm{R}(q,\sigma) = \{\, p \mid q \in \delta(p,\sigma)\,\}$ and $\mathcal{L}(A^\mathrm{R}) = \mathcal{L}(A)^\mathrm{R}$.

An equivalence relation $\equiv$ on $Q$ is *left invariant* w.r.t. an NFA $A$ if it is right invariant w.r.t. $A^\mathrm{R}$. This means that

- $\equiv \subseteq (Q-I)^2 \cup I^2$ and
- $\forall p, q \in Q, \sigma \in \Sigma$, if $p \equiv q$, then $\forall p' \in \delta^\mathrm{R}(p,\sigma)\ \exists q' \in \delta^\mathrm{R}(q,\sigma)$ such that $p' \equiv q'$.

For left-invariant relations and determinised automata we find the following situation:

**Lemma 2.** *Let $A$ be a trim NFA and consider $\equiv$ a left-invariant relation w.r.t. $A$. Then,* $\mathrm{D}(A) \simeq \mathrm{D}(A/\equiv)$.

*Proof.* Let $A = \langle Q, \Sigma, \delta, I, F\rangle$ and let $p, q \in Q$ be such that $p \equiv q$. We will show that for every accessible state $S$ in $\mathrm{D}(A)$, $p \in S$ implies $q \in S$. The result is true for the initial state, as $p \equiv q$ implies that $p \in I$ iff $q \in I$. Let $S = \delta_{\mathrm{D}}(S', \sigma)$ and suppose that the result is true for $S'$. If $p \in S$ there exists some $p' \in S'$ such that $\delta(p', \sigma) = p$, i.e., $p' \in \delta^{\mathrm{R}}(p, \sigma)$. Since $p \equiv q$, there is some $q' \in \delta^{\mathrm{R}}(q, \sigma)$ such that $p' \equiv q'$, and consequently $q' \in S'$. Then, $q \in S = \delta_{\mathrm{D}}(S', \sigma)$ results from the definition of the determinisation. We have that the bijection $\varphi : Q_{\mathrm{D}} \longrightarrow (Q/\equiv)_{\mathrm{D}}$ defined by $\varphi(S) = \{[q] \mid q \in S\}$ is an isomorphism between $\mathrm{D}(A)$ and $\mathrm{D}(A/\equiv)$. □

## 3. The Position and the Follow Automata

In this section we recall the definition of the position automaton and several related automata constructions. In particular, the determinisation of the position automaton, some $\varepsilon$-NFAs, and the follow automaton are considered. We show that the latter can be obtained directly from the regular expression.

To decide if a word is represented by a regular expression, one can scan the symbols of the regular expression in a specific way. For instance, given $\alpha = a(bb + aba)^\star b$ the word $abbabab$ can be obtained by scanning the first $a$, the two consecutive $b$s and then the second $a$, the third $b$, the third $a$, and the last $b$. This illustrates that uniquely identifying each letter of a regular expression is important for word recognition. Formally, given $\alpha \in \mathrm{RE}$, one can mark each occurrence of a letter $\sigma$ with its position in $\alpha$, considering reading it from left to right. The resulting regular expression is a *marked* regular expression $\overline{\alpha}$ with all symbols distinct and over the alphabet $\Sigma_{\overline{\alpha}}$. Then, a position $i \in [1, |\alpha|_\Sigma]$ corresponds to the symbol $\sigma_i$ in $\overline{\alpha}$, and consequently to exactly one occurrence of $\sigma$ in $\alpha$. For instance, $\overline{\alpha} = a_1(b_2b_3 + a_4b_5a_6)^\star b_7$. The same notation is used for unmarking, $\overline{\overline{\alpha}} = \alpha$. Let $\mathsf{Pos}(\alpha) = \{1, 2, \ldots, |\alpha|_\Sigma\}$, and $\mathsf{Pos}_0(\alpha) = \mathsf{Pos}(\alpha) \cup \{0\}$.

Positions were used by Glushkov [14] to define an NFA equivalent to $\alpha$, usually called the *position automaton* or Glushkov automaton ($\mathcal{A}_{\mathrm{POS}}(\alpha)$). Each state of the automaton, except for the initial state, corresponds to a position and there exists a transition from a position $i$ to a position $j$ by a letter $\sigma$ such that $\overline{\sigma_j} = \sigma$, if $\sigma_i$ can be followed by $\sigma_j$ in some word represented by $\overline{\alpha}$. More formally this reads as follows: the sets characterising the positions that can begin, end or be followed in words of $\mathcal{L}(\overline{\alpha})$ are, respectively:

$$\begin{aligned} \mathsf{First}(\alpha) &= \{\, i \mid \sigma_i w \in \mathcal{L}(\overline{\alpha}) \,\}, \\ \mathsf{Last}(\alpha) &= \{\, i \mid w\sigma_i \in \mathcal{L}(\overline{\alpha}) \,\}, \end{aligned}$$

and given $i \in \mathsf{Pos}(\alpha)$,

$$\mathsf{Follow}(\alpha, i) = \{\, j \mid u\, \sigma_i\sigma_j v \in \mathcal{L}(\overline{\alpha}) \,\}.$$

These sets can be defined inductively in the structure of the marked regular

expression as follows [6, 7, 16]:

$$\begin{aligned}
\mathsf{First}(\sigma_i) &= \{i\},\\
\mathsf{First}(\alpha_1+\alpha_2) &= \mathsf{First}(\alpha_1)\cup\mathsf{First}(\alpha_2),\\
\mathsf{First}(\alpha_1\alpha_2) &= \mathsf{First}(\alpha_1)\cup\varepsilon(\alpha_1)\,\mathsf{First}(\alpha_2),\\
\mathsf{First}(\alpha^\star) &= \mathsf{First}(\alpha).
\end{aligned}$$

The equations for $\mathsf{Last}$ coincide with those for $\mathsf{First}$ except for the case of concatenation:

$$\mathsf{Last}(\alpha_1\alpha_2) = \mathsf{Last}(\alpha_2)\cup\varepsilon(\alpha_2)\,\mathsf{Last}(\alpha_1).$$

Finally, for $\mathsf{Follow}$ we have the following:

$$\begin{aligned}
\mathsf{Follow}(\sigma_i,i) &= \emptyset,\\
\mathsf{Follow}(\alpha_1+\alpha_2,i) &= \begin{cases}\mathsf{Follow}(\alpha_1,i) & \text{if } i\in\mathsf{Pos}(\alpha_1),\\ \mathsf{Follow}(\alpha_2,i) & \text{if } i\in\mathsf{Pos}(\alpha_2),\end{cases}\\
\mathsf{Follow}(\alpha_1\alpha_2,i) &= \begin{cases}\mathsf{Follow}(\alpha_1,i) & \text{if } i\in\mathsf{Pos}(\alpha_1)\wedge i\notin\mathsf{Last}(\alpha_1),\\ \mathsf{Follow}(\alpha_1,i)\cup\mathsf{First}(\alpha_2) & \text{if } i\in\mathsf{Last}(\alpha_1),\\ \mathsf{Follow}(\alpha_2,i) & \text{if } i\in\mathsf{Pos}(\alpha_2).\end{cases}\\
\mathsf{Follow}(\alpha^\star,i) &= \begin{cases}\mathsf{Follow}(\alpha,i) & \text{if } i\notin\mathsf{Last}(\alpha),\\ \mathsf{Follow}(\alpha,i)\cup\mathsf{First}(\alpha) & \text{otherwise.}\end{cases}
\end{aligned}$$

We also define $\mathsf{Last}_0(\alpha)=\mathsf{Last}(\alpha)\cup\varepsilon(\alpha)\{0\}$, $\mathsf{Follow}(\alpha,0)=\mathsf{First}(\alpha)$ and

$$\mathsf{Follow}(\alpha)=\bigcup_{i\in\mathsf{Pos}(\alpha)}\{\,(i,j)\mid j\in\mathsf{Follow}(\alpha,i)\,\}.$$

Furthermore, given $S\in 2^{\mathsf{Pos}_0(\alpha)}$ let $\mathsf{Follow}(\alpha,S)=\bigcup_{i\in S}\mathsf{Follow}(\alpha,i)$. Then, the *position automaton* for $\alpha$ is

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

where $\delta_{\mathrm{POS}}(i,\sigma)=\{\,j\mid j\in\mathsf{Follow}(\alpha,i)\text{ and }\sigma=\overline{\sigma_j}\,\}$.

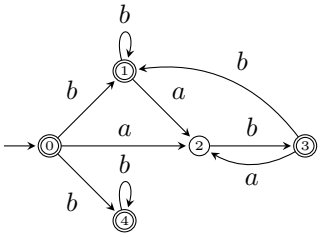
**Proposition 3** ([14]). *$\mathcal{L}(\mathcal{A}_{\mathrm{POS}}(\alpha))=\mathcal{L}(\alpha)$.*

The following example gives some intuition on the construction of the position automaton and its behaviour. Note that $\varepsilon(0)=\varepsilon(\alpha)$, and we will use either one or the other as it will be more convenient.

**Example 1.** *Consider $\alpha=(b+ab)^\star+b^\star$ with $\overline{\alpha}=(b_1+a_2b_3)^\star+b_4^\star$. Then, $\mathsf{First}(\alpha)=\{1,2,4\}$, $\mathsf{Last}_0(\alpha)=\{0,1,3,4\}$ and*

$$\mathsf{Follow}(\alpha)=\{(1,1),(1,2),(2,3),(3,1),(3,2),(4,4)\}.$$

*The position automaton $\mathcal{A}_{\mathrm{POS}}$ for $\alpha$ is depicted below.*




□

Note that each state, different from 0, in the position automaton corresponds to a symbol $\sigma_i$ in $\overline{\alpha}$, where $\sigma = \overline{\sigma_i}$ is the symbol just read. Thus, one can define a function $\mathsf{Select}$ that selects from a set of positions $S \subseteq \mathsf{Pos}(\alpha)$, those that correspond to a given letter, i.e., $\mathsf{Select} : 2^{\mathsf{Pos}(\alpha)} \times \Sigma \longrightarrow 2^{\mathsf{Pos}(\alpha)}$ is defined by

$$\mathsf{Select}(S, \sigma) = \{\, i \mid i \in S \text{ and } \overline{\sigma_i} = \sigma \,\}.$$

Then, the transition function $\delta_{\mathrm{POS}}$ can be defined by composing $\mathsf{Follow}$ with $\mathsf{Select}$, i.e.,

$$\delta_{\mathrm{POS}}(i, \sigma) \quad = \quad \mathsf{Select}(\mathsf{Follow}(\alpha, i), \sigma). \tag{2}$$

The same notion[1] was used by McNaughton and Yamada [19] to define an automaton which corresponds to the determinisation of the position automaton. With the definition of $\delta_{\mathrm{POS}}$ in (2) and considering the determinisation algorithm, the McNaughton-Yamada DFA can be defined as

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

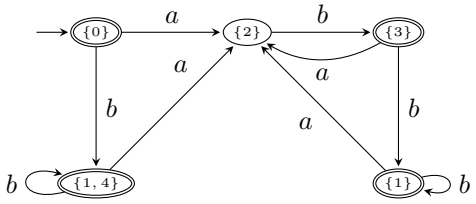
where $Q_{\mathrm{MY}} \subseteq 2^{\mathsf{Pos}_0(\alpha)}$, $F_{\mathrm{MY}} = \{\, S \in Q_{\mathrm{MY}} \mid \varepsilon(S) = \varepsilon \,\}$ and for $S \in 2^{\mathsf{Pos}(\alpha)}$ and $\sigma \in \Sigma$,

$$\delta_{\mathrm{MY}}(S, \sigma) = \mathsf{Select}(\mathsf{Follow}(\alpha, S), \sigma). \tag{3}$$

**Proposition 4** ([19])**.** $\mathcal{L}(\mathcal{A}_{\mathrm{MY}}(\alpha)) = \mathcal{L}(\alpha)$.

Let us continue with an example for a McNaughton-Yamada DFA.

**Example 2.** *Applying the McNaughton-Yamada construction to $\alpha$ from Example 1, we obtain the following DFA, $\mathcal{A}_{\mathrm{MY}}(\alpha)$:*




[1] Some authors use slightly different notions of marking [12, 19], which have in common that each symbol in the marked expression corresponds to exactly one occurrence of a symbol in the original expression.

*The states of this automaton are sets of positions. Given $S \subseteq \mathsf{Pos}_0(\alpha)$ and a letter $\sigma$, one computes the target of a transition from $S$ by $\sigma$, by first considering the set of all positions following in $\alpha$ some position in $S$, and then selecting the ones that correspond to $\sigma$. For instance,*

$$\delta_{\mathrm{MY}}(\{1,4\},a) = \mathsf{Select}(\mathsf{Follow}(\alpha,\{1,4\}),a) = \mathsf{Select}(\{1,2,4\},a) = \{2\},$$

*and*

$$\delta_{\mathrm{MY}}(\{1,4\},b) = \mathsf{Select}(\{1,2,4\},b) = \{1,4\}. \qquad \square$$

In the forthcoming we review the Thompson like construction of the follow automaton $\mathcal{A}_{\mathrm{F}}$, which was introduced by Ilie and Yu in [16]. We show that one can directly construct this automaton by an appropriate state labelling inspired by the position automaton $\mathcal{A}_{\mathrm{POS}}$.

### *3.1. The Follow Automaton $\mathcal{A}_{\mathrm{F}}$*

The most used conversion from regular expressions to equivalent $\varepsilon$-NFAs is the Thompson conversion [24], $\mathcal{A}_{\varepsilon\text{-T}}$. An improved use of $\varepsilon$-transitions lead to the definition of the $\varepsilon$-follow automaton [16], $\mathcal{A}_{\varepsilon\text{- F}}$—compare this automaton with that constructed in [22]. Both constructions can be defined inductively on the structure of the regular expressions and are schematically presented in Figure 2.

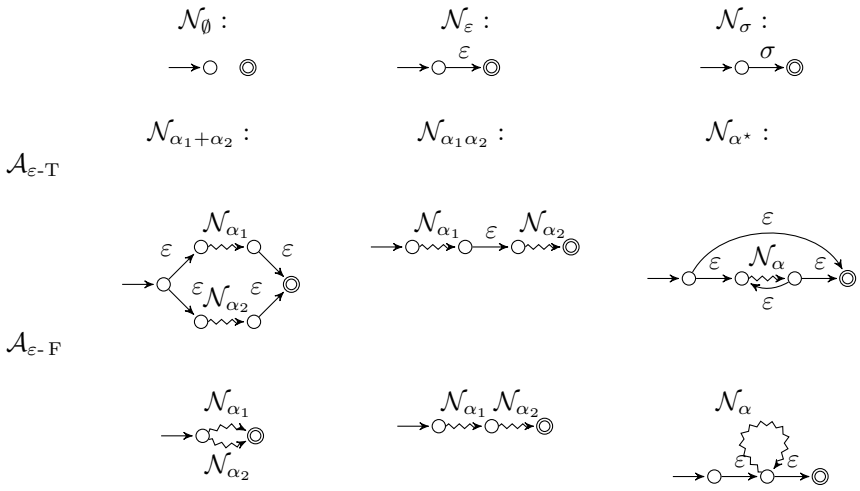



Figure 2: $\mathcal{A}_{\varepsilon\text{-T}}$ and $\mathcal{A}_{\varepsilon\text{- F}}$ constructions. For atomic expressions, the two constructions coincide.

From a Thompson automaton, if $\varepsilon$-transitions are eliminated in an adequate manner, the position automaton is obtained [1, 13]. Eliminating $\varepsilon$-transitions from the $\varepsilon$-follow automaton, the resulting automaton is the *follow automaton* $\mathcal{A}_{\mathrm{F}}(\alpha)$ which was introduced by Ilie and Yu [16] in 2003.

**Proposition 5** ([16])**.** $\mathcal{L}(\mathcal{A}_{\mathrm{F}}(\alpha)) = \mathcal{L}(\alpha)$.

They also showed that the follow automaton is a quotient of the position automaton, obtained by identifying positions with the same $\mathsf{Follow}$ set. For instance, in the position automaton of Example 1, one can see that $\mathsf{Follow}(\alpha,1) = \mathsf{Follow}(\alpha,3) = \{1,2\}$, and that 1 and 3 are both accepting states. Formally, Ilie and Yu considered the right-invariant equivalence relation $\equiv_{\mathrm{F}}$ defined on the set of states $\mathsf{Pos}_0(\alpha)$, w.r.t. $\mathcal{A}_{\mathrm{POS}}(\alpha)$, by

$$i \equiv_{\mathrm{F}} j \;\Leftrightarrow\; \mathsf{Follow}(\alpha,i) = \mathsf{Follow}(\alpha,j) \text{ and } \varepsilon(i) = \varepsilon(j),$$

and showed that $\mathcal{A}_{\mathrm{F}}(\alpha) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha)/\equiv_{\mathrm{F}}$. Based on this result we present a new definition of the automaton $\mathcal{A}_{\mathrm{F}}(\alpha)$ computed directly from $\alpha$, by labelling states not with positions $i \in \mathsf{Pos}_0(\alpha)$, but with their $\mathsf{Follow}$ sets and their finality.

**Definition 1.** *Let* $\mathcal{A}_{\mathrm{F}}(\alpha) = \langle \mathsf{F}(\alpha), \Sigma, \delta_{\mathrm{F}}, (\mathsf{Follow}(\alpha,0), \varepsilon(0)), F_{\mathrm{F}} \rangle$, *where*

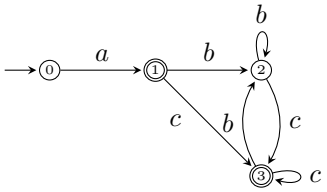
$$\mathsf{F}(\alpha) = \{\, (\mathsf{Follow}(\alpha,i), \varepsilon(i)) \mid i \in \mathsf{Pos}_0(\alpha) \,\} \subseteq 2^{\mathsf{Pos}(\alpha)} \times \{\varepsilon, \emptyset\},$$

$F_{\mathrm{F}} = \{\, (S,c) \in \mathsf{F}(\alpha) \mid c = \varepsilon \,\}$, *and for* $(S,c) \in \mathsf{F}(\alpha)$ *and* $\sigma \in \Sigma$,

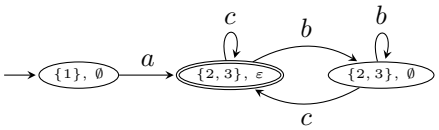
$$\delta_{\mathrm{F}}((S,c),\sigma) = \{\, (\mathsf{Follow}(\alpha,j), \varepsilon(j)) \mid j \in \mathsf{Select}(S,\sigma) \,\}.$$

The transition function $\delta_{\mathrm{F}}$ is defined as a composition of $\mathsf{Select}$ with $\mathsf{Follow}$, instead of $\mathsf{Follow}$ with $\mathsf{Select}$ as for $\delta_{\mathrm{POS}}$ (and $\delta_{\mathrm{MY}}$). The following example illustrates the need of considering the finality of a position in the state labels.

**Example 3.** *Consider the expression* $\alpha = a(b^\star c)^\star$ *with* $\overline{\alpha} = a_1(b_2^\star c_3)^\star$. *The position automaton for* $\alpha$, $\mathcal{A}_{\mathrm{POS}}(\alpha)$, *is the following:*




*It is* $(\mathsf{Follow}(\alpha,0), \varepsilon(0)) = (\{1\}, \emptyset)$, $(\mathsf{Follow}(\alpha,1), \varepsilon(1)) = (\mathsf{Follow}(\alpha,3), \varepsilon(3)) = (\{2,3\}, \varepsilon)$ *and* $(\mathsf{Follow}(\alpha,2), \varepsilon(2)) = (\{2,3\}, \emptyset)$. *We can see that the states* 1, 2 *and* 3 *have the same* $\mathsf{Follow}$ *value but different finalities. Considering the relation* $\equiv_{\mathrm{F}}$ *it is easy to see that the states* 1 *and* 3 *will be merged in* $\mathcal{A}_{\mathrm{POS}}(\alpha)/\equiv_{\mathrm{F}}$. *The follow automaton,* $\mathcal{A}_{\mathrm{F}}(\alpha)$, *is depicted below.*




□

From now on, in automata diagrams, we will omit the indication of the finality in labels of states, since this is already explicit from the representation of the nodes. With the last definition of $\mathcal{A}_{\mathrm{F}}$ we obtain an alternative proof of the result by Ilie and Yu.

**Proposition 6.** $\mathcal{A}_{\mathrm{F}}(\alpha) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha)/{\equiv_{\mathrm{F}}}$.

*Proof.* Let $\varphi_{\mathrm{F}} : \mathsf{Pos}_0(\alpha)/{\equiv_{\mathrm{F}}} \longrightarrow \mathsf{F}(\alpha)$ be defined by $\varphi_{\mathrm{F}}([i]) = (\mathsf{Follow}(\alpha, i), \varepsilon(i))$. By definition, $\varphi_{\mathrm{F}}$ is a bijection and preserves initial as well as final states. Furthermore, for $[i] \in \mathsf{Pos}_0(\alpha)/{\equiv_{\mathrm{F}}}$ and $\sigma \in \Sigma$ we have

$$\begin{aligned}\varphi_{\mathrm{F}}(\delta_{\mathrm{POS}}/{\equiv_{\mathrm{F}}}([i], \sigma)) &= \varphi_{\mathrm{F}}(\{\, [j] \mid j \in \mathsf{Select}(\mathsf{Follow}(\alpha, i), \sigma) \,\}) \\ &= \{\, \varphi_{\mathrm{F}}([j]) \mid j \in \mathsf{Select}(\mathsf{Follow}(\alpha, i), \sigma) \,\} \\ &= \delta_{\mathrm{F}}((\mathsf{Follow}(\alpha, i), \varepsilon(i)), \sigma) = \delta_{\mathrm{F}}(\varphi_{\mathrm{F}}([i]), \sigma).\end{aligned}$$

This shows that $\varphi_{\mathrm{F}}$ is an isomorphism. □

## 4. Automata Based on Pointed Expressions

Next we review two deterministic automata constructions, $\mathcal{A}_{\mathrm{MB}}$ and $\mathcal{A}_{\mathrm{MA}}$, that are based on recent approaches by Asperti et al. [3] and by Nipkow and Traytel [21] using pointed expressions. In a pointed regular expression, several positions are selected, and are graphically marked with a *point* corresponding to a letter. Those automata correspond to two different interpretations of a *pointed* expression, i.e., of a given set of positions $S$: in the first case, given a letter $\sigma$ one selects which positions from $S$ correspond to that letter and then determines which possible positions can follow; in the second case the set of positions $S$ corresponds to where one can be *after* reading the letter $\sigma$. For instance, the pointed regular expression $a(\bullet bb + \bullet aba)^\star \bullet b$ characterises the set of positions $\{2, 4, 7\}$. Intuitively, these are the positions which have been reached after reading some prefix of an input word. Asperti et al. thought that their algorithm "*au point*" computed a DFA isomorphic to $\mathcal{A}_{\mathrm{MY}}(\alpha)$, but Nipkow and Traytel [21] showed that their construction led to a dual automaton and called it *mark before*, $\mathcal{A}_{\mathrm{MB}}$, while $\mathcal{A}_{\mathrm{MY}}$ was isomorphic to a *mark after*, $\mathcal{A}_{\mathrm{MA}}$. Using the notation of the previous section, a transition in $\mathcal{A}_{\mathrm{MA}}$ is a composition of $\mathsf{Follow}$ with $\mathsf{Select}$ similarly as described in (2), while in $\mathcal{A}_{\mathrm{MB}}$ it will be a composition of $\mathsf{Select}$ with $\mathsf{Follow}$. Because of the behaviour of the transition function $\delta_{\mathrm{MY}}$ of $\mathcal{A}_{\mathrm{MY}}(\alpha)$, Nipkow and Traytel called this construction *mark after* ($\mathcal{A}_{\mathrm{MA}}(\alpha)$).

In this section, we show that the $\mathcal{A}_{\mathrm{MB}}$ is isomorphic to a quotient of the determinisation of $\mathcal{A}_{\mathrm{F}}$, and as a corollary it follows that $\mathcal{A}_{\mathrm{MA}}$ ($\mathcal{A}_{\mathrm{MY}}$) cannot be smaller than $\mathcal{A}_{\mathrm{MB}}$ (as already stated by Nipkow and Traytel). Moreover, we also consider the case, where one restricts pointed regular expressions with only *one* point marking a position. Obviously, the *mark after* automaton of single pointed expressions is related to the position automaton.

*4.1. The Automaton $\mathcal{A}_{\mathrm{MB}}$ Versus $\mathrm{D}(\mathcal{A}_{\mathrm{F}})$*

As mentioned above, Asperti *et al.* introduced the notion of pointed regular expression in order to obtain a compact representation of a set of positions. However, a point was 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 $\mathcal{A}_{\mathrm{POS}}$ and $\mathcal{A}_{\mathrm{MA}}$. The resulting construction was called *mark before*, $\mathcal{A}_{\mathrm{MB}}$, by Nipkow and Traytel. In our framework, this means that $\delta_{\mathrm{MB}}$ is a composition of $\mathsf{Follow}$ with $\mathsf{Select}$. Formally, given $\alpha \in \mathrm{RE}$, let

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

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

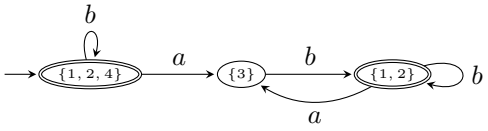
$$\delta_{\mathrm{MB}}((S, c), \sigma) = (\mathsf{Follow}(\alpha, \mathsf{Select}(S, \sigma)), \varepsilon(\mathsf{Select}(S, \sigma))),$$

and $F_{\mathrm{MB}} = \{\, (S, c) \mid c = \varepsilon \,\}$. In $Q_{\mathrm{MB}}$ we consider only the states that are accessible from the initial state by $\delta_{\mathrm{MB}}$.

**Proposition 7** ([3, 21])**.** $\mathcal{L}(\mathcal{A}_{\mathrm{MB}}(\alpha)) = \mathcal{L}(\alpha)$.

As before, we continue with a small example.

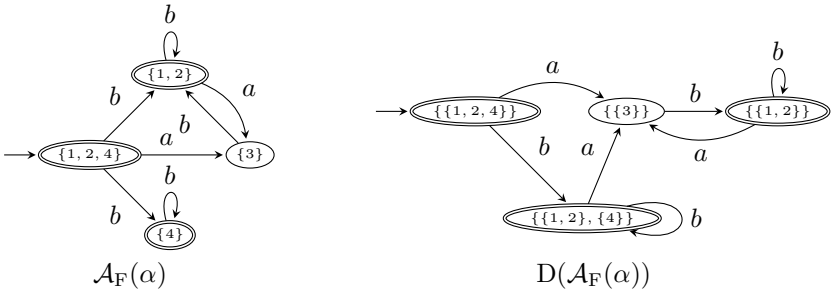
**Example 4.** *Consider again the regular expression $\alpha$ from Example 1. The $\mathcal{A}_{\mathrm{MB}}(\alpha)$ DFA is depicted below.*




*Note that the first state label is the set $\mathsf{First}(\alpha)$, and one can see that two states are saved when comparing with $\mathcal{A}_{\mathrm{MA}}$, in Example 2.* □

One could expect that the $\mathcal{A}_{\mathrm{MB}}$ construction was isomorphic to the determinisation of $\mathcal{A}_{\mathrm{F}}$. But we will see that in general that is not the case. The determinisation of $\mathcal{A}_{\mathrm{F}}$, $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha)) = \langle Q_{\mathrm{D(F)}}, \Sigma, \delta_{\mathrm{D(F)}}, \{(\mathsf{Follow}(\alpha, 0), \varepsilon(0))\}, F_{\mathrm{D(F)}} \rangle$, can be obtained by the subset construction.

**Example 5.** *Considering again the regular expression $\alpha$ from Example 1, the $\mathcal{A}_{\mathrm{F}}(\alpha)$ and $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))$ are respectively:*



*It is clear that* $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))$ *is not isomorphic to* $\mathcal{A}_{\mathrm{MB}}(\alpha)$ *(see Example 4). However states labeled by* $\{(\{1,2,4\},\varepsilon)\}$ *and* $\{(\{1,2\},\varepsilon),(\{4\},\varepsilon)\}$ *in* $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))$ *are merged, the DFA* $\mathcal{A}_{\mathrm{MB}}(\alpha)$ *is obtained. Next we prove that if certain sets of sets in the determinisation of* $\mathcal{A}_{\mathrm{F}}$ *are flattened the resulting automaton is isomorphic to* $\mathcal{A}_{\mathrm{MB}}$. □

Let $\equiv_s$ be the equivalence relation on $2^{\mathsf{F}(\alpha)}$ defined by,

$$I \equiv_s J \;\Leftrightarrow\; \bigcup_{(S,\_)\in I} S = \bigcup_{(S,\_)\in J} S \text{ and } \varepsilon(I) = \varepsilon(J).$$

We find the following situation:

**Proposition 8.** $\mathcal{L}\left(\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))/\equiv_s\right) = \mathcal{L}(\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha)))$.

*Proof.* We need to proof that $\equiv_s$ is right invariant w.r.t. $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))$. Let $I, J \in Q_{\mathrm{D(F)}}$, such that $I \equiv_s J$. Then, by definition of $\equiv_s$, $I$ is a final state if and only if $J$ is final. Note that the subset construction implies that

$$\delta_{\mathrm{D(F)}}(I,\sigma) = \bigcup_{(S,c)\in I} \{\, (\mathsf{Follow}(\alpha,j),\varepsilon(j)) \mid j \in S \text{ and } \sigma = \overline{\sigma_j}\,\}.$$

We want to show that for every $\sigma \in \Sigma$, one has $\delta_{\mathrm{D(F)}}(I,\sigma) \equiv_s \delta_{\mathrm{D(F)}}(J,\sigma)$. In fact, we show that both sets are identical, since

$$\begin{aligned}
&(T,d) \in \delta_{\mathrm{D(F)}}(I,\sigma)\\
&\quad\Leftrightarrow\quad \exists (S,c) \in I, j \in S, \text{ s. t. } \sigma = \overline{\sigma_j} \text{ and } (T,d) = (\mathsf{Follow}(\alpha,j),\varepsilon(j))\\
&\quad\Leftrightarrow\quad \exists (S',c') \in J, j \in S', \text{ s. t. } \sigma = \overline{\sigma_j} \text{ and } (T,d) = (\mathsf{Follow}(\alpha,j),\varepsilon(j))\\
&\quad\Leftrightarrow\quad (T,d) \in \delta_{\mathrm{D(F)}}(J,\sigma).
\end{aligned}$$

This proves the stated claim. □

Next we show that the determinised follow automaton $\mathcal{A}_{\mathrm{F}}$ w.r.t. the relation $\equiv_s$ is isomorphic to $\mathcal{A}_{\mathrm{MB}}$.

**Proposition 9.** $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))/\equiv_s \simeq \mathcal{A}_{\mathrm{MB}}(\alpha)$.

*Proof.* Consider $\varphi_{\mathrm{S}} : Q_{\mathrm{D(F)}}/\equiv_s \longrightarrow Q_{\mathrm{MB}}$ defined by $\varphi_{\mathrm{S}}([I]) = (\bigcup_{(S,\_)\in I} S, \varepsilon(I))$. It follows directly from the definition of $\equiv_s$ that $\varphi_{\mathrm{S}}$ is injective. Furthermore, by the definition of $\varphi_{\mathrm{S}}$, one has that $[I]$ is a final state if and only if $\varphi_{\mathrm{S}}([I])$ is final. Now, consider the initial state $[\{(\mathsf{Follow}(\alpha,0),\varepsilon(0))\}]$ of the DFA $\mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha))/\equiv_s$. Then, $\varphi_{\mathrm{S}}([\{(\mathsf{Follow}(\alpha,0),\varepsilon(0))\}]) = (\mathsf{Follow}(\alpha,0),\varepsilon(0))$, which is the initial state of $\mathcal{A}_{\mathrm{MB}}(\alpha)$. Next we show that for every $\sigma \in \Sigma$ and $[I] \in Q_{\mathrm{D(F)}}/\equiv_s$,

$$\varphi_{\mathrm{S}}(\delta_{\mathrm{D(F)}}/\equiv_s([I],\sigma)) = \delta_{\mathrm{MB}}(\varphi_{\mathrm{S}}([I]),\sigma).$$

Let $S_I = \bigcup_{(S,\_)\in I} S$. Then

$$\begin{aligned}
&\varphi_{\mathrm{S}}(\delta_{\mathrm{D(F)}}/{\equiv_s}([I],\sigma)) \\
&\quad = \varphi_{\mathrm{S}}([\{\,(\mathsf{Follow}(\alpha,j),\varepsilon(j)) \mid j \in S_I \text{ and } \sigma = \overline{\sigma_j}\,\}]) \\
&\quad = \varphi_{\mathrm{S}}([\{\,(\mathsf{Follow}(\alpha,j),\varepsilon(j)) \mid j \in \mathsf{Select}(S_I,\sigma)\,\}]) \\
&\quad = (\{\,i \mid i \in \mathsf{Follow}(\alpha,j) \text{ and } j \in \mathsf{Select}(S_I,\sigma)\,\}, \varepsilon(\mathsf{Select}(S_I,\sigma)) \\
&\quad = (\mathsf{Follow}(\alpha,\mathsf{Select}(S_I,\sigma)), \varepsilon(\mathsf{Select}(S_I,\sigma))) \\
&\quad = \delta_{\mathrm{MB}}((S_I,\varepsilon(I)),\sigma) \\
&\quad = \delta_{\mathrm{MB}}(\varphi_{\mathrm{S}}([I]),\sigma).
\end{aligned}$$

This also proves that $\varphi_{\mathrm{S}}$ is surjective. □

Nipkow and Traytel presented a homomorphism from $\mathcal{A}_{\mathrm{MA}}$ to $\mathcal{A}_{\mathrm{MB}}$, showing that $\mathcal{A}_{\mathrm{MA}}$ cannot be smaller than $\mathcal{A}_{\mathrm{MB}}$. The same result is a direct corollary of the above results.

**Corollary 1.** $\mathcal{A}_{\mathrm{MB}}(\alpha) \simeq (\mathcal{A}_{\mathrm{MA}}(\alpha)/{\equiv_{\mathrm{F}}})/{\equiv_s}$.

*Proof.* By Proposition 9, we have $\mathcal{A}_{\mathrm{MB}}(\alpha) \simeq \mathrm{D}\,(\mathcal{A}_{\mathrm{F}}(\alpha))/{\equiv_s}$. But, by Proposition 6, $\mathrm{D}\,(\mathcal{A}_{\mathrm{F}}(\alpha))/{\equiv_s} \simeq \mathrm{D}\,(\mathcal{A}_{\mathrm{POS}}(\alpha)/{\equiv_{\mathrm{F}}})/{\equiv_s}$, which by Lemma 1 is isomorphic to $(\mathrm{D}\,(\mathcal{A}_{\mathrm{POS}}(\alpha))/{\equiv_{\mathrm{F}}})/{\equiv_s} = (\mathcal{A}_{\mathrm{MA}}(\alpha)/{\equiv_{\mathrm{F}}})/{\equiv_s}$. □

We continue Example 2 explaining this isomorfism in more detail.

**Example 6.** *Considering $\mathcal{A}_{\mathrm{MY}}$ ($\mathcal{A}_{\mathrm{MA}}$) from Example 2, we have $\{1\} \equiv_{\mathrm{F}} \{3\}$, since $[1]_{\equiv_{\mathrm{F}}} = [3]_{\equiv_{\mathrm{F}}}$. Furthermore, using the isomorphism $\varphi_{\mathrm{F}}$ from Proposition 6,*

$$\varphi_{\mathrm{F}}(\{[0]_{\equiv_{\mathrm{F}}}\}) = \{(\{1,2,4\},\varepsilon)\} \equiv_s \{(\{1,2\},\varepsilon),(\{4\},\varepsilon)\} = \varphi_{\mathrm{F}}(\{[1]_{\equiv_{\mathrm{F}}},[4]_{\equiv_{\mathrm{F}}}\}). \quad (5)$$

*4.2. The Dual Position Automaton*

If one considers pointed regular expressions with only one point marking a position to be visited when reading a letter, an NFA, dual of $\mathcal{A}_{\mathrm{POS}}$ ($\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$), can be defined. We show that its determinisation yields $\mathcal{A}_{\mathrm{MB}}$. Given a regular expression $\alpha$, with $n = |\alpha|_\Sigma = |\,\mathsf{Pos}(\alpha)|$, the set of states of $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ is $\mathsf{Pos}(\alpha)$ plus an unique final state $n+1$. The set of initial states is $\mathsf{Follow}(\alpha,0) \cup \varepsilon(\alpha)\{n+1\}$. From a state $i \in \mathsf{Pos}(\alpha)$ reading $\sigma \in \Sigma$ one can move to $\mathsf{Follow}(\alpha,i)$ if $\overline{\sigma_i} = \sigma$. That is, by first selecting $\mathsf{Select}(\{i\},\sigma)$ which is $i$ if $\overline{\sigma_i} = \sigma$, and empty otherwise, and then applying $\mathsf{Follow}$. Moreover, if $\varepsilon(i) = \varepsilon$ there is a transition to $n+1$. Formally,

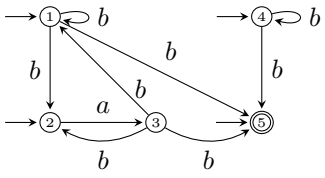
**Definition 2.** *Let*

$$\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha) = \langle \mathsf{Pos}(\alpha) \cup \{n+1\}, \Sigma, \delta_{\overleftarrow{\mathrm{POS}}}, \mathsf{Follow}(\alpha,0) \cup \varepsilon(\alpha)\{n+1\}, \{n+1\}\rangle,$$

*with* $\delta_{\overleftarrow{\mathrm{POS}}}(i,\sigma) = \mathsf{Follow}(\alpha,\mathsf{Select}(\{i\},\sigma)) \cup \varepsilon(\mathsf{Select}(\{i\},\sigma))\{n+1\}$.

This means that $\delta_{\overleftarrow{\mathrm{POS}}}(i,\sigma) = \mathsf{Follow}(\alpha,i) \cup \varepsilon(i)\{n+1\}$, only if $i \in \mathsf{Pos}(\alpha)$ and $\overline{\sigma_i} = \sigma$, being the empty set otherwise.

**Example 7.** *Considering again the regular expression $\alpha$ from Example 1, the $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)$ is the following:*




Observe that for each state of $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ all transitions *leaving* it have the same label. This is exactly the opposite of the position automaton $\mathcal{A}_{\mathrm{POS}}$, where for each state all transitions *into it* have the same label.

**Proposition 10.** $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)) \simeq \mathcal{A}_{\mathrm{MB}}(\alpha)$.

*Proof.* One has $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)) = \langle 2^{\mathsf{Pos}(\alpha)\cup\{n+1\}}, \Sigma, \delta_{\mathrm{D}(\overleftarrow{\mathrm{POS}})}, I_{\mathrm{D}(\overleftarrow{\mathrm{POS}})}, F_{\mathrm{D}(\overleftarrow{\mathrm{POS}})}\rangle$, where $I_{\mathrm{D}(\overleftarrow{\mathrm{POS}})} = \mathsf{Follow}(\alpha,0) \cup \varepsilon(\alpha)\{n+1\}$, $F_{\mathrm{D}(\overleftarrow{\mathrm{POS}})} = \{\, S \mid n+1 \in S \,\}$, and for $S \in \mathsf{Pos}(\alpha) \cup \{n+1\}$ and $\sigma \in \Sigma$,

$$\begin{aligned}\delta_{\mathrm{D}(\overleftarrow{\mathrm{POS}})}(S,\sigma) &= \bigcup_{i\in S} \mathsf{Follow}(\alpha, \mathsf{Select}(\{i\},\sigma)) \cup \varepsilon(\mathsf{Select}(\{i\},\sigma))\{n+1\} \\ &= \mathsf{Follow}(\alpha, \mathsf{Select}(S,\sigma)) \cup \varepsilon(\mathsf{Select}(S,\sigma))\{n+1\}.\end{aligned}$$

Note that, $n+1 \in \delta_{\mathrm{D}(\overleftarrow{\mathrm{POS}})}(S,\sigma)$ if and only if $\varepsilon(\mathsf{Select}(S,\sigma)) = \varepsilon$. Then the map $\varphi_{\mathrm{M}}(S) = (S, \varepsilon(S))$ defines an isomorphism between $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{POS}}})$ and $\mathcal{A}_{\mathrm{MB}}$. □

From the above proposition we obtain:

**Corollary 2.** $\mathcal{L}(\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)) = \mathcal{L}(\alpha)$.

We end this section noticing that a dual of the follow automaton, $\mathcal{A}_{\overleftarrow{\mathrm{F}}}$, can also be defined which will be a quotient of $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$. But we postpone its discussion to Section 6.

## 5. Derivative Based Constructions

We recall some constructions of automata from regular expressions based on derivatives and variants, such as, e.g., Brzozowski's construction by derivatives and the construction of Mirkin [20] and Antimirov [2] by partial derivatives. Here we focus on known results characterizing these automata as quotients of the position automaton. Then we consider a construction due to Yamamoto, the prefix automaton, and we show how it relates to the partial derivatives constructions. We also consider the dual of the partial derivative automaton and introduce a dual of the prefix automaton. Moreover we obtain that the

determinisation of the dual of the prefix automaton leads to the smaller DFA amongst the ones studied in this paper.

Brzozowski [8] defined a DFA equivalent to a regular expression using the notion of derivative. The derivative of a regular expression $\alpha$ w.r.t. $\sigma \in \Sigma$ is a regular expression $\mathrm{d}_\sigma(\alpha)$, such that $\mathcal{L}(\mathrm{d}_\sigma(\alpha)) = \{\, w \mid \sigma w \in \mathcal{L}(\alpha) \,\}$ and it is inductively defined by:

$$\begin{aligned}
\mathrm{d}_\sigma(\emptyset) = \mathrm{d}_\sigma(\varepsilon) &= \emptyset,\\
\mathrm{d}_\sigma(\sigma') &= \begin{cases} \{\varepsilon\} & \text{if } \sigma' = \sigma \\ \emptyset & \text{otherwise,} \end{cases}\\
\mathrm{d}_\sigma(\alpha + \alpha') &= \mathrm{d}_\sigma(\alpha) + \mathrm{d}_\sigma(\alpha'),\\
\mathrm{d}_\sigma(\alpha\alpha') &= \begin{cases} \mathrm{d}_\sigma(\alpha)\alpha' & \text{if } \varepsilon(\alpha) = \emptyset \\ \mathrm{d}_\sigma(\alpha)\alpha' + \mathrm{d}_\sigma(\alpha') & \text{otherwise,} \end{cases}\\
\mathrm{d}_\sigma(\alpha^\star) &= \mathrm{d}_\sigma(\alpha)\alpha^\star.
\end{aligned}$$

This notion can be extended to words: $\mathrm{d}_\varepsilon(\alpha) = \alpha$ and $\mathrm{d}_{\sigma w}(\alpha) = \mathrm{d}_w(\mathrm{d}_\sigma(\alpha))$. The set of all derivatives $\{\, \mathrm{d}_w(\alpha) \mid w \in \Sigma^\star \,\}$ of $\alpha$ may not be finite. For finiteness, Brzozowski considered the quotient of that set modulo some regular expression equivalences, $\mathcal{D}(\alpha)$. Here we consider the equivalences to be the associativity, commutativity and idempotence of $+$ ($\mathsf{ACI}$). The *Brzozowski's automaton* for $\alpha$ is

$$\mathcal{A}_{\mathrm{B}}(\alpha) = \langle \mathcal{D}(\alpha), \Sigma, \delta_{\mathrm{B}}, [\alpha], F_{\mathrm{B}} \rangle,$$

where $F_{\mathrm{B}} = \{\, [q] \in \mathcal{D}(\alpha) \mid \varepsilon(q) = \varepsilon \,\}$, and $\delta_{\mathrm{B}}([q], \sigma) = [\mathrm{d}_\sigma(q)]$, for all $[q] \in \mathcal{D}(\alpha)$ and $\sigma \in \Sigma$.

**Proposition 11** ([8])**.** $\mathcal{L}(\mathcal{A}_{\mathrm{B}}(\alpha)) = \mathcal{L}(\alpha)$.

We recall that Berry and Sethi [4] defined the position automaton using derivatives of a marked expression.

The partial derivative automaton of a regular expression was introduced independently by Mirkin [20] and Antimirov [2] as a nondeterministic contrapart of Brzozowski automaton. Champarnaud and Ziadi [10] proved that the two formulations are equivalent. For a regular expression $\alpha \in \mathrm{RE}$ and a symbol $\sigma \in \Sigma$, the set of partial derivatives of $\alpha$ w.r.t. $\sigma$ is defined inductively as follows:

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

where for any $S \subseteq \mathrm{RE}$ , we define $S\alpha' = \{\, \alpha\alpha' \mid \alpha \in S \,\}$ if $\alpha' \neq \emptyset, \varepsilon$ (and analogously for $\alpha' S$). The definition of partial derivatives can be extended in a natural way to sets of regular expressions, words, and languages. We have that $\mathcal{L}(\mathrm{d}_w(\alpha)) = \mathcal{L}(\partial_w(\alpha))$ for $w \in \Sigma^*$. The set of all partial derivatives of $\alpha$ w.r.t. words is denoted by $\mathsf{PD}(\alpha) = \partial_{\Sigma^*}(\alpha)$. The *partial derivative automaton* of $\alpha$ is

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

where $F_{\mathrm{PD}} = \{\, \tau \in \mathsf{PD}(\alpha) \mid \varepsilon(\tau) = \varepsilon \,\}$, and $\delta_{\mathrm{PD}}(\tau, \sigma) = \partial_\sigma(\tau)$, for all $\tau \in \mathsf{PD}(\alpha)$ and $\sigma \in \Sigma$.

**Proposition 12** ([2, 20])**.** $\mathcal{L}(\mathcal{A}_{\mathrm{PD}}(\alpha)) = \mathcal{L}(\alpha)$.

Champarnaud and Ziadi [11] proved that $\mathcal{A}_{\mathrm{PD}}$ is a quotient of the position automaton $\mathcal{A}_{\mathrm{POS}}$ by the right-invariant equivalence relation $\equiv_c$. Given a position $i$, either $\partial_{w\sigma_i}(\overline{\alpha}) = \emptyset$ for all $w \in \Sigma_{\overline{\alpha}}^\star$, or there is some expression $c_i(\alpha)$ such that for all $w \in \Sigma_{\overline{\alpha}}^\star$, $\partial_{w\sigma_i}(\overline{\alpha})$ is either empty or equal to the singleton $\{c_i(\alpha)\}$, which can be computed by the following rules. For $i \in \mathsf{Pos}(\alpha)$, we have $c_i(\emptyset) = c_i(\varepsilon) = \emptyset$, and $c_i(\sigma_i) = \varepsilon$. Now consider $\overline{\alpha}$ of the form $\alpha_1 + \alpha_2$, $\alpha_1\alpha_2$ or $\alpha_1^\star$. If $i$ occurs in $\alpha_1$, then $c_i(\alpha_1 + \alpha_2) = c_i(\alpha_1)$, $c_i(\alpha_1\alpha_2) = c_i(\alpha_1)\alpha_2$, and $c_i(\alpha_1^\star) = c_i(\alpha_1)\alpha_1^\star$. If $i$ occurs in $\alpha_2$, then $c_i(\alpha_1 + \alpha_2) = c_i(\alpha_1\alpha_2) = c_i(\alpha_2)$.

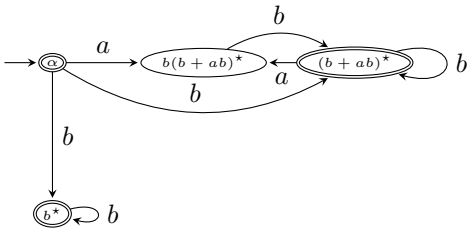
For $i, j \in \mathsf{Pos}_0(\alpha)$ and considering $c_0(\alpha) = \overline{\alpha}$, one has

$$i \equiv_c j \Leftrightarrow \overline{c_i(\alpha)} = \overline{c_j(\alpha)}.$$

**Proposition 13** ([11])**.** $\mathcal{A}_{\mathrm{PD}}(\alpha) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha)/\equiv_c$.

An example for $\mathcal{A}_{\mathrm{PD}}$ is given next.

**Example 8.** *For the expression $\alpha = (b + ab)^\star + b^\star$ from Example 1, the $\mathcal{A}_{\mathrm{PD}}$ automaton is the following:*




*On the other hand, $c_1(\alpha) = (b_1 + a_2b_3)^\star$, $c_2(\alpha) = b_3(b_1 + a_2b_3)^\star$, $c_3(\alpha) = (b_1 + a_2b_3)^\star$ and $c_4(\alpha) = b_4^\star$. Thus, $1 \equiv_c 3$ and those two states of $\mathcal{A}_{\mathrm{POS}}$ merged into the state labeled by $(b + ab)^\star$ in $\mathcal{A}_{\mathrm{PD}}$.*

It is worth mentioning that derivatives and partial derivatives w.r.t. a word can also be considered on the right, i.e., $\overleftarrow{\mathrm{d}}_w(\alpha)$ and $\overleftarrow{\partial}_w(\alpha)$, with $\mathcal{L}(\overleftarrow{\mathrm{d}}_w(\alpha)) = \mathcal{L}(\overleftarrow{\partial}_w(\alpha)) = \{\, x \mid xw \in \mathcal{L}(\alpha) \,\}$. In the same way one can define the right partial derivative automaton $\mathcal{A}_{\overleftarrow{\mathrm{PD}}}$ which is a quotient of $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ [18]. This is obviously related to the reversal of the expression and will be discussed in more detail in Section 6.

*5.1. Prefix automaton and its dual*

In this section we consider the *prefix automaton* construction, $\mathcal{A}_{\mathrm{Pre}}$, initially defined as a quotient of the Thompson's automaton $\mathcal{A}_{\varepsilon\text{-T}}$ by Yamamoto [25]. In the same paper the author presented the *suffix automaton* which is isomorphic to the partial derivative automaton $\mathcal{A}_{\mathrm{PD}}$. However, the prefix automaton does not coincides with the right-derivative automaton.

**Proposition 14** ([25])**.** $\mathcal{L}(\mathcal{A}_{\mathrm{Pre}}(\alpha)) = \mathcal{L}(\alpha)$.

Maia et al. [18, 17] characterised the $\mathcal{A}_{\mathrm{Pre}}$ automaton as a solution of a system of left regular expression equations (similar to Mirkin's equational system that defines $\mathcal{A}_{\mathrm{PD}}$) and expressed it as a quotient of $\mathcal{A}_{\mathrm{POS}}$ by a left-invariant equivalence relation, $\equiv_\ell$. Every state in $\mathcal{A}_{\mathrm{Pre}}$ is labelled either with $\varepsilon$ or with an expression of the form $\alpha\sigma$, which describes the left-language of that state. In the following we define a function $\mathsf{R}$, which given an expression $\alpha$ computes a set of normalised expressions of the form $\alpha'\sigma$, modulo associativity of concatenation, and such that $\mathcal{L}(\alpha) = \mathcal{L}(\mathsf{R}_\varepsilon(\alpha))$, where $\mathsf{R}_\varepsilon(\alpha) = \mathsf{R}(\alpha) \cup \varepsilon(\alpha)$. Note that we consider an expression $\sigma$ to be of the form $\alpha'\sigma$ with $\alpha' = \varepsilon$. For $\alpha \in \mathrm{RE}$, the set $\mathsf{R}(\alpha)$ is given by

$$\begin{aligned} \mathsf{R}(\emptyset) = \mathsf{R}(\varepsilon) &= \emptyset, \\ \mathsf{R}(\sigma) &= \{\sigma\}, \\ \mathsf{R}(\alpha + \alpha') &= \mathsf{R}(\alpha) \cup \mathsf{R}(\alpha'), \\ \mathsf{R}(\alpha\alpha') &= \alpha\,\mathsf{R}(\alpha') \cup \varepsilon(\alpha')\,\mathsf{R}(\alpha), \\ \mathsf{R}(\alpha^\star) &= \alpha^\star\,\mathsf{R}(\alpha). \end{aligned} \tag{6}$$

Starting with $\mathsf{R}_\varepsilon(\alpha)$ as the set of final states, the automaton $\mathcal{A}_{\mathrm{Pre}}(\alpha)$ can then be successively constructed backwards as follows. For each state of the form $\alpha'\sigma$ the set $\mathsf{R}_\varepsilon(\alpha')$ is computed and a transition by $\sigma$ is added from each element $\alpha'' \in \mathsf{R}_\varepsilon(\alpha')$ to $\alpha'\sigma$. The state labelled by $\varepsilon$ is the initial state of $\mathcal{A}_{\mathrm{Pre}}(\alpha)$. Formally, consider the function $\mathsf{p}_w(\alpha)$ for words $w \in \Sigma^\star$ defined as follows:

$$\begin{aligned} \mathsf{p}_\varepsilon(\alpha) &= \mathsf{R}_\varepsilon(\alpha), \\ \mathsf{p}_{\sigma w}(\alpha) &= \bigcup_{\alpha'\sigma \,\in\, \mathsf{p}_w(\alpha)} \mathsf{R}_\varepsilon(\alpha'). \end{aligned}$$

We have that $\mathcal{L}(\mathsf{p}_w(\alpha)) = \{\, x \mid xw \in \mathcal{L}(\alpha) \,\}$. Let $\mathsf{Pre}(\alpha) = \bigcup_{w\in\Sigma^\star} \mathsf{p}_w(\alpha)$. The *prefix* automaton of $\alpha$ is

$$\mathcal{A}_{\mathrm{Pre}}(\alpha) = \langle \mathsf{Pre}(\alpha), \Sigma, \delta_{\mathrm{Pre}}, \varepsilon, \mathsf{R}_\varepsilon(\alpha) \rangle,$$

where $\delta_{\mathrm{Pre}} = \{\, (\alpha'', \sigma, \alpha'\sigma) \mid \alpha'\sigma \in \mathsf{Pre}(\alpha),\, \alpha'' \in \mathsf{R}_\varepsilon(\alpha'),\, \sigma \in \Sigma \,\}$, that is, $\delta^{\mathrm{R}}_{\mathrm{Pre}}(\alpha'\sigma, \sigma) = \mathsf{R}_\varepsilon(\alpha')$, for all $\alpha'\sigma \in \mathsf{Pre}(\alpha)$.

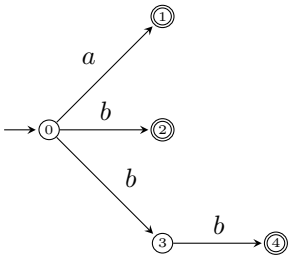
Considering a marked regular expression $\overline{\alpha}$ all the expressions of $\mathsf{Pre}(\overline{\alpha})$ are of the form that $\alpha'\sigma_i$ or $\varepsilon$. For each position $i \in \mathsf{Pos}(\alpha)$ there exists an unique $\alpha'\sigma_i \in \mathsf{Pre}(\overline{\alpha})$. Let $p_i(\alpha)$ be that expression. One has, $\mathcal{A}_{\mathrm{Pre}}(\overline{\alpha}) \simeq \mathcal{A}_{\mathrm{POS}}(\overline{\alpha})$. For $i, j \in \mathsf{Pos}_0(\alpha)$ the relation $\equiv_\ell$ defined below is left-invariant w.r.t. $\mathcal{A}_{\mathrm{POS}}$,

$$i \equiv_\ell j \;\Leftrightarrow\; \overline{p_i(\alpha)} = \overline{p_j(\alpha)}.$$

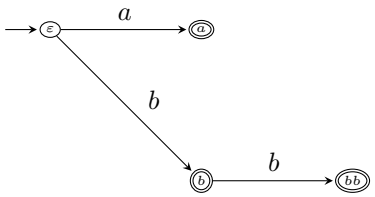
**Proposition 15** ([18]). $\mathcal{A}_{\mathrm{Pre}}(\alpha) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha)/\equiv_\ell$.

The following example explains the relation $\equiv_\ell$ in more detail.

**Example 9.** *Considering the regular expression* $\alpha = (a + b) + bb$. *The automata* $\mathcal{A}_{\mathrm{POS}}(\alpha)$ *and* $\mathcal{A}_{\mathrm{Pre}}(\alpha)$ *are depicted below, where* $2 \equiv_\ell 3$.




$\mathcal{A}_{\mathrm{POS}}(\alpha)$




$\mathcal{A}_{\mathrm{Pre}}(\alpha)$

However the determinisation of $\mathcal{A}_{\mathrm{Pre}}$ is isomorphic to the determinisation of $\mathcal{A}_{\mathrm{POS}}$, i.e., $\mathcal{A}_{\mathrm{MY}}$. This fact is a direct consequence of Lemma 2 presented in Section 2.

**Proposition 16.** $\mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha)) \simeq \mathcal{A}_{\mathrm{MY}}(\alpha)$.

Considering the concepts used to define $\mathcal{A}_{\mathrm{Pre}}$ in a symmetric manner, a new automaton construction from an expression can be obtained. We will show that that construction is closely related to the partial derivative automaton $\mathcal{A}_{\mathrm{PD}}$ and to the Brzozowski automaton $\mathcal{A}_{\mathrm{B}}$. First, let $\mathsf{L}(\alpha)$ be a set of normalised expressions of the form $\sigma\alpha'$, which is computed as $\mathsf{R}$ except for concatenation and Kleene star:

$$\mathsf{L}(\alpha\alpha') = \mathsf{L}(\alpha)\alpha' \cup \varepsilon(\alpha)\,\mathsf{L}(\alpha'),$$
$$\mathsf{L}(\alpha^\star) = \mathsf{L}(\alpha)\alpha^\star.$$

Again, let $\mathsf{L}_\varepsilon(\alpha) = \mathsf{L}(\alpha) \cup \varepsilon(\alpha)$. Moreover, let

$$\overleftarrow{\mathsf{p}}_\varepsilon(\alpha) = \mathsf{L}_\varepsilon(\alpha)$$
$$\overleftarrow{\mathsf{p}}_{w\sigma}(\alpha) = \bigcup_{\sigma\alpha' \in \overleftarrow{\mathsf{p}}_w(\alpha)} \mathsf{L}_\varepsilon(\alpha').$$

It is immediate that $\mathcal{L}(\alpha) = \mathcal{L}(\mathsf{L}_\varepsilon(\alpha))$ and $\mathcal{L}(\overleftarrow{\mathsf{p}}_w(\alpha)) = \{\, x \mid wx \in \mathcal{L}(\alpha) \,\}$.

**Definition 3.** *Let* $\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha) = \langle \overleftarrow{\mathrm{Pre}}(\alpha), \Sigma, \delta_{\overleftarrow{\mathrm{Pre}}}, \mathsf{L}_\varepsilon(\alpha), \varepsilon \rangle$, *where*

$$\overleftarrow{\mathrm{Pre}}(\alpha) = \bigcup_{w \in \Sigma^\star} \overleftarrow{\mathsf{p}}_w(\alpha)$$

*and* $\delta_{\overleftarrow{\mathrm{Pre}}}(\alpha', \sigma) = \mathsf{L}_\varepsilon(\alpha'')$ *if* $\alpha' = \sigma\alpha''$, *and* $\delta_{\overleftarrow{\mathrm{Pre}}}(\alpha', \sigma) = \emptyset$ *otherwise.*

It follows that

**Proposition 17.** $\mathcal{L}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha)) = \mathcal{L}(\alpha)$.

Moreover from the subset construction we have,

$$\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha)) = \langle Q_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}, \Sigma, \delta_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}, \mathsf{L}_\varepsilon(\alpha), F_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}\rangle,$$

where $F_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})} = \{\, X \in Q_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})} \mid \varepsilon \in X \,\}$ and $\delta_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}(X, \sigma) = \bigcup_{\sigma\alpha' \in X} \mathsf{L}_\varepsilon(\alpha')$. In the following, we will show that this automaton is a quotient of both, the Brzozowski automaton and of the determinisation of the partial derivative automaton. The right-invariant relation is the same for both cases.

**Lemma 18.** $\mathsf{L}_\varepsilon(\alpha) = \bigcup_{\sigma\in\Sigma} \sigma\partial_\sigma(\alpha) \cup \varepsilon(\alpha)$.

*Proof.* The proof is by induction on the structure of $\alpha$. For $\emptyset$ and $\varepsilon$ the result is trivially true. For $\sigma \in \Sigma$, $\bigcup_{\sigma'\in\Sigma} \sigma'\partial_{\sigma'}(\sigma) \cup \varepsilon(\sigma) = \{\sigma\} = \mathsf{L}_\varepsilon(\sigma)$. For the remaining cases it is sufficient to apply the definitions of $\partial_\sigma(\alpha)$ and $\mathsf{L}_\varepsilon(\alpha)$, and the fact that $\mathsf{L}_\varepsilon(\alpha) = \bigcup_{\sigma\in\Sigma} \sigma\partial_\sigma(\alpha) \cup \varepsilon(\alpha)$ implies $\mathsf{L}(\alpha) = \bigcup_{\sigma\in\Sigma} \sigma\partial_\sigma(\alpha)$. □

Given $X, X' \subseteq \mathsf{PD}(\alpha)$ we define

$$X \equiv_{\mathsf{L}_\varepsilon} X' \Leftrightarrow \mathsf{L}_\varepsilon(X) = \mathsf{L}_\varepsilon(X'),$$

where $\mathsf{L}_\varepsilon(X)$ denotes $\bigcup_{\alpha'\in X} \mathsf{L}_\varepsilon(\alpha')$.

**Proposition 19.** $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha))/{\equiv_{\mathsf{L}_\varepsilon}} \simeq \mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$.

*Proof.* Let $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha)) = \langle Q_{\mathrm{D}(\mathrm{PD})}, \Sigma, \delta_{\mathrm{D}(\mathrm{PD})}, \{\alpha\}, F_{\mathrm{D}(\mathrm{PD})}\rangle$. We first show that $\equiv_{\mathsf{L}_\varepsilon}$ is right-invariant w.r.t. $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha))$. We have

$$\equiv_{\mathsf{L}_\varepsilon} \subseteq (Q_{\mathrm{D}(\mathrm{PD})} \setminus F_{\mathrm{D}(\mathrm{PD})})^2 \cup F^2_{\mathrm{D}(\mathrm{PD})},$$

because $\varepsilon \in \mathsf{L}_\varepsilon(X)$ if and only if there exists $\alpha' \in X$ such that $\varepsilon(\alpha') = \varepsilon$. Now, consider two states $X$ and $X'$ in $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha))$ such that $X \equiv_{\mathsf{L}_\varepsilon} X'$. Then

$$\begin{aligned}
\delta_{\mathrm{D}(\mathrm{PD})}(X, \sigma) &= \bigcup_{\alpha'\in X} \partial_\sigma(\alpha') \\
&= \{\, \alpha'' \mid \sigma\alpha'' \in \mathsf{L}_\varepsilon(X) \,\} \\
&= \{\, \alpha'' \mid \sigma\alpha'' \in \mathsf{L}_\varepsilon(X') \,\} \\
&= \delta_{\mathrm{D}(\mathrm{PD})}(X', \sigma).
\end{aligned}$$

Consider $\varphi_{\mathrm{N}} : Q_{\mathrm{D}(\mathrm{PD})}/{\equiv_{\mathsf{L}_\varepsilon}} \longrightarrow Q_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}$ defined by $\varphi_{\mathrm{N}}([X]) = \mathsf{L}_\varepsilon(X)$. We want to show that $\varphi_{\mathrm{N}}$ defines an isomorphism between $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha))/{\equiv_{\mathsf{L}_\varepsilon}}$ and $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$. Injectivity follows from the definition of $\equiv_{\mathsf{L}_\varepsilon}$. We have $\varphi_{\mathrm{N}}([\{\alpha\}]) = \mathsf{L}_\varepsilon(\alpha)$ and $[X]$ is a final state of $\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha))/{\equiv_{\mathsf{L}_\varepsilon}}$ if and only if $\varepsilon \in \mathsf{L}_\varepsilon(X) =$

$\varphi_{\mathrm{N}}([X])$. For the transition functions, let $X \in Q_{\mathrm{D(PD)}}/{\equiv_{\mathsf{L}_\varepsilon}}$ and $\sigma \in \Sigma$,

$$\begin{aligned}
\varphi_{\mathrm{N}}(\delta_{\mathrm{D(PD)}}/{\equiv_{\mathsf{L}_\varepsilon}}([X],\sigma)) &= \varphi_{\mathrm{N}}([\{\, \alpha'' \mid \sigma\alpha'' \in \mathsf{L}_\varepsilon(X) \,\}]) \\
&= \mathsf{L}_\varepsilon(\{\, \alpha'' \mid \sigma\alpha'' \in \mathsf{L}_\varepsilon(X) \,\}) \\
&= \bigcup_{\sigma\alpha'' \in \mathsf{L}_\varepsilon(X)} \mathsf{L}_\varepsilon(\alpha'') \\
&= \delta_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}(\mathsf{L}_\varepsilon(X),\sigma) \\
&= \delta_{\mathrm{D}(\overleftarrow{\mathrm{Pre}})}(\varphi_{\mathrm{N}}([X]),\sigma).
\end{aligned}$$

To prove that $\varphi_{\mathrm{N}}$ is surjective note that any state of $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$ is a $\mathsf{L}_\varepsilon(X)$ for $X \in Q_{\mathrm{D(PD)}}$.

□

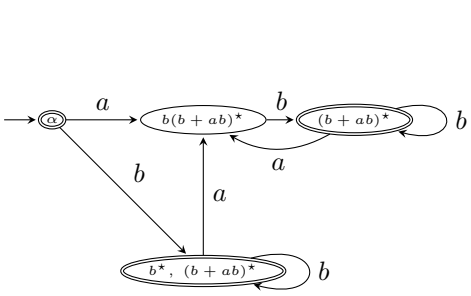
Asperti et al. [3] showed that quotients of the Brzozowski automaton $\mathcal{A}_{\mathrm{B}}(\alpha)$ and of the *mark before* automaton $\mathcal{A}_{\mathrm{MB}}(\alpha)$, by two different right-invariant relations, lead to two isomorphic automata. For $\mathcal{A}_{\mathrm{B}}(\alpha)$ that relation is precisely ${\equiv_{\mathsf{L}_\varepsilon}}$. Adapting the arguments in the proof of Proposition 19 and using the fact that $\mathsf{L}_\varepsilon(\mathrm{d}_\sigma(\alpha)) = \mathsf{L}_\varepsilon(\mathrm{d}_\sigma(\mathsf{L}_\varepsilon(\alpha)))$, which was shown in [3], we obtain that those isomorphic automata are also isomorphic to $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$, i.e.,

**Corollary 3.** $\mathcal{A}_{\mathrm{B}}(\alpha)/{\equiv_{\mathsf{L}_\varepsilon}} \simeq \mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$.

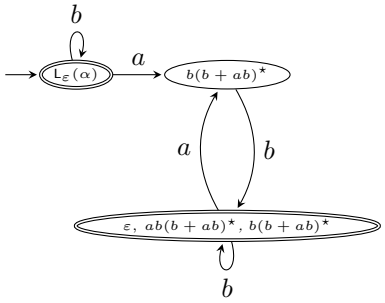
**Example 10.** *Consider the expression* $\alpha = (b+ab)^\star + b^\star$ *from Example 1 and* $\mathcal{A}_{\mathrm{PD}}(\alpha)$ *presented in Example 8. We have*

$$\mathsf{L}_\varepsilon(\alpha) = \{\varepsilon, bb^\star, b(b+ab)^\star, ab(b+ab)^\star\} = \mathsf{L}_\varepsilon(\{(b+ab)^\star, b^\star\})$$

*and the following automata:*




$\mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha))$




$\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$

*For this expression one has* $\mathcal{A}_{\mathrm{B}}(\alpha) \simeq \mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$.

## 6. Reversals and Automata Constructions

Given $\alpha$, any of the automata constructions in the previous sections can be applied to $\alpha^{\mathrm{R}}$. If one reverses the resulting automaton, an alternative automaton construction for $\mathcal{L}(\alpha)$ is obtained.

In this section we establish some relations between the direct constructions and the double reversed ones. We show that $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)$ and obtain analogous relations for some of the quotients of $\mathcal{A}_{\mathrm{POS}}$. Then we show that determinising any quotient of $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}$ by a right-invariant relation is the same as determinising $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}$ and thus, by Proposition 10, the resulting automata are all isomorphic to $\mathcal{A}_{\mathrm{MB}}$. The same does not hold if one considers quotients by a left-invariant relation, and we illustrate that with the $\mathcal{A}_{\mathrm{Pre}}$ construction.

The following lemma establishes the relation between positions of letters in $\overline{\alpha}$ and the positions of the corresponding occurrences in $\overline{\alpha^{\mathrm{R}}}$.

**Lemma 20.** *Consider a regular expression $\alpha$ with $|\alpha|_\Sigma = n$ and the bijection $\varphi_{\mathrm{P}} : \mathsf{Pos}_0(\alpha) \longrightarrow \mathsf{Pos}(\alpha) \cup \{n+1\}$ defined by $\varphi_{\mathrm{P}}(i) = n+1-i$. Then, $\varphi_{\mathrm{P}}(\varphi_{\mathrm{P}}(i)) = i$, $\varphi_{\mathrm{P}}(\mathsf{Pos}(\alpha)) = \mathsf{Pos}(\alpha)$ and $\varphi_{\mathrm{P}}(0) = n+1$. Furthermore, the following hold for $i, j \in \mathsf{Pos}(\alpha)$:*

- $i \in \mathsf{Last}(\alpha)$ *iff* $\varphi_{\mathrm{P}}(i) \in \mathsf{First}(\alpha^{\mathrm{R}})$;
- $i \in \mathsf{First}(\alpha)$ *iff* $\varphi_{\mathrm{P}}(i) \in \mathsf{Last}(\alpha^{\mathrm{R}})$;
- $j \in \mathsf{Follow}(\alpha, i)$ *iff* $\varphi_{\mathrm{P}}(i) \in \mathsf{Follow}(\alpha^{\mathrm{R}}, \varphi_{\mathrm{P}}(j))$;
- $\overline{\sigma_i}$ *in* $\overline{\alpha}$ *is the same letter as* $\overline{\sigma_{\varphi_{\mathrm{P}}(i)}}$ *in* $\overline{\alpha^{\mathrm{R}}}$.

Our first result on reversals of expressions and automata reads as follows:

**Proposition 21.** $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)$.

*Proof.* Consider $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}} = \langle \mathsf{Pos}_0(\alpha^{\mathrm{R}}), \Sigma, \delta^{\mathrm{R}}_{\mathrm{POS}}, \mathsf{Last}_0(\alpha^{\mathrm{R}}), \{0\} \rangle$ where

$$\delta^{\mathrm{R}}_{\mathrm{POS}} = \{\, (i, \overline{\sigma_i}, j) \mid i \in \mathsf{Follow}(\alpha^{\mathrm{R}}, j),\ j \neq 0 \,\} \cup \{\, (i, \overline{\sigma_i}, 0) \mid i \in \mathsf{First}(\alpha^{\mathrm{R}}) \,\}.$$

We show that $\varphi_{\mathrm{P}}$, as defined in Lemma 20, is an isomorphism between the automata $\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}$ and $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}(\alpha)$. For the transition relations we have,

$$\begin{aligned}
&\varphi_{\mathrm{P}}(\delta^{\mathrm{R}}_{\mathrm{POS}}) \\
&\quad = \quad \varphi_{\mathrm{P}}(\{\, (i, \overline{\sigma_{\varphi_{\mathrm{P}}(i)}}, j) \mid \varphi_{\mathrm{P}}(j) \in \mathsf{Follow}(\alpha, \varphi_{\mathrm{P}}(i)),\ \varphi_{\mathrm{P}}(j) \neq n+1 \,\}) \\
&\qquad\qquad \cup\, \varphi_{\mathrm{P}}(\{\, (i, \overline{\sigma_{\varphi_{\mathrm{P}}(i)}}, 0) \mid \varphi_{\mathrm{P}}(i) \in \mathsf{Last}(\alpha) \,\}) \\
&\quad = \quad \{\, (\varphi_{\mathrm{P}}(i), \overline{\sigma_{\varphi_{\mathrm{P}}(i)}}, \varphi_{\mathrm{P}}(j)) \mid \varphi_{\mathrm{P}}(j) \in \mathsf{Follow}(\alpha, \varphi_{\mathrm{P}}(i)),\ \varphi_{\mathrm{P}}(j) \neq n+1 \,\} \\
&\qquad\qquad \cup\, \{\, (\varphi_{\mathrm{P}}(i), \overline{\sigma_{\varphi_{\mathrm{P}}(i)}}, n+1) \mid \varphi_{\mathrm{P}}(i) \in \mathsf{Last}(\alpha) \,\}) \\
&\quad = \quad \delta_{\overleftarrow{\mathrm{POS}}}.
\end{aligned}$$

Showing that $\varphi_{\mathrm{P}}$ preserves initial and final states is easy and left to the reader. □

A right-invariant relation $\equiv$ on the position automaton corresponds to a left-invariant relation in its dual, $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$. In particular, we have $\mathcal{A}_{\mathrm{PD}}(\alpha^{\mathrm{R}}) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})/\equiv_c$ and consequently $\mathcal{A}_{\mathrm{PD}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}/\equiv_c$. Moreover, Maia

et al. [18] showed that $\mathcal{A}_{\mathrm{PD}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}^{\leftarrow}_{\mathrm{PD}}(\alpha)$. For the follow automaton construction it is also true that $\mathcal{A}_{\mathrm{F}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}/\equiv_{\mathrm{F}}$. So, this automaton, which we denote by $\mathcal{A}^{\leftarrow}_{\mathrm{F}}$ can also be defined as a quotient of $\mathcal{A}^{\leftarrow}_{\mathrm{POS}}$ by a left-invariant relation.

If we consider a quotient of the position automaton by a left-invariant relation $\equiv$, its dual construction is a quotient of $\mathcal{A}^{\leftarrow}_{\mathrm{POS}}$ by $\equiv$, which is a right-invariant relation on $\mathcal{A}^{\leftarrow}_{\mathrm{POS}}$. In particular, $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}}) \simeq \mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})/\equiv_{\ell}$, where $\equiv_{\ell}$ is left-invariant. Consequently, $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}/\equiv_{\ell}$, where $\equiv_{\ell}$ is right-invariant. In the following we show that $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$ automaton. For this, we consider the following lemma, which relates the states in $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})$ with the states in $\mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$. Given a set $S$ of regular expressions, let $S^{\mathrm{R}} = \{\, \alpha^{\mathrm{R}} \mid \alpha \in S \,\}$.

**Lemma 22.** *For all $\alpha \in \mathrm{RE}$, one has $\mathsf{R}(\alpha^{\mathrm{R}})^{\mathrm{R}} = \mathsf{L}(\alpha)$.*

*Proof.* The proof is by structural induction on $\alpha$, for which we only show the case of concatenation. Let

$$\begin{aligned}
\mathsf{L}(\alpha_1\alpha_2) &= \mathsf{L}(\alpha_1)\alpha_2 \cup \varepsilon(\alpha_1)\,\mathsf{L}(\alpha_2) \\
&= \mathsf{R}(\alpha_1^{\mathrm{R}})^{\mathrm{R}}\alpha_2 \cup \varepsilon(\alpha_1)\,\mathsf{R}(\alpha_2^{\mathrm{R}})^{\mathrm{R}} \\
&= (\alpha_2^{\mathrm{R}}\,\mathsf{R}(\alpha_1^{\mathrm{R}}))^{\mathrm{R}} \cup (\varepsilon(\alpha_1^{\mathrm{R}})\,\mathsf{R}(\alpha_2^{\mathrm{R}}))^{\mathrm{R}} \\
&= (\alpha_2^{\mathrm{R}}\,\mathsf{R}(\alpha_1^{\mathrm{R}}) \cup \varepsilon(\alpha_1^{\mathrm{R}})\,\mathsf{R}(\alpha_2^{\mathrm{R}}))^{\mathrm{R}} \\
&= \mathsf{R}(\alpha_2^{\mathrm{R}}\alpha_1^{\mathrm{R}})^{\mathrm{R}} = \mathsf{R}((\alpha_1\alpha_2)^{\mathrm{R}})^{\mathrm{R}}.
\end{aligned}$$

The remaining operations can be shown with similar arguments. □

As a consequence, one also has $\mathsf{R}_{\varepsilon}(\alpha^{\mathrm{R}})^{\mathrm{R}} = \mathsf{L}_{\varepsilon}(\alpha)$. Now it is sufficient to note that the initial state of $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})$ is $\varepsilon$, which is the unique finite state of $\mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$. The set of final states in $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})$ is $\mathsf{R}_{\varepsilon}(\alpha^{\mathrm{R}})$, and the set of initial states in $\mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$ is $\mathsf{L}_{\varepsilon}(\alpha)$. Furthermore, there is a transition entering a state of the form $\alpha'\sigma$ from a state $\alpha''$ in $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})$ if and only if $\alpha'' \in \mathsf{R}_{\varepsilon}(\alpha')$ and that transition is labelled by $\sigma$. On the other hand, there is a transition from a state of the form $\alpha'\sigma$ to a state $\alpha''$ in $\mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$ if and only if $\alpha'' \in \mathsf{L}_{\varepsilon}(\alpha')$ and that transition is labelled by $\sigma$. We conclude that $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}}$ is isomorphic to $\mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$.

**Proposition 23.** $\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}} \simeq \mathcal{A}^{\leftarrow}_{\mathrm{Pre}}(\alpha)$.

Now we consider, the determinisation of the above constructions. For $\mathcal{A}^{\leftarrow}_{\mathrm{POS}}$, the determinisation of any quotient by a left-invariant relation is isomorphic to $\mathcal{A}_{\mathrm{MB}}$, which is a direct consequence of Lemma 2 and of the fact that a right-invariant relation in an NFA $A$ is a left-invariant relation in $A^{\mathrm{R}}$. This has direct consequences for all constructions, that can be obtained as a quotient of the position automaton by some right-invariant relation. In particular, we have the following.

**Proposition 24.**

$$\mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}) \simeq \mathrm{D}(\mathcal{A}_{\mathrm{PD}}(\alpha^{\mathrm{R}})^{\mathrm{R}}) \simeq \mathrm{D}(\mathcal{A}_{\mathrm{F}}(\alpha^{\mathrm{R}})^{\mathrm{R}}) \simeq \mathcal{A}_{\mathrm{MB}}(\alpha).$$

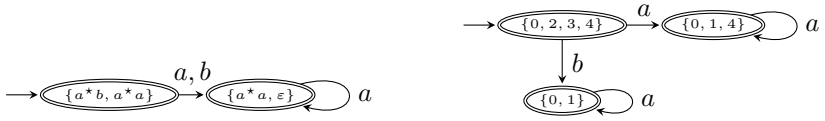
However, if we consider quotients of the position automaton by a left-invariant relation $\equiv$, the determinisation of its dual construction is a quotient of $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$ by $\equiv$, which is a right-invariant relation on $\mathcal{A}_{\overleftarrow{\mathrm{POS}}}$. By Lemma 1, we have for $\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}$ that

**Proposition 25.**

$$\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha)) \simeq \mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}}) \simeq \mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}}/{\equiv_\ell}) \simeq \mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}})/{\equiv_\ell}.$$

From this result and the relations obtained in Section 5.1, we conclude that $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$ is the smallest automaton among the constructions studied in this paper.

**Example 11.** *For $\alpha = a^\star + (a+b)a^\star$, where $\overline{\alpha^{\mathrm{R}}} = a_1^\star(a_2 + b_3) + a_4^\star$, we have $\mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}}) \not\simeq \mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$, which are depicted below.*




$\mathrm{D}(\mathcal{A}_{\mathrm{Pre}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$ $\qquad$ $\mathrm{D}(\mathcal{A}_{\mathrm{POS}}(\alpha^{\mathrm{R}})^{\mathrm{R}})$

## 7. Taxonomy

Recall Figure 1, where the relations between the different automata studied are graphically represented. We highlight that $\mathrm{D}(\mathcal{A}_{\overleftarrow{\mathrm{Pre}}}(\alpha))$ is the smallest automaton among the deterministic ones, but is not always the minimal. Brzozowski [9] showed that for a trim NFA $\mathcal{A}$, $\mathrm{D}(\mathcal{A})$ is minimal if $\mathcal{A}^{\mathrm{R}}$ is deterministic. Consequently, one obtains the nice property that, whenever $X(\alpha)$ is a deterministic automaton, for instance $\mathcal{A}_{\mathrm{MB}}$ or $\mathcal{A}_{\mathrm{MA}}$, then $\mathrm{D}(X(\alpha^{\mathrm{R}})^{\mathrm{R}})$ is the minimal DFA for $\mathcal{L}(\alpha)$. We note that the size of $\mathcal{A}_{\mathrm{B}}$, $\mathrm{D}(\mathcal{A}_{\mathrm{PD}})$ and $\mathcal{A}_{\mathrm{MB}}$ are incomparable. Some examples follow. For $a^\star b^\star + a^\star bbb$, automaton $\mathcal{A}_{\mathrm{B}}$ is smaller than $\mathrm{D}(\mathcal{A}_{\mathrm{PD}})$, and for $a^\star(a + (aa))^\star$, automaton $\mathrm{D}(\mathcal{A}_{\mathrm{PD}})$ is the smaller. For $aa + bba$ we find that $\mathrm{D}(\mathcal{A}_{\mathrm{PD}})$ is smaller than $\mathcal{A}_{\mathrm{MB}}$. Finally, for $(b((a+a)+a^\star))b$, the device $\mathcal{A}_{\mathrm{MB}}$ is smaller than $\mathcal{A}_{\mathrm{B}}$ and than $\mathrm{D}(\mathcal{A}_{\mathrm{PD}})$.

## References


[1] Allauzen, C., Mohri, M., 2006. A unified construction of the Glushkov, follow, and Antimirov automata. In: Kralovic, R., Urzyczyn, P. (Eds.), 31st MFCS. Vol. 4162 of LNCS. Springer, pp. 110–121.

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

[3] Asperti, A., Coen, C. S., Tassi, E., 2010. Regular expressions, au point. CoRR abs/1010.2604.
URL http://arxiv.org/abs/1010.2604

[4] Berry, G., Sethi, R., 1986. From regular expressions to deterministic automata. Theoret. Comput. Sci. 48 (3), 117–126.

[5] Broda, S., Holzer, M., Maia, E., Moreira, N., Reis, R., 2017. On the mother of all automata: the position automaton. In: Charlier, É., Leroy, J., Rigo, M. (Eds.), Proc. 21st DLT (Developments in Language Theory). Vol. 10396 of LNCS. Springer, pp. 134–146.

[6] Broda, S., Machiavelo, A., Moreira, N., Reis, R., 2012. On the average size of Glushkov and partial derivative automata. Internat. J. Found. Comput. Sci. 23 (5), 969–984.

[7] Brüggemann-Klein, A., 1993. Regular expressions into finite automata. Theoret. Comput. Sci. 48, 197–213.

[8] Brzozowski, J., 1964. Derivatives of regular expressions. J. ACM (11), 481–494.

[9] Brzozowski, J. A., 1962. Canonical regular expressions and minimal state graphs for definite events. In: Mathematical Theory of Automata. MRI Symposia Series. Polytechnic Press, Polytechnic Institute of Brooklyn, NY,12, pp. 529–561.

[10] Champarnaud, J. M., Ziadi, D., 2001. From Mirkin's prebases to Antimirov's word partial derivatives. Fundam. Inform. 45 (3), 195–205.

[11] Champarnaud, J. M., Ziadi, D., 2002. Canonical derivatives, partial derivatives and finite automaton constructions. Theoret. Comput. Sci. 289, 137–163.

[12] Chen, H., Yu, S., 2012. Derivatives of regular expressions and an application. In: Dinneen, M. J., Khoussainov, B., Nies, A. (Eds.), Computation, Physics and Beyond, WTCS 2012. Vol. 7160 of LNCS. Springer, pp. 343–356.

[13] Giammarresi, D., Ponty, J.-L., Wood, D., 1998. Glushkov and Thompson constructions: A synthesis. HKUST TCSC-98-11, The Department of Science & Engineering, Theoretical Cmputer Science Group, The Hong Kong University of Science and Technology.

[14] Glushkov, V. M., 1961. The abstract theory of automata. Russian Math. Surveys 16, 1–53.

[15] Gruber, H., Holzer, M., Dec. 2015. From finite automata to regular expressions and back—a summary on descriptional complexity. Internat. J. Found. Comput. Sci. 26 (8), 1009–1040.

[16] Ilie, L., Yu, S., 2003. Follow automata. Inf. Comput. 186 (1), 140–162.

[17] Maia, E., 2015. On the descriptional complexity of some operations and simulations of regular models. Ph.D. thesis, Faculdade de Ciências da Universidade do Porto.

[18] Maia, E., Moreira, N., Reis, R., 2015. Prefix and right-partial derivative automata. In: Soskova, M., Mitrana, V. (Eds.), 11th CiE. Vol. 9136 of LNCS. Springer, pp. 258–267.

[19] McNaughton, R., Yamada, H., 1960. Regular expressions and state graphs for automata. IEEE Trans. Comput. 9, 39–47.

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

[21] Nipkow, T., Traytel, D., 2014. Unified decision procedures for regular expression equivalence. In: Klein, G., Gamboa, R. (Eds.), 5th ITP. Vol. 8558 of LNCS. Springer, pp. 450–466.

[22] Ott, G., Feinstein, N. H., Oktober 1961. Design of sequential machines from their regular expressions. J. ACM 8 (4), 585–600.

[23] Sakarovitch, J., 2009. Elements of Automata Theory. Cambridge University Press.

[24] Thompson, K., 1968. Regular expression search algorithm. Commun. ACM 11 (6), 410–422.

[25] Yamamoto, H., 2014. A new finite automaton construction for regular expressions. In: Bensch, S., Freund, R., Otto, F. (Eds.), 6th NCMA. Vol. 304 of books@ocg.at. Österreichische Computer Gesellschaft, pp. 249–264.