The Alternating Normal Form of Braids and Its Minimal Automaton

Vincent Jugé
LIGM, CNRS & Univ Gustave Eiffel, Marne-la-Vallée, France
IRIF, CNRS & Université Paris Cité, France

June Roupin
LIGM, CNRS & Univ Gustave Eiffel, Marne-la-Vallée, France

Abstract

The alternating normal form of braids is a well-known normal form on standard braid monoids. This normal form is regular: the language it identifies with is regular. We give a characterisation of the minimal automaton of this language and compute its size, both in terms of number of states and of transitions, depending on the number of generators of the monoid.

2012 ACM Subject Classification Mathematics of computing → Enumeration

Keywords and phrases Automata, braids, enumeration, normal forms

Digital Object Identifier 10.4230/LIPIcs.AofA.2024.23

1 Introduction

The group of braids with n strands, commonly denoted by B_n, is the group of isotopy classes of geometric braids with n strands. In [2], E. Artin proved that this group enjoyed the following finite presentation:

$$B_n = \left\langle \sigma_1, \dots, \sigma_{n-1} \;\middle|\; \begin{array}{l} \sigma_i\sigma_j = \sigma_j\sigma_i \text{ when } j \geqslant i+2 \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \end{array} \right\rangle .$$

The relations $\sigma_i\sigma_j = \sigma_j\sigma_i$ are called *commutation* relations; relations $\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}$ are called *braid* relations. They come with the *simplification* relations $\sigma_i\sigma_i^{-1} = \varepsilon$, where ε denotes the neutral element of the group. These three kinds of relations are illustrated in Figure 1. Braid elements $\sigma_1, \dots, \sigma_{n-1}$ are called *Artin generators*.

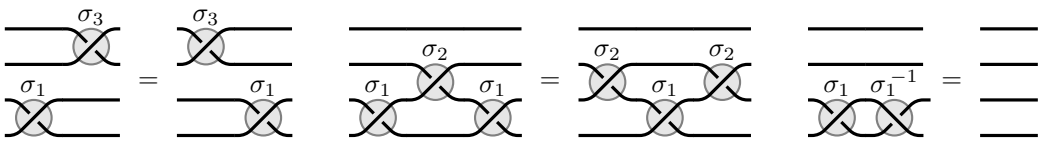

Figure 1 Commutation relation $\sigma_1\sigma_3 = \sigma_3\sigma_1$, braid relation $\sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2$ and simplification relation $\sigma_1\sigma_1^{-1} = \varepsilon$. The latter relation is valid only in the group B_n.

Braid groups enjoy numerous algebraic, combinatorial and geometric properties, many of which are connected with the study of the (standard) braid monoid B_n^+: this is the monoid positively generated by the generators σ_i, i.e., the least subset of B_n containing generators $\sigma_1, \dots, \sigma_{n-1}$ (but not their inverses) and stable by product:

$$B_n^+ = \left\langle \sigma_1, \dots, \sigma_{n-1} \;\middle|\; \begin{array}{l} \sigma_i\sigma_j = \sigma_j\sigma_i \text{ when } j \geqslant i+2 \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \end{array} \right\rangle^+ .$$

Here are some properties of the monoid B_n^+ [1, 4, 6, 8, 12, 14]:

a) The braid monoid B_n^+ is simplifiable: whenever $\alpha\beta\gamma = \alpha\beta'\gamma$, we have $\beta = \beta'$.

b) The left-divisibility ordering, defined by $\alpha \leqslant_{\mathsf{L}} \beta$ whenever there exists a braid $\gamma \in B_n^+$ (also denoted by $\alpha^{-1}\beta$) such that $\alpha\gamma = \beta$, is a lattice: any two elements α and β have a greatest common divisor $\alpha \wedge \beta$ and a least common multiple $\alpha \vee \beta$.

c) Similarly, the right-divisibility ordering, defined by $\beta \geqslant_{\mathsf{R}} \alpha$ whenever there exists a braid $\gamma \in B_n^+$ (also denoted by $\beta\alpha^{-1}$) such that $\beta = \gamma\alpha$, is a lattice.

d) The braid $\Delta_n = \sigma_1(\sigma_2\sigma_1)(\sigma_3\sigma_2\sigma_1)\cdots(\sigma_{n-1}\sigma_{n-2}\cdots\sigma_2\sigma_1)$, called the *Garside element* of the monoid B_n^+, is the least common multiple of the family of Artin generators for both the left- and right-divisibility orderings. Both its left and its right divisors coincide with the set of positive braids in which any two strands cross each other at most once; such braids are called *simple* braids. Furthermore, Δ_n obeys the relations $\Delta_n\sigma_i = \sigma_{n-i}\Delta_n$, which makes the inner automorphism $\phi_n\colon \beta \mapsto \Delta_n^{-1}\beta\Delta_n$ an involution of B_n^+.

e) The function $\mathsf{s}_n\colon \beta \mapsto \beta \wedge \Delta_n$, which selects the largest simple left divisor of a braid β, obeys the identity $\mathsf{s}_n(\alpha\beta) = \mathsf{s}_n(\alpha\mathsf{s}_n(\beta))$.

Properties **a)** to **c)** give rise to recursive decompositions, some of which we will focus on below: Property **a)** allows factoring a braid into factors on which we will be able to work independently, and Properties **b)** and **c)** will allow, under some conditions, to select the *largest* divisor of a braid that belongs to a given set. For instance, the following result is a consequence of Properties **a)** and **b)**:

f) The submonoid B_{n-1}^+, generated by $\sigma_1, \ldots, \sigma_{n-2}$ and called a *parabolic* submonoid of B_n^+, is a sub-lattice of B_n^+. Thus, each braid $\beta \in B_n^+$ has a largest left divisor in B_{n-1}^+, denoted by $\mathsf{h}_n(\beta)$ and called the *n-head* of β: its left divisors are the left divisors of β that belong to B_{n-1}^+. The corresponding right divisor $\mathsf{h}_n(\beta)^{-1}\beta$, denoted by $\mathsf{b}_n(\beta)$, is called the *n-body* of β.

The effect of the functions s_4, h_4 and b_4 and of the automorphism ϕ_4 on the braid $\beta = \sigma_2\sigma_3\sigma_3\sigma_1$ is illustrated in Figure 2: s_4 selects the largest simple left divisor of β, which is not necessarily a prefix of the representation of β we started from; h_4 selects the largest left divisor of β that can be written without using the generator σ_3, and b_4 selects the corresponding right divisor; finally, ϕ_4 replaces each generator σ_i of β by σ_{4-i}.

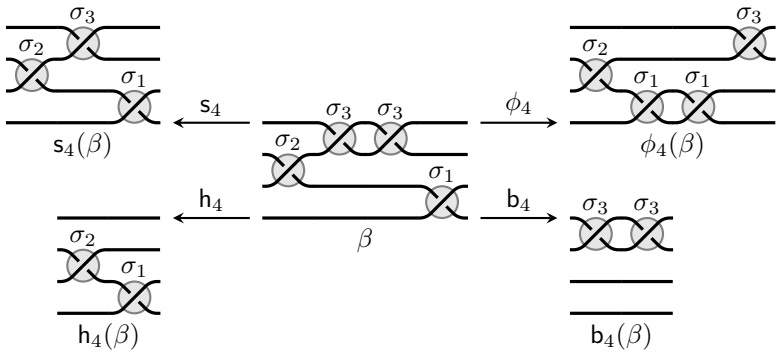

Figure 2 Applying s_4, h_4, b_4 and ϕ_4 to the braid $\beta = \sigma_2\sigma_3\sigma_3\sigma_1$.

The above presentation identifies each braid $\beta \in B_n^+$ with an equivalence class of words over the alphabet $\mathbb{A}_n = \{\sigma_1, \ldots, \sigma_{n-1}\}$. A *normal form* is then a language containing exactly one word $\mathsf{NF}(\beta)$ in each equivalence class β, which will be a preferred representative of β.

For a normal form to be useful, the following tasks should be as easy as possible [8]:

- deciding whether a word w belongs to the normal form;
- transforming a word w representing a braid β into the word $\mathsf{NF}(\beta)$;
- computing, given two words $w = \mathsf{NF}(\beta)$ and $w' = \mathsf{NF}(\beta')$, the word $\mathsf{NF}(\beta\beta')$.

In this article, we focus on the first question, for which a possible answer is: "the normal form should be a regular set, and its minimal automaton should be small".

In that context, among a plethora of other normal forms, let us mention three similar normal forms on braids: the Garside normal form [7], the lexicographically minimal normal form [13] and the alternating normal form [3, 5]: the Garside normal form is the most well-known normal form on braid monoids, and all three are regular.

▶ **Definition 1.** *The Garside normal form of a braid $\beta \in B_n^+$ is inductively defined as the following factorisation of β into simple braids: we set $\mathsf{Gar}_n(\beta) = \beta$ when β is simple, and $\mathsf{Gar}_n(\beta) = \mathsf{s}_n(\beta)\mathsf{Gar}_n(\mathsf{s}_n(\beta)^{-1}\beta)$ otherwise. If necessary, each simple divisor can then be written as a product of generators σ_i.*

▶ **Definition 2.** *The lexicographically minimal normal form of a braid $\beta \in B_n^+$ is denoted by $\mathsf{LexMin}_n(\beta)$. It is the word representing the braid $\beta \in B_n^+$ that is minimal for the lexicographic ordering induced by the ordering $\sigma_1 < \sigma_2 < \cdots < \sigma_{n-1}$ on Artin generators.*

Alternatively, the word $\mathsf{LexMin}_n(\beta)$ may be inductively defined by $\mathsf{LexMin}_2(\sigma_1^k) = \sigma_1^k$ or, if $n \geqslant 3$, by $\mathsf{LexMin}_n(\beta) = \mathsf{LexMin}_{n-1}(\beta)$ when $\beta \in B_{n-1}^+$, and

$$\mathsf{LexMin}_n(\beta) = \mathsf{LexMin}_{n-1}(\mathsf{h}_n(\beta))\sigma_{n-1}\mathsf{LexMin}_n((\mathsf{h}_n(\beta)\sigma_{n-1})^{-1}\beta)$$

otherwise.

▶ **Definition 3.** *The alternating normal form of a braid $\beta \in B_n^+$ is denoted by $\mathsf{Alt}_n(\beta)$. It is inductively defined by $\mathsf{Alt}_2(\sigma_1^k) = \sigma_1^k$ or, if $n \geqslant 3$, by $\mathsf{Alt}_n(\beta) = \mathsf{Alt}_{n-1}(\beta)$ when $\beta \in B_{n-1}^+$, and $\mathsf{Alt}_n(\beta) = \mathsf{Alt}_{n-1}(\mathsf{h}_n(\beta))\phi_n(\mathsf{Alt}_n(\phi_n(\mathsf{b}_n(\beta))))$ otherwise*[1]*.*

This normal form is tightly connected to the rotating normal form [11], a similar normal form defined on the *dual* braid monoid B_n^{+*}, which is the sub-monoid of B_n positively generated by braids of the form $\sigma_u\sigma_{u-1}\cdots\sigma_{v+1}\sigma_v\sigma_{v+1}^{-1}\sigma_{v+2}^{-1}\cdots\sigma_u^{-1}$. Below, we study the alternating normal form, by constructing explicitly its minimal automaton and counting its states and transitions.

Figure 3 presents the three representatives of the braid $\beta = \sigma_3\sigma_2\sigma_2\sigma_3\sigma_1\sigma_1\sigma_3$ that belong to the Garside, lexicographically minimal and alternating normal forms, illustrating that these three words may all differ from each other.

Property **e)** provides us with a co-deterministic automaton that, once given as input a word $w = w_0w_1\cdots w_{k-1}$ representing a braid β, computes at each step the braid $\mathsf{s}_n(w_{\geqslant i})$, where $w_{\geqslant i} = w_iw_{i+1}\cdots w_{k-1}$: indeed, it suffices to observe that $\mathsf{s}_n(w_{\geqslant i}) = \mathsf{s}_n(w_i\mathsf{s}_n(w_{\geqslant i+1}))$, and to precompute s_n on braids of the form $\sigma_i\gamma$, where σ_i is an Artin generator and γ is simple. This automaton itself helps proving that the three above normal forms are regular: we check that w is in

1) Garside normal form by verifying that it starts with a prefix $w_{<i} = w_0w_1\cdots w_{i-1}$ representing $\mathsf{s}_n(w)$, and then that $w_{\geqslant i}$ is in Garside normal form;

[1] One may often find a *mirrored* version of this normal form, in which, instead of extracting the largest left divisor of β in B_{n-1}^+, one extracts the largest right divisor. The languages induced by both versions are mirrors of each other.

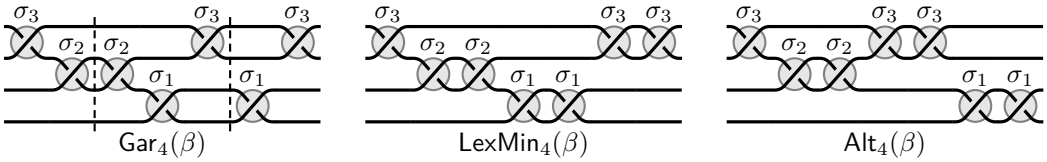

Figure 3 Normal forms of the braid $\beta = \sigma_3\sigma_2\sigma_2\sigma_3\sigma_1\sigma_1\sigma_3$. Vertical dashed bars separate the simple braids $\sigma_3\sigma_2$, $\sigma_2\sigma_3\sigma_1$ and $\sigma_1\sigma_3$ into which β was factored to give its Garside normal form.

2) lexicographically minimal normal form by verifying that each letter w_i is the least Artin generator that left-divides $\mathsf{s}_n(w_{\geqslant i})$;
3) alternating normal form by finding the smallest index i such that $w_i = \sigma_{n-1}$ (if any), verifying that σ_{n-1} is the only Artin generator left-dividing $\mathsf{s}_n(w_{\geqslant i})$, and verifying that $w_{<i}$ and $\phi_n(w_{\geqslant i})$ are in alternating normal form.

Similar arguments would prove that the rotating normal form is also regular.

Although such observations yield automata recognising the three above normal forms, these automata are non-deterministic, and determinising them might result in unreasonably large deterministic automata. Explicit minimal automata for the lexcicographically minimal normal form were constructed in [9]; there, it is proved that the minimal automaton recognising $\mathsf{LexMin}_n(B_n^+)$ has $2F_{2n+1} - n(n+1)/2 - 2 \approx 2\Phi^{2n+1}/\sqrt{5}$ states, where F_k is the k^{th} Fibonacci number and $\Phi = (1+\sqrt{5})/2$ is the Golden Ratio. Non-necessarily minimal automata for the rotating normal form were constructed in [11]. The minimal automaton of the Garside normal form, viewed as a language over the alphabet $\mathsf{s}_n(B_n^+)$ of simple braids, has 2^{n-1} states and ob_n transitions, where ob_n is the n^{th} *ordered Bell number*. Finally, the minimal automata for the Garside normal form (viewed as a language over $\mathbb{A}_n$, choosing a canonical representative of each simple braid), for the alternating normal form and for the rotating normal form have net yet been investigated.

In this article, we prove the following results.

▶ **Theorems 19 & 20.** *The minimal automaton of the language* $\mathsf{Alt}_n(B_n^+)$ *is an explicit automaton* $\mathscr{A}_n$ *with* s_n *states and* t_n *transitions, where* $s_1 = 1$, $t_1 = 0$, *and*

$$s_n = \frac{25 \times 2^{2n-3} - 9n^2 + 3n + 7}{27} \text{ and } t_n = \frac{(225n - 290)2^{2n-5} - 9n^3 - 9n^2 + 93n - 77}{81}$$

whenever $n \geqslant 2$.

In particular, $\mathscr{A}_n$ has asymptotically $25 \times 2^{2n-3}/27$ states and an average of $3n/4 - 29/30$ transitions per state; this largely exceeds the growth rate of the monoid B_n^+, which is only $3.233636\ldots$ [10]. Like the minimal automaton of $\mathsf{LexMin}_n(B_n^+)$, the size of $\mathscr{A}_n$ is exponential in n; the exponent is larger, being $2^2 = 4$ here instead of $\Phi^2 \approx 2.618$.

2 Characterising words in alternating normal form

In this section, we briefly present key results paving the way for Theorem 10, which is an "automata-flavoured" characterisation of words $w \in \mathbb{A}_n^*$ in alternating normal form. These results are based on the *ad hoc* notions of *chain*, or *chain containment* and of *rigid chain containment*, the latter two being braid invariants. Their full proofs are omitted in this paper.

▶ **Definition 4.** *Let $u \geqslant v$ be two integers. The braid word $\sigma_{u\to v} = \sigma_u\sigma_{u-1}\cdots\sigma_v$, which is the only factorisation of the braid it represents, is called a* (u, v)-chain. *Then, we say that a word $w \in \mathbb{A}_n^*$* contains a (u, v)-chain *if $\sigma_{u\to v}$ is a subword of w, i.e., if w admits a factorisation of the form $w = w^{(u)}\sigma_u w^{(u-1)}\sigma_{u-1}\cdots w^{(v)}\sigma_v w^{(v-1)}$. If, furthermore, no factor $w^{(i)}$ contains any occurrence of the letters σ_i or σ_{i+1}, we say that w* rigidly contains *a (u, v)-chain.*

Although the braid word $\sigma_u\sigma_{u+1}\cdots\sigma_v$ is not a chain when $u < v$, it will also be denoted by $\sigma_{u\to v}$; considering such words may be useful since ϕ_n exchanges $\sigma_{u\to v}$ and $\sigma_{(n-u)\to(n-v)}$.

▶ **Lemma 5.** *Let u and v be two integers such that $u \geqslant v$, and let w and w' be two words representing the same braid $\beta \in B_n^+$. If w contains a (u, v)-chain, so does w'. Similarly, if w rigidly contains a (u, v)-chain, so does w'; in that case, we have $\beta \geqslant_{\mathsf{R}} \sigma_{u\to v}$.*

Proof idea. It suffices to treat the case where w and w' are related by a single commutation or braid relation. Then, if w rigidly contains a (u, v)-chain, an induction on i shows that $\beta \geqslant_{\mathsf{R}} \sigma_{i\to v}$ whenever $u \geqslant i \geqslant v$. ◀

This generalises the property that containing a letter σ_u, i.e., a (u, u)-chain, is a braid invariant.

▶ **Lemma 6.** *Let $v \leqslant n-1$ be an integer. A braid $\beta \in B_n^+$ contains an $(n-1, v)$-chain if and only if its n-body contains a letter σ_v.*

Proof idea. Let $\mathsf{h}_n(\beta)$ and $\mathsf{b}_n(\beta)$ be represented by two braid words $w \in \mathbb{A}_{n-1}^*$ and $w' \in \mathbb{A}_n^*$. If their concatenation ww' contains an $(n-1, v)$-chain, the leftmost letter of that chain must already belong to w', and so must its rightmost letter σ_v. Conversely, if $\mathsf{b}_n(\beta)$ contains a letter σ_v, so does w', and each occurrence of a letter $\sigma_i \neq \sigma_{n-1}$ in w' must be preceded by an occurrence of the letter σ_{i+1}, thereby proving that w' contains an $(n-1, v)$-chain. ◀

To obtain the desired characterisation, we introduce the notions of *left* and *right sets* of a braid.

▶ **Definition 7.** *The* left set *of a braid β is defined as the set $\mathsf{L}(\beta) = \{i : \sigma_i \leqslant_{\mathsf{L}} \beta\}$, and the* right set *of β is defined as the set $\mathsf{R}(\beta) = \{i : \beta \geqslant_{\mathsf{R}} \sigma_i\}$.*

▶ **Lemma 8.** *Let $v \leqslant n-1$ be an integer and $\beta \in B_n^+$ be a braid such that $\mathsf{L}(\beta) = \{n-1\}$. Either β is a chain or there exists an integer $v \leqslant n-1$ such that $\sigma_{(n-1)\to v}\sigma_v$ is a prefix of each word representing β.*

Proof idea. Assuming that β is not a chain, let $\sigma_{(n-1)\to v}\sigma_u$ be a left divisor of β in which v is chosen minimal. If $u \geqslant v+1$, then $u-1 \in \mathsf{L}(\beta)$, which is impossible; Lemma 6 proves that $u \geqslant v-1$, and the minimality of v forbids the case $u = v-1$. ◀

The interest of these notions arises from the following result, which relates each braid $\beta \in B_n^+$ with braids $\beta\sigma_{n\to v} \in B_{n+1}^+$:

▶ **Proposition 9.** *Let $v \leqslant n$ be an integer. A braid $\beta \in B_n^+$ contains an $(n-1, v-1)$-chain if and only if $\mathsf{L}(\beta) = \mathsf{L}(\beta\sigma_{n\to v})$.*

Proof idea. If β contains no $(n-1, v-1)$-chain, $\mathsf{b}_n(\beta)$ contains no letter σ_{v-1}: it is the commutative product of two braids γ and γ', with generators in $\{\sigma_1, \ldots, \sigma_{v-2}\}$ and $\{\sigma_v, \ldots, \sigma_{n-1}\}$, respectively. But then, $\beta\sigma_{n\to v} = \mathsf{h}_n(\beta)\sigma_{n\to v}\gamma\phi_\uparrow(\gamma')$, where the morphism $\phi_\uparrow$ maps each generator σ_i such that $i \geqslant v$ to the generator σ_{i+1}. Since $\mathsf{h}_n(\beta)$ belongs to B_{n-1}^+, it commutes with σ_n, which ends up left-dividing $\beta\sigma_{n\to v}$, but not β.

Conversely, if β contains an $(n-1, v-1)$-chain, every word w representing $\beta\sigma_{n\to v}$ both contains an $(n-1, v-1)$-chain and rigidly contains an (n, v)-chain. However, we can prove that each occurrence of a letter σ_i of the former chain lies to the left of the occurrence of the letter σ_{i+1} of the latter chain. Thus, $w_{\geqslant 1}$ rigidly contains an (n, v)-chain, and Lemma 5 proves that the braid β' represented by $w_{\geqslant 1}$ is right-divided by $\sigma_{n\to v}$. This means that $\beta = w_0(\beta'\sigma_{n\to v}^{-1})$ is left-divided by w_0, this reasoning being valid for each letter $w_0 \in \mathsf{L}(\beta\sigma_{n\to v})$. ◀

From these results, we can derive the following characterisation of the alternating normal form $\mathsf{Alt}_n(B_n^+)$.

▶ **Theorem 10.** *A word $w \in \mathbb{A}_n^*$ belongs to $\mathsf{Alt}_n(B_n^+)$ if and only if $n = 2$ or $n \geqslant 3$ and w has a (necessarily unique) factorisation $w = w^{(0)}\phi_n(w^{(1)})\phi_n^2(w^{(2)})\cdots\phi_n^k(w^{(k)})$ such that:*

1. *each of the words $w^{(0)}, w^{(1)}, \ldots, w^{(k)}$ belongs to $\mathsf{Alt}_{n-1}(B_{n-1}^+)$;*
2. *each of the words $\phi_{n-1}(w^{(1)}), \phi_{n-1}(w^{(2)}), \ldots, \phi_{n-1}(w^{(k)})$ belongs to $\mathsf{Alt}_{n-1}(B_{n-1}^+)$ and starts with the letter σ_{n-2};*
3. *for all $i \geqslant 1$ and $v \geqslant 1$, if $\sigma_{(n-1)\to v}$ is a prefix of $\phi_n(w^{(i+1)})\phi_n^2(w^{(i+2)})\cdots\phi_n^{k-i}(w^{(k)})$, the word $w^{(i)}$ contains an $(n-2, v-1)$-chain.*

Proof idea. Given a braid $\beta \in B_n^+$ such that $w = \mathsf{Alt}_n(\beta)$, the factors $w^{(i)}$ are the n-heads of the braids $\beta^{(0)}, \beta^{(1)}, \ldots$ given by $\beta^{(0)} = \beta$ and $\beta^{(i+1)} = \phi_n(\mathsf{b}_n(\beta^{(i)}))$. Statement **1** is true by induction on n. Then, for all $i \geqslant 1$, we have $\mathsf{L}(\mathsf{h}_n(\beta^{(i)})) \subseteq \mathsf{L}(\beta^{(i)}) = \{1\}$; thus, $w^{(i)}$ starts with the letter σ_1, and ϕ_n exchanges the alternating normal forms of the braids $\mathsf{h}_n(\beta^{(i)})$ and $\phi_n(\mathsf{h}_n(\beta^{(i)}))$. In other words, $\phi_{n-1}(w^{(i)})$ starts with the letter σ_{n-2} and coincides with the word $\mathsf{Alt}_{n-1}(\phi_{n-1}(\mathsf{h}_n(\beta^{(i)})))$, which proves statement **2**. Finally, when $i \geqslant 1$, an induction on k proves that $w^{(i)}\phi_n(w^{(i+1)})\cdots\phi_n^{i+k}(w^{(i+k)}) = \mathsf{Alt}_n(\beta^{(i)})$, and since $\mathsf{L}(\beta^{(i)}) = \{1\}$, it must coincide with its (non-empty) subset $\mathsf{L}(w^{(i)})$, thereby making statement **3** a consequence of Lemma 6 and Proposition 9.

Conversely, given a factorisation $w^{(0)}\phi_n(w^{(1)})\cdots\phi_n^k(w^{(k)})$ of a word $w \in \mathbb{A}_n^*$ that makes statements **1** to **3** valid, we prove by induction on n and k that each word $w^{(i)}$ is the alternating normal form of the n-head of the braid $\beta^{(i)}$, where $\beta^{(0)}$ is the word represented by w and $\beta^{(i+1)} = \phi_n(\mathsf{b}_n(\beta^{(i)}))$. The induction hypothesis proves that both words $w' = w^{(0)}$ and $w'' = w^{(1)}\phi_n(w^{(2)})\cdots\phi_n^{k-1}(w^{(k)})$ are the alternating normal forms of braids $\beta' \in B_{n-1}^+$ and $\beta'' \in B_n^+$, and it remains to prove that $\mathsf{L}(\beta'') \subseteq \{1\}$, which we do by induction on k.

This is vacuously true when $k = 0$ and, when $k \geqslant 1$, the induction hypothesis ensures that $w^{(1)}$ and $w^{(2)}\phi_n(w^{(3)})\cdots\phi_n^{k-2}(w^{(k)})$ are the alternating forms of $\mathsf{h}_n(\beta'')$ and $\phi_n(\mathsf{b}_n(\beta''))$. Since $\mathsf{L}(\mathsf{b}_n(\beta'')) \subseteq \{n-1\}$, Lemma 8 proves that $\mathsf{s}_n(\mathsf{b}_n(\beta''))$ coincides with a chain $\sigma_{n-1\to v}$ that is a prefix of each word representing $\mathsf{b}_n(\beta'')$, including $\phi_n(w^{(2)})\phi_n^2(w^{(3)})\cdots\phi_n^{k-1}(w^{(k)})$. Thus, statement **3** proves that $w^{(1)}$, or, equivalently, $\mathsf{h}_n(\beta'')$, contains an $(n-2, v-1)$-chain. Consequently, since $\mathsf{L}(\beta'') = \mathsf{L}(\mathsf{h}_n(\beta'')\mathsf{b}_n(\beta'')) = \mathsf{L}(\mathsf{h}_n(\beta''\mathsf{s}_n(\mathsf{b}_n(\beta'')))) = \mathsf{L}(\mathsf{h}_n(\beta''\sigma_{n-1\to v}))$, Proposition 9 proves that $\mathsf{L}(\beta'') = \mathsf{L}(\mathsf{h}_n(\beta''))$. But $\phi_{n-1}(w^{(1)})$ is the alternating normal form of a braid that must coincide with $\phi_{n-1}(\mathsf{h}_n(\beta''))$, and its first letter is σ_{n-2}, which means that the $(n-1)$-head of $\phi_{n-1}(\mathsf{h}_n(\beta''))$ is empty, i.e., that $\mathsf{L}(\phi_{n-1}(\mathsf{h}_n(\beta''))) \subseteq \{n-2\}$. It follows, as desired, that $\mathsf{L}(\mathsf{h}_n(\beta'')) \subseteq \{1\}$. ◀

3 Minimal automata

In this section, we explicitly build the minimal automaton of the language $\mathscr{L}_n = \mathsf{Alt}_n(B_n^+)$. In order to do so, a crucial step lies in building the minimal automaton of the language $\mathsf{Alt}_n(\mathsf{b}_n(B_n^+))$. Noting that $\mathsf{b}_n(B_n^+)$ coincides with the set of braids $\beta \in B_n^+$

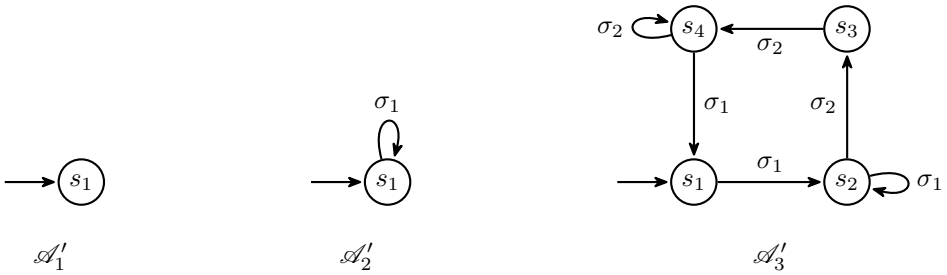

Figure 4 Minimal automata $\mathscr{A}'_1$ to $\mathscr{A}'_3$.

such that $\mathsf{L}(\beta) \subseteq \{\sigma_{n-1}\}$, it turns out that $\phi_n(\mathsf{Alt}_n(\beta)) = \mathsf{Alt}_n(\phi_n(\beta))$ whenever $\beta \in \mathsf{b}_n(B_n^+)$. Thus, we also look at the language $\mathscr{L}'_n = \mathsf{Alt}_n(\phi_n(\mathsf{b}_n(B_n^+)))$, which is connected to $\mathscr{L}_n$ by the relation $\mathscr{L}_n = \mathscr{L}_{n-1}\phi_n(\mathscr{L}'_n)$. In particular, $\mathscr{L}'_n$ is the language of alternating normal forms of braids $\beta \in B_n^+$ such that $\mathsf{L}(\beta) \subseteq \{\sigma_1\}$; viewing B_n^+ as a subset of B_{n+1}^+ proves that $\mathscr{L}'_n \subseteq \mathscr{L}'_{n+1}$, which is the reason why we chose to study $\mathscr{L}'_n$ and not its conjugate $\phi_n(\mathscr{L}'_n)$.

Both languages $\mathscr{L}_n$ and $\mathscr{L}'_n$ are prefix-closed: when they contain a word w, they also contain all its prefixes. Thus, we identify the minimal automaton of $\mathscr{L}_n$ with a tuple $\mathscr{A}_n = (V_n, \mathbb{A}_n, \delta_n, \imath_n)$, in which V_n denotes the set of states, all of which are accepting; $\mathbb{A}_n$ is the alphabet; $\delta_n \colon V_n \times \mathbb{A}_n \mapsto V_n$ denotes the transition function; and $\imath_n \in V_n$ denotes the initial state of the automaton. Similarly, we identify the minimal automaton of $\mathscr{L}'_n$ with a tuple $\mathscr{A}'_n = (V'_n, \mathbb{A}_n, \delta'_n, \imath'_n)$, and we shall first focus on constructing the automata $\mathscr{A}'_n$. However, in order to define this latter automaton, we will first define an auxiliary automaton $\mathscr{A}''_n = (V''_n, \mathbb{A}_n, \delta''_n, \imath''_n)$ that will recognise $\mathscr{L}'_n$ but not be minimal; the automaton $\mathscr{A}'_n$ will be built by minimising $\mathscr{A}''_n$.

The languages $\mathscr{L}'_1$ and $\mathscr{L}'_2$ consist of the empty word and of all words on the alphabet $\mathbb{A}_2 = \{\sigma_1\}$, respectively. Then, when $n \geqslant 3$, we focus on building the automata $\mathscr{A}_n$ and $\mathscr{A}'_n$ based on the automata $\mathscr{A}_{n-1}$ and $\mathscr{A}'_{n-1}$. Of course, we may use Theorem 10 directly to compute any automaton $\mathscr{A}_n$; for instance, $\mathscr{L}'_3$ consists of words of the form $(\sigma_1\sigma_1^*\sigma_2^2\sigma_2^*\sigma_1)^*$ and their prefixes, from which we deduce the automaton $\mathscr{A}'_3$ given in Figure 4.

Below, we proceed in four steps: first, we present a few preliminary results; second, we construct an automaton $\mathscr{A}''_n$ that recognises the language $\mathscr{L}'_n$; third, we construct the minimal automaton $\mathscr{A}'_n$ of $\mathscr{L}'_n$; fourth, we construct the minimal automaton $\mathscr{A}_n$ of $\mathscr{L}_n$ itself.

3.1 Preliminary results

When $n \geqslant 3$, and due to Theorem 10, a word $w \in \mathbb{A}_n^*$ belongs to $\mathscr{L}'_n$ if and only if w has a factorisation $w = w^{(0)}\phi_n(w^{(1)}) \cdots \phi_n^k(w^{(k)})$ such that:

2' each word $w^{(i)}$ belongs to $\mathscr{L}'_{n-1}$;

3' for all $i \geqslant 0$ and $v \geqslant 1$, if $\sigma_{(n-1)\to v}$ is a prefix of $\phi_n(w^{(i+1)})\phi_n^2(w^{(i+2)}) \cdots \phi_n^{i+k}(w^{(k)})$, the word $w^{(i)}$ contains a $(n-2, v-1)$-chain.

These are variants of criteria **2** and **3**. Criterion **3'** also requires, while reading a word $w \in \mathscr{L}'_{n-1}$, recalling the least integer v (if any) for which w contains an $(n-2, v)$-chain. This integer is denoted by $\mathsf{ch}_{n-1}(w)$; when w contains no $(n-2, v)$-chain at all, i.e., when $w \in \mathbb{A}_{n-2}^*$, we set $\mathsf{ch}_{n-1}(w) = n-1$. In particular,

i) if $k = 0$, we simply have $\mathsf{ch}_{n-1}(w) = n-1$;

ii) if $k = 1$, the integer $\mathsf{ch}_{n-1}(w)$ may vary between 2 and $n-2$;

iii) if $k \geqslant 2$, the word w contains the $(n-2, 1)$-chain $\sigma_{(n-2)\to 1}$, and $\mathsf{ch}_{n-1}(w) = 1$.

The following results allow determining $\mathsf{ch}_{n-1}(w)$ in case **ii)**.

▶ **Lemma 11.** *Let w be a word belonging to $\mathscr{L}_n$, and let σ_u be its rightmost letter, or $\sigma_u = \sigma_1$ if w is empty. For all $v \leqslant n-1$, the word $w\sigma_{u\to v}$ also belongs to $\mathscr{L}_n$; furthermore, if w belongs to $\mathscr{L}'_n$, so does $w\sigma_{u\to v}$. Finally, if $n \geqslant 2$ and w is a non-empty word in $\mathscr{L}'_n$, there exist at least two integers x and y such that $w\sigma_x$ and $w\sigma_y$ belong to $\mathscr{L}'_{n+1}$.*

Proof. We prove both statements of Lemma 11 separately. First, let ℓ be the length of w, and let $w' = w\sigma_{u\to v}$. The first statement being immediate when $\ell = 0$, we assume that $\ell \geqslant 1$. Since the braid word $\sigma_u\sigma_{u\to v}$ is the only representative of its braid, we have $\mathsf{s}_n(\sigma_u\sigma_{u\to v}) = \sigma_u$. A backward induction on i proves then that $\mathsf{s}_n(w_{\geqslant i}) = \mathsf{s}_n(w'_{\geqslant i})$ for all $i \leqslant \ell - 1$. It follows from the remark **3)** of page 4 that $w\sigma_{u\to v} \in \mathscr{L}_n$, and that $\mathsf{L}(w\sigma_{u\to v}) = \mathsf{L}(w)$, thereby proving that $w\sigma_{u\to v} \in \mathscr{L}'_n$ if $w \in \mathscr{L}'_n$.

We prove the last statement by induction on n. If $n = 2$, the word w is of the form $w = \sigma_1^\ell$, and $w\sigma_1$ and $w\sigma_2$ belong to $\mathscr{L}'_3$. If $n \geqslant 3$, let $w = w^{(0)}\phi_n(w^{(1)})\cdots\phi_n^k(w^{(k)})$ be the factorisation of w given in Theorem 10. If $k = 0$, the word w belongs to $\mathscr{L}'_{n-1}$, and the induction hypothesis also proves that there exist two integers x and y for which $w\sigma_x$ and $w\sigma_y$ belong to $\mathscr{L}'_n$. Otherwise, $k \geqslant 1$, and w both ends with some letter σ_u and contains the letter σ_{n-1}, which proves that both $w\sigma_u$ and $w\sigma_n$ belong to $\mathscr{L}'_{n+1}$. ◀

▶ **Lemma 12.** *Let w be a word belonging to both $\mathscr{L}'_{n-1}$ and $\mathscr{L}'_{n-2}\phi_n(\mathscr{L}'_{n-2})$, and let $w^{(0)}$ be its longest prefix belonging to $\mathbb{A}^*_{n-2}$:*

- *if w has a factorisation of the form $w = w^{(0)}\sigma_{(n-2)\to v}$, then $\mathsf{ch}_{n-1}(w) = v$;*
- *otherwise, let v be the least integer such that $w\sigma_v \in \mathscr{L}'_{n-1}$: we have $\mathsf{ch}_{n-1}(w) = v+1$.*

Proof. The first part of Lemma 12 being immediate, we focus on the second part. In that case, let $w^{(1)}$ be the suffix of w such that $w = w^{(0)}w^{(1)}$, and let $\beta \in B^+_{n-1}$ be the braid represented by w. By construction, $w^{(1)}$ belongs to $\phi_{n-1}(\mathbb{A}^*_{n-2})$, and $w^{(1)} = \mathsf{Alt}_{n-1}(\mathsf{b}_{n-1}(\beta))$. Furthermore, by Lemma 6, $\mathsf{ch}_{n-1}(w)$ is simply the least letter of $w^{(1)}$, say, y.

In particular, when $z \leqslant y-2$, the word $w\sigma_z$ contains no $(n-2,z)$-chain, and Lemma 6 prevents it from belonging to $\mathscr{L}'_{n-1}$. Conversely, the word $\phi_{n-1}(w^{(1)}\sigma_{y-1}) = \phi_{n-1}(w^{(1)})\sigma_{n-y}$ satisfies both criteria **2'** and **3'** that mark it as a member of $\mathscr{L}'_{n+1-y}$, and thus of $\mathscr{L}'_{n-1}$. Moreover, $w^{(1)}$ is not a chain, so that the maximal chains that are prefixes of $w^{(1)}$ and of $w^{(1)}\sigma_{y-1}$ coincide with each other. Consequently, $w\sigma_{y-1}$ itself belongs to $\mathscr{L}'_{n-1}$. ◀

As a consequence of Lemma 12, for each word w in $\mathscr{L}'_{n-1}$, the integer $\mathsf{ch}_{n-1}(w)$ depends only on whether $\sigma_{n-2}\sigma_1$ is a subword of w and, if not, on the residual of w (i.e., on the set $\{x \in \mathbb{A}^*_{n-1} : wx \in \mathscr{L}'_{n-1}\}$). For each automaton $\mathscr{A} = (V, \mathbb{A}_{n-1}, \delta, \imath)$ recognising the language $\mathscr{L}'_{n-1}$, the residual of a word $w \in \mathscr{L}'_{n-1}$ depends only on the state $s = \delta(\imath, w)$ of $\mathscr{A}$ to which w is mapped. Thus, below, for each state s of $\mathscr{A}'_{n-1}$, we simply note $\mathsf{ch}_{n-1}(s)$ the common value of the integers $\mathsf{ch}_{n-1}(w)$ when $\delta'(\imath'_{n-1}, w) = s$ and $\sigma_{n-2}\sigma_1$ is not a subword of w; if no such word w exists, we set $\mathsf{ch}_{n-1}(s) = 1$.

3.2 Construction and correctness of the automaton $\mathscr{A}''_n$

Here, we give a construction of an automaton $\mathscr{A}''_n$ that recognises the language $\mathscr{L}'_n$. This automaton *looks like* its minimal equivalent $\mathscr{A}'_n$ represented in Figure 5 when $n = 4$. The semantics of its states is given in the beginning of the proof of Proposition 13.

▶ **Proposition 13.** *Given an integer $n \geqslant 4$, let $\mathscr{A}'_{n-1} = (V'_{n-1}, \mathbb{A}_{n-1}, \delta'_{n-1}, \imath'_{n-1})$ be the minimal automaton recognising $\mathscr{L}'_{n-1}$. The language $\mathscr{L}'_n$ is recognised by the (deterministic, non-minimal) automaton $\mathscr{A}''_n = (V''_n, \mathbb{A}_n, \delta''_n, \imath''_n)$ defined as follows. The state set of $\mathscr{A}''_n$ is given by $V''_n = ((V'_{n-1}\times\{\top,\bot\}) \cup P''_n)\times\{\mathsf{Id}_n, \phi_n\}$, where we set $P''_n = \{p^j_i : 2 \leqslant j \leqslant i \leqslant n-1\}$; its initial state is $\imath''_n = (\imath'_{n-1}, \bot, \mathsf{Id}_n)$; and its transition function δ''_n is given by:*

a. $\delta''_n((s, \mathsf{f}, \mathsf{Id}_n), \sigma_i) = (\delta'_{n-1}(s, \sigma_i), \mathsf{f}, \mathsf{Id}_n)$ *when* $\mathsf{ch}_{n-1}(s) \neq 2$ *or* $i \neq 1$*;*
b. $\delta''_n((s, \mathsf{f}, \mathsf{Id}_n), \sigma_1) = (\delta'_{n-1}(s, \sigma_1), \top, \mathsf{Id}_n)$ *when* $\mathsf{ch}_{n-1}(s) = 2$*;*
c. $\delta''_n((p_i^j, \mathsf{Id}_n), \sigma_{n+1-i}) = (p_{i-1}^j, \mathsf{Id}_n)$ *when* $p_{i-1}^j \in P_n$*;*
d. $\delta''_n((p_i^j, \mathsf{Id}_n), \sigma_{n-i}) = (\delta'_{n-1}(\imath'_{n-1}, \sigma_{1\to(n-i)}\sigma_{n-i}), \bot, \mathsf{Id}_n)$ *when* $p_i^j \in P_n$*;*
e. $\delta''_n((s, \bot, \mathsf{Id}_n), \sigma_{n-1}) = (p_{n-1}^j, \phi_n)$ *when* $3 \leqslant j \leqslant n-1$ *and* $\mathsf{ch}_{n-1}(s) = j-1$*;*
f. $\delta''_n((s, \top, \mathsf{Id}_n), \sigma_{n-1}) = (p_{n-1}^2, \phi_n)$*;*
g. $\delta''_n((s, \mathsf{f}, \phi_n), \sigma_i) = (s', \mathsf{f}', \phi_n^{k+1})$ *when* $\delta''_n((s, \mathsf{f}, \mathsf{Id}_n), \sigma_{n-i}) = (s', \mathsf{f}', \phi_n^k)$*;*
h. $\delta''_n((s, \mathsf{f}, \phi_n^k), \sigma_i)$ *is not defined in all other cases.*

Proof. Let $w = w^{(0)}\phi_n(w^{(1)}) \cdots \phi_n^k(w^{(k)})$ be the factorisation a word $w \in \mathscr{L}'_n$ given in Theorem 10. Here is the intended semantics of the state $\delta''_n(\imath''_n, w)$ to which w is sent. The state $\delta''_n(\imath''_n, w)$ shall be of the form $(s, \mathsf{f}, \phi_n^k)$ or (p_i^j, ϕ_n^k), thereby indicating whether the factorisation of w contains an odd or an even number of factors. In general, i.e., when $k = 0$ or when $\phi_n(w^{(k)})$ is *not* a chain, we shall set $s = (\delta'_{n-1}(\imath'_{n-1}, w^{(k)}), \mathsf{f}, \phi_n^k)$, where f is a boolean flag set to $\mathsf{f} = \top$ (i.e., $\mathsf{f} = \textbf{true}$) if $w^{(k)}$ contains an $(n-2, 1)$-chain, and $\mathsf{f} = \bot$ (i.e., $\mathsf{f} = \textbf{false}$) otherwise. However, if $k \geqslant 1$ and $\phi_n(w^{(k)})$ is an $(n-1, v)$ chain, let $x = \mathsf{ch}_{n-1}(w^{(k-1)})$: the state $\delta''_n(\imath''_n, w)$ shall be the pair (p_v^{x+1}, ϕ_n^k), thereby indicating that $w^{(k)}$ is an $(n-1, v)$-chain that can be extended to form an $(n-1, x+1)$-chain, but not more.

We can now prove by induction on ℓ that, for each word $w \in \mathscr{L}'_n$ of length ℓ, the state $\delta''_n(\imath''_n, w)$ has the intended semantics, thereby demonstrating that $\mathscr{A}''_n$ recognises the language $\mathscr{L}'_n$. The base case $\ell = 0$ follows from our choice for $\imath''_n$; now, assuming that $\ell \geqslant 1$ and that $\ell - 1$ satisfies the inductive property, we wish to prove that ℓ also satisfies this property, by considering the cases **a** and **h** separately.

Case **a** is the general case obtained when facing a state $\delta''_n(\imath''_n, w)$ for which k is even: reading a letter σ_i will just let the word $w^{(k)}$ grow inside the language $\mathscr{L}'_{n-1}$, without changing the flag f or parity of k. By contrast, case **b** happens when f changes from $\bot$ to $\top$: indeed, the word $w^{(k)}$ already contained an $(n-2, 2)$-chain, and adding a letter σ_1 yields an $(n-2, 1)$-chain. Thus, the transition labelled σ_1 targets the state $(\delta'_{n-1}(s, \sigma_1), \top, \mathsf{Id}_n)$ instead of the state $(\delta'_{n-1}(s, \sigma_1), \bot, \mathsf{Id}_n)$ that might otherwise have been expected.

Then, cases **c** and **d** focus on transitions leaving a state (p_i^j, Id_n): either we read the letter σ_{n+1-i} (in case **c**) and we just transformed the $(n-2, i)$-chain $\phi_n(w^{(k)})$ into an $(n-2, i-1)$-chain $\phi_n(w^{(k)})\sigma_{i-1}$, or we read the letter σ_{n-i} (in case **c**), in which case $w^{(k)}\sigma_{n-i}$ stops being a chain, and we should just remember the same amount of information as if we were reading the word $w^{(k)}\sigma_{n-i}$ instead of $w^{(0)}\phi_n(w^{(1)})w^{(2)} \cdots w^{(k)}\sigma_{n-i}$.

Cases **e** and **f** focus on transitions labelled σ_{n-1}: they target the state $(p_{n-1}^{x+1}s, \phi_n)$, where x is the least integer such that $w^{(k)}$ contained an $(n-2, x)$-chain; that integer x is given by Lemma 12 in case **e**, where $w^{(k)}$ contains no $(n-2, 1)$-chain; it is just 1 in case **f**.

Finally, case **g** replicates all cases **a** to **f**, but when k is odd instead of even. And case **h** consists in observing that cases **a** to **g** already cover all possible transitions of the automaton $\mathscr{A}''_n$. ◀

3.3 Construction, correctness and minimality of the automaton $\mathscr{A}'_n$

When $n \geqslant 4$, the automaton $\mathscr{A}''_n$ is not minimal. A first reason is that some states have the same residuals, and should thus be merged; we recall that the residual of a state $s \in V''_n$ is the language $\mathscr{L}'_n(s)$ of those words $w \in \mathbb{A}_n^*$ for which $\delta''_n(s, w)$ exists and is accepting. A second reason is that, when $n \geqslant 5$, some states of $\mathscr{A}''_n$ are not even accessible. Thus, we shall transform $\mathscr{A}''_n$ into the minimal automaton $\mathscr{A}'_n$ of $\mathscr{L}'_n$.

▶ **Definition 14.** *Given an integer $n \geqslant 4$, let $\mathscr{A}''_n = (V''_n, \mathbb{A}_n, \delta''_n, \imath'_n)$ be the automaton built in Proposition 13. The automaton $\mathscr{A}'_n = (V'_n, \mathbb{A}_n, \delta'_n, \imath'_n)$ is defined from $\mathscr{A}''_n$ by*

- *merging the states $(\imath'_{n-1}, \bot, \phi^k_n)$ and $(p^{n-1}_{n-1}, \phi^k_n)$ for each $\phi^k_n \in \{\mathsf{Id}_n, \phi_n\}$; this amounts to deleting the state $(p^{n-1}_{n-1}, \phi^k_n)$ and redirecting toward $(\imath'_{n-1}, \bot, \phi^k_n)$ those transitions of $\mathscr{A}''_n$ that were targeted toward $(p^{n-1}_{n-1}, \phi^k_n)$;*
- *deleting states of the form $((p^j_i, \mathsf{Id}_{n-1}), \bot, \phi^k_n)$, where $p^j_i \in P'_{n-1}$ and $\phi^k_n \in \{\mathsf{Id}_n, \phi_n\}$.*

In particular, the actual state set of $\mathscr{A}'_n$ is $V'_n = ((\overline{V}'_{n-1} \times \{\bot\}) \cup (V'_{n-1} \times \{\top\}) \cup P'_n) \times \{\mathsf{Id}_n, \phi_n\}$, where $P'_n = P''_n \setminus \{p^{n-1}_{n-1}\}$ and $\overline{V}'_{n-1} = V'_{n-1} \setminus (P'_{n-1} \times \{\mathsf{Id}_{n-1}\})$.

The reason why we shall merge the states $\imath''_n = (\imath'_{n-1}, \bot, \mathsf{Id}_n)$ and $(p^{n-1}_{n-1}, \mathsf{Id}_n)$ is that both states have only one outgoing transition, labelled by σ_1, and whose target state is $\delta''_n(\imath''_n, \sigma_1)$. Similarly, we shall merge the states $(\imath'_{n-1}, \bot, \phi_n)$ and (p^{n-1}_{n-1}, ϕ_n). Below, we will consider $(p^{n-1}_{n-1}, \phi^k_n)$ as an alias for $(\imath'_{n-1}, \bot, \phi^k_n)$.

Then, we shall also delete states of the form $((p^j_i, \mathsf{Id}_{n-1}), \bot, \phi^k_n)$ because they are not accessible. Indeed, by construction of δ''_{n-1}, every path in $\mathscr{A}''_{n-1}$ (or in $\mathscr{A}'_{n-1}$) and ending in such a state $(p^j_i, \mathsf{Id}_{n-1})$ must have previously visited a state of the form (s, ϕ_{n-1}). Thus, every word w for which $\delta'_{n-1}(\imath'_{n-1}, w) = (p^j_i, \mathsf{Id}_{n-1})$ already contains an $(n-2, 1)$-chain, thereby proving that $\delta''_n(\imath'_n, w) = ((p^j_i, \mathsf{Id}_{n-1}), \top, \mathsf{Id}_n)$. It follows, as announced, that $((p^j_i, \mathsf{Id}_{n-1}), \bot, \mathsf{Id}_n)$ is inaccessible, and we prove similarly that $((p^j_i, \mathsf{Id}_{n-1}), \bot, \phi_n)$ is also inaccessible.

An example of this construction, when $n = 4$, is given in Figure 5, where we start from the 4-state automaton $\mathscr{A}'_3$, whose states are denoted by s_1 to s_4, and obtain the 20-state automaton $\mathscr{A}'_4$. We wish we had represented the automaton $\mathscr{A}'_5$, thereby showing why replacing V'_{n-1} by $\overline{V}'_{n-1}$ is important, but $\mathscr{A}'_5$ contains 86 states, which is difficult to read. For the sake of readability, each state $(s_i, \mathsf{f}, \phi^k_n)$ is denoted by s^{f}_i, with a bar when $\phi^k_n = \phi_n$, i.e., when k is odd. We added the dangling states (p^j_i, ϕ^k_n), which are also denoted by p^j_i, with a bar when $\phi^k_n = \phi_n$.

While the above paragraphs prove that the automaton $\mathscr{A}'_n$ recognises the same language as $\mathscr{A}''_n$, i.e., the language $\mathscr{L}'_n$, we shall now prove that its states are accessible and have pairwise distinct residuals. Once again, we proceed by induction and, since this result is clear when $n \leqslant 3$, we assume that $n \geqslant 4$ and that $\mathscr{A}'_{n-1}$ is already known to be minimal.

In Lemma 15, we prove not only that each state of $\mathscr{A}'_n$ is accessible, but also most states can be reached via a word that does not contain any $(n-1, 1)$-chain. This will be crucial toward proving, in Lemma 16, that for any two distinct states s and s', there exists a word that can be read from s but not from s', or from s' but not from s.

▶ **Lemma 15.** *The automaton $\mathscr{A}'_n$ is strongly connected. Furthermore, for each state s in $\overline{V}'_n$, i.e., each state s distinct from the states (p^j_i, Id_n) for which $p^j_i \in P'_n$, there exists a word w that does not contain any $(n-1, 1)$-chain and such that $\delta'_n(\imath'_n, w) = s$.*

Proof. The result being visibly correct when $n \leqslant 3$, let us assume that $n \geqslant 4$. Given a state $s \in V'_{n-1}$, the induction hypothesis ensures that there exists a word $w \in \mathscr{L}'_{n-1}$ without $(n-1, 1)$-chain for which $\delta'_{n-1}(\imath'_{n-1}, w) = s$. Since $\mathscr{A}'_{n-1}$ is strongly connected, it also tells us that there exists a non-empty word $w' \in \mathscr{L}'_{n-1}$ for which $\delta'_{n-1}(\imath'_{n-1}, \sigma_1 w') = \imath'_{n-1}$; it follows that $\delta'_n(\imath'_n, w'') = (\imath'_{n-1}, \top, \mathsf{Id}_n)$, where $w'' = \sigma_{1\to(n-2)}\sigma_{(n-2)\to 1}w'$.

Finally, for each state $p^j_i \in P'_n$, the word $w^j_i = \sigma_{1\to(n-2)}\sigma_{(n-2)\to(j-1)}\sigma_{(n-1)\to i}$ obeys the relation $\delta'_n(\imath'_n, w^j_i) = (p^j_i, \phi_n)$; identifying the states $(\imath'_{n-1}, \bot, \phi_n)$ and p^{n-1}_{n-1} also makes this construction valid when $p^j_i = p^{n-1}_{n-1}$. We complete the proof by observing that:

- $(s, \bot, \mathsf{Id}_n) = \delta'_n(\imath'_n, w)$;
- $(s, \top, \mathsf{Id}_n) = \delta'_n(\imath'_n, w''w)$;
- $(p^j_i, \phi_n) = \delta'_n(\imath'_n, w^j_i)$, including when $p^j_i = p^{n-1}_{n-1}$, i.e., when $(p^j_i, \phi_n) = (\imath'_{n-1}, \bot, \phi_n)$;

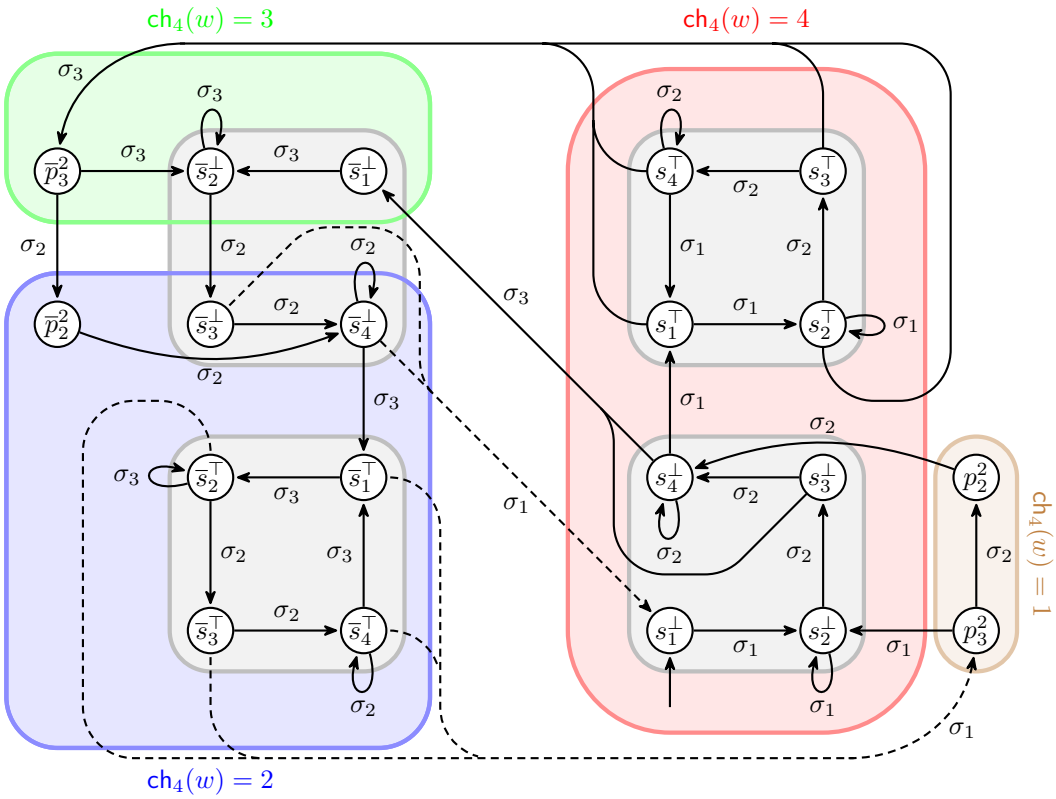

Figure 5 Automaton $\mathscr{A}'_4$, which contains four copies of $\mathscr{A}'_3$ (circled in gray, two of which miss an edge) and a few dangling states p_i^j. Erasing dashed edges gives us an automaton that recognises the language $\mathscr{L}'_4 \cap (\mathscr{L}'_3\phi_4(\mathscr{L}'_3))$, in which p_2^2 and p_3^2 are no longer accessible, and whose states have been split into four classes, coloured in brown, blue, green and red: a state s lies in the blue (resp., green, red) class when $\mathsf{ch}_4(w) = 2$ (resp., 3, 4) for all the words $w \in \mathscr{L}'_3\phi_4(\mathscr{L}'_3)$ such that $\delta'_4(\imath'_4, w) = s$. Once a dashed edge has been taken, we just have $\mathsf{ch}_4(w) = 1$. Similarly, we have $\mathsf{ch}_3(w) = 1$ (resp., 2, 3) for all the words $w \in \mathscr{L}'_3$ such that $\delta'_4(\imath'_4, w) = s$ when $s \in \{s_1^\top, s_2^\top, s_3^\top, s_4^\top\}$ (resp., $\{s_3^\perp, s_4^\perp\}$, $\{s_1^\perp, s_2^\perp\}$).

- $(s, \perp, \phi_n) = \delta'_n((\imath'_{n-1}, \perp, \phi_n), \phi_n(w))$;
- $(s, \top, \phi_n) = \delta'_n((\imath'_{n-1}, \perp, \phi_n), \phi_n(w''w))$;
- $(p_i^j, \mathsf{Id}_n) = \delta'_n((\imath'_{n-1}, \perp, \phi_n), \phi_n(w_i^j))$, including when $(p_i^j, \mathsf{Id}_n) = (\imath'_{n-1}, \perp, \mathsf{Id}_n) = \imath'_n$. ◀

▶ **Lemma 16.** *The states of $\mathscr{A}'_n$ have pairwise distinct residuals.*

Proof. Let s and s' be distinct states of $\mathscr{A}'_n$, and let w and w' be non-empty words in $\mathscr{L}'_n$ such that $\delta'_n(\imath'_n, w) = s$ and $\delta'_n(\imath'_n, w') = s'$. In addition, let $w^{(0)}\phi_n(w^{(1)})\cdots\phi_n^k(w^{(k)})$ and $w'^{(0)}\phi_n(w'^{(1)})\cdots\phi_n^\ell(w'^{(\ell)})$ be their factorisations given by Theorem 10; we assume that w and w' were chosen so that k and ℓ are minimal.

We first prove that the states of the form $(t, \mathsf{f}, \mathsf{Id}_n)$ or (p_i^j, Id_n) have pairwise distinct residuals. We call such states "Id_n-states", as opposed to "ϕ_n-states":

- If s is of the form (p_i^j, Id_n), including if $i = n-1$, the only letter in $\mathscr{L}'_n(s)$ is σ_{n-i}. On the contrary, $\mathscr{L}'_n(s')$ contains at least two letters, both if s' is of the form (p_i^j, Id_n) with $i \neq j$, since two such letters are σ_{n-i} or σ_{n+1-i}, or if s' is of the form $(t', \mathsf{f}', \mathsf{Id}_n)$, because of Lemma 11 (which we can use because we forced w and w' to be non-empty). Thus, each state (p_i^j, Id_n) has a residual distinct from all other Id_n-states.

- If s is of the form (p_i^j, Id_n), it is the only state such that $\delta'_n(s, \sigma_{(n+1-j)\to(n-i)}) = (p_i^i, \mathsf{Id}_n)$. Thus, the state $\delta'_n(s', \sigma_{(n+1-j)\to(n-i)})$ either fails to exist or does not coincide with (p_i^i, Id_n): in both cases, s and s' have distinct residuals.
- If both s and s' are of the form $(t, \mathsf{f}, \mathsf{Id}_n)$ and $(t', \mathsf{f}', \mathsf{Id}_n)$, either $t \neq t'$, in which case the induction hypothesis proves that $\mathscr{L}'_n(s) \cap \mathscr{L}'_{n-1} = \mathscr{L}'_{n-1}(t) \neq \mathscr{L}'_{n-1}(t') = \mathscr{L}'_n(s') \cap \mathscr{L}'_{n-1}$, or $\mathsf{f} \neq \mathsf{f}'$, in which case the chain $\sigma_{(n-1)\to 2}$ belongs to exactly one of the residuals $\mathscr{L}'_n(s)$ and $\mathscr{L}'_n(s')$.

Similarly, all ϕ_n-states have distinct residuals.

Finally, assume that s is an Id_n-state and that s' is a ϕ_n-state. Let σ_u be the last letter of w, with $u \leqslant n-2$, and let $z = \mathsf{ch}_{n-1}(s)$. We have $\delta'_n(s, \sigma_{u\to(n-1)}) = (p_{n-1}^{z+1}, \phi_n)$. On the contrary, either the state $s_1 = \delta'_n(s', \sigma_u)$ is a ϕ_n-state, in which case the state $s_2 = \delta'_n(s', \sigma_{u\to(n-2)})$ is also a ϕ_n-state and $\delta'_n(s', \sigma_{u\to(n-1)}) = \delta'_n(s_2, \sigma_{n-1})$ differs from (p_{n-1}^{z+1}, ϕ_n), or s_1 is an Id_n-state of the form $(p_{n-1}^i, \mathsf{Id}_n)$, in which case $\delta'_n(s', \sigma_{u\to(n-1)}) = \delta'_n(s_1, \sigma_{(u+1)\to(n-1)})$ is also an Id_n-state. ◀

As a consequence of Lemmas 15 and 16, we obtain the following result.

▶ **Proposition 17.** *The automaton $\mathscr{A}'_n$ is the minimal automaton of the language $\mathscr{L}'_n$.*

3.4 Construction, correctness and minimality of the automaton $\mathscr{A}_n$

We finally construct the automata $\mathscr{A}_n$ as follows.

▶ **Definition 18.** *First, $\mathscr{A}_1 = \mathscr{A}'_1$ is the automaton with one unique (necessarily initial) state and no transition, and $\mathscr{A}_2 = \mathscr{A}'_2$ is the automaton with one unique state and one loop labelled by σ_1. Then, when $n \geqslant 3$, the state set and initial state of $\mathscr{A}_n$ are defined by $V_n = V_{n-1} \cup V'_n$ and $\imath_n = \imath_{n-1}$, and the transition function δ_n is defined as follows:*

- $\delta_n(s, \sigma_u) = \delta_{n-1}(s, \sigma_u)$ *when* $s \in V_{n-1}$ *and* $u \leqslant n-2$;
- $\delta_n(s, \sigma_{n-1}) = \delta'_n(\imath'_n, \sigma_1)$ *when* $s \in V_{n-1}$;
- $\delta_n(s, \sigma_u) = \delta'_n(s, \sigma_{n-u})$ *when* $s \in V'_n$

The relation $\mathscr{L}_n = \mathscr{L}_{n-1}\phi_n(\mathscr{L}'_n)$ proves that this automaton recognises the set $\mathscr{L}_n$. Then, we prove by induction that each state s of $\mathscr{A}_n$ is accessible: when $s \in V_{n-1}$, this is just the induction hypothesis, and when $s \in V'_n$, Lemma 15 proves that s is accessible from $\delta'_n(\imath'_n, \sigma_1)$, which is itself accessible *via* the one-letter word σ_{n-1}.

Our last task consists in proving that any two states s and s' of $\mathscr{A}_n$ have pairwise distinct residuals. When s and s' belong to V_{n-1}, this is the induction hypothesis, and when s and s' belong to V'_n, this is the irreducibility of $\mathscr{A}'_n$. Finally, when $s \in V_{n-1}$ and $s' \in V'_n$, let w be a word such that $\delta_n(\imath_n, w) = s$, and let σ_u be its last letter, or $\sigma_u = \sigma_1$ in case w is empty. Then, let $w' = \sigma_{u\to 1}$. Lemma 11 proves that ww' belongs to $\mathscr{L}_{n-1}$, so that $\delta_n(s, w'\sigma_{n-1}) = \delta'_n(\imath'_n, \sigma_1)$; this is a state of $\mathscr{A}'_n$ that differs from all states (p_i^j, Id_n). By contrast, if they ever exist, the state $\delta_n(s', w') = \delta'_n(s', \phi_n(w'))$ is a ϕ_n-state, so that $\delta_n(s', w'\sigma_{n-1}) = \delta'_n(\delta'_n(s', \phi_n(w')), \sigma_1)$ is a state $(p_{n-1}^i, \mathsf{Id}_n)$ – which may be the state $(p_{n-1}^{n-1}, \mathsf{Id}_n) = \imath'_n$. From the above discussion results the following theorem.

▶ **Theorem 19.** *The automaton $\mathscr{A}_n$ is the minimal automaton of the language $\mathsf{Alt}_n(B_n^+)$.*

4 Size of the minimal automata

This final section is devoted to evaluate the size of the minimal automaton $\mathscr{A}_n$ of the language $\mathsf{Alt}_n(B_n^+)$, both in terms of states and of transitions.

▶ **Theorem 20.** *The automaton $\mathscr{A}_n$ has s_n states and t_n transitions, where $s_1 = 1$, $t_1 = 0$, and*

$$s_n = \frac{25 \times 2^{2n-3} - 9n^2 + 3n + 7}{27} \text{ and } t_n = \frac{(225n - 290)2^{2n-5} - 9n^3 - 9n^2 + 93n - 77}{81}$$

whenever $n \geqslant 2$.

Proof. Below, let s'_n and t'_n denote the number of states and of transitions of the automaton $\mathscr{A}'_n$. First, we have $s'_1 = s'_2 = 1$, $s'_3 = 4$ and $t'_1 = 0$, $t'_2 = 1$, $t'_3 = 6$. Then, when $n \geqslant 3$, note that $|P'_n| = |P'_{n-1}| + (n-2)$. It follows that

$$s'_n = |V'_n| = 2((|V'_{n-1}| - |P'_{n-1}|) + |V'_{n-1} + |P'_n|) = 4s'_{n-1} + 2(n-2),$$

and an immediate induction proves that $s'_n = (25 \times 2^{2n-5} - 6n + 4)/9$ for all $n \geqslant 3$.

Furthermore, each state (p_i^j, ϕ_n^k) is of out-degree 2 when $i > j$, and 1 when $i = j$; the latter cases occurs $n-3$ times, and thus, $2|P'_n| - (n-3)$ transitions leave a state (p_i^j, Id_n). In addition, those Id_n-states of $\mathscr{A}'_n$ from which one can read a letter σ_{n-1} are the states $(s, \mathsf{f}, \mathsf{Id}_n)$ for which $\mathsf{f} = \top$ or s is a ϕ_{n-1}-state of $\mathscr{A}'_{n-1}$; there are s'_{n-1} states in the first family, and $s'_{n-1}/2$ states in the second family but not in the first one. Consequently, in total, there are

$$(t'_{n-1} - (2|P'_{n-1}| - (n-4))) + t'_{n-1} + (2|P'_n| - (n-3)) + 3s'_{n-1}/2$$

transitions leaving Id_n-states, and $t'_n = 4t'_{n-1} + 3s'_{n-1} + 2(2n-5)$. Hence, another induction proves that $t'_n = ((25n-35)2^{2n-7} - 2n + 4)/3$ for all $n \geqslant 3$.

Finally, $s_1 = s_2 = 1$, $t_1 = 0$ and $t_2 = 1$, whereas $s_n = s_{n-1} + s'_n$ and $t_n = t_{n-1} + s_{n-1} + t'_n$ for all $n \geqslant 3$. Thus, an easy induction proves once again that $s_n = (25 \times 2^{2n-3} - 9n^2 + 3n + 7)/27$ and $t_n = ((225n - 290)2^{2n-5} - 9n^3 - 9n^2 + 93n - 77)/81$ whenever $n \geqslant 2$. ◀

5 Open problems and perspectives

The above study of the minimal automaton leaves wide open a few questions, which we intend to explore in follow-up work.

Linear-time recognition algorithm

Precomputing in time $\mathcal{O}(s_n + t_n) = \mathcal{O}(n2^n)$ the automaton $\mathscr{A}_n$ gives us an algorithm that will then detect in time $\mathcal{O}(\ell)$ whether an ℓ-letter word $w \in \mathbb{A}_n^*$ is in alternating normal form. However, when ℓ is small, this precomputation may seem prohibitively costly. Instead, our recursive description of the automata $\mathscr{A}_n$ and $\mathscr{A}'_n$ also provides us with a simple algorithm that will run in time $\mathcal{O}(n\ell)$. Indeed, we can simulate the execution of a path in $\mathscr{A}_n$ as follows:

1. a pointer indicates which is the largest letter σ_{k-1} we have read so far, which means that we are currently reading a word in the automaton $\mathscr{A}'_k$;
2. each state of $\mathscr{A}'_k$ can be represented by a list of the form $s = (p_u^v, \phi_i^?, \mathsf{f}_{i+1}, \phi_{i+1}^?, \ldots, \mathsf{f}_k, \phi_k^?)$, where $p_u^v \in P'_i$, each flag f_j is a boolean $\bot$ or $\top$, and each morphism $\phi_j^?$ is either Id_j or ϕ_j;
3. our recursive description of the transition function δ_n makes it easy to compute in time $\mathcal{O}(n)$ the list that represents the state $\delta_n(s, \sigma_a)$ for all letters $\sigma_a \in \mathbb{A}_n$, provided that this state is well-defined.

Automaticity

Another natural question concerns the automaticity of the alternating normal form, which can be summarised as follows. For each generator $\sigma_i \in \mathbb{A}_n$, we wish to recognise those pairs of words (w, w') representing braids β and β' such that $\beta = \sigma_i\beta'$ (this is *left* automaticity) or $\beta = \beta'\sigma_i$ (this is *right* automaticity). In practice, the words w and w' having distinct lengths, the pair (w, w') shall be represented as a word on the alphabet $(\mathbb{A}_n \cup \{\bullet\})^2$, where $\bullet$ is a padding symbol; *synchronous* automaticity requires that the only padding symbol should be the rightmost symbol of w', and *asynchronous* automaticity allows placing padding symbols wherever we want. It is shown in [5, Proposition 6.10] that the alternating normal form is not asynchronously left-automatic. It might still be right-automatic; this should be the subject of a subsequent article.

Rotating normal form

A close cousin to the alternating normal form is the rotating normal form, already mentioned in the introduction. This normal form is not defined on the standard braid monoid B_n^+ itself, but on the dual braid monoid B_n^{+*} positively generated by the generators $\sigma_{i,j} = (\sigma_{(i+1)\to j})^{-1}\sigma_{i\to j}$. This monoid enjoys properties similar to the properties **a)** to **e)** of page 2. However, there, the Garside element is a braid δ_n for which the inner automorphism $\varphi_n\colon \beta \mapsto \delta_n^{-1}\beta\delta_n$ is not an involution of B_n^{+*} when $n \geqslant 3$, but is of order n.

Local criteria similar to Theorem 10 were found in [11], which help characterising words in rotating normal form and proving that this normal form is regular. Nevertheless, the resulting automaton is not yet guaranteed to be minimal, being analogous to our automaton $\mathscr{A}_n''$ rather than to its minimal variant $\mathscr{A}_n'$. Thus, we intend to replicate our study of the alternating normal form to the rotating normal form, possibly studying its right automaticity as well.

Random generation

In [13], V. Gebhardt and J. González-Meneses focus on the problem of generating uniformly at random a braid $\beta \in B_n^+$ of length $\ell \geqslant 0$. By identifying each braid β with the word $\mathsf{MinLex}_n(\beta)$, they reduce this problem to that of generating a word of length ℓ in a regular language, whose minimal automaton they computed. A crucial step is then to compute the number of paths of length ℓ, in the automaton, that may leave a given state s. Doing so efficiently requires:

- identifying the set of *minimal forbidden patterns*, i.e., the minimal words (for the prefix ordering) that do not label a path leaving the state s;
- applying inclusion-exclusion formulas based on that set;
- using structural properties of that set to perform only polynomially many (and not exponentially many) calls to inclusion-exclusion formulas.

As a result, they obtain an algorithm that generates β in time $\mathcal{O}(n^4 \log(n)\ell^3 \log(\ell))$. It might be possible to adapt this approach to efficiently count paths leaving a state of $\mathscr{A}_n$, thereby obtaining another sampling algorithm, with a similar complexity.

References

1 Sergei Adian. Fragments of the word Δ in the braid group. *Matematicheskie Zametki*, 36(1):25–34, 1984. doi:10.1007/BF01139549.

2 Emil Artin. Theorie der Zöpfe. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 4, pages 47–72. Springer, 1925. doi:10.1007/BF02950718.

3 Serge Burckel. The wellordering on positive braids. *Journal of Pure and Applied Algebra*, 120(1):1–17, 1997. doi:10.1016/S0022-4049(96)00072-2.

4 Ruth Charney. Artin groups of finite type are biautomatic. *Mathematische Annalen*, 292(1):671–683, 1992. doi:10.1007/BF01444642.

5 Patrick Dehornoy. Alternating normal forms for braids and locally Garside monoids. *Journal of pure and applied algebra*, 212(11):2413–2439, 2008. doi:10.1016/j.jpaa.2008.03.027.

6 Patrick Dehornoy, François Digne, Eddy Godelle, Daan Krammer, and Jean Michel. *Foundations of Garside theory*, volume 22. Citeseer, 2015. doi:10.4171/139.

7 Pierre Deligne. Les immeubles des groupes de tresses généralisés. *Inventiones mathematicae*, 17:273–302, 1972. doi:10.1007/BF01406236.

8 David Epstein. *Word processing in groups*. CRC Press, 1992. doi:10.1201/9781439865699.

9 Ramón Flores and Juan González-Meneses. On lexicographic representatives in braid monoids. *Journal of Algebraic Combinatorics*, 52(4):561–597, 2020. doi:10.1007/s10801-019-00913-7.

10 Ramón Flores and Juan González-Meneses. On the growth of Artin–Tits monoids and the partial theta function. *Journal of Combinatorial Theory, Series A*, 190:105623, 2022. doi:10.1016/j.jcta.2022.105623.

11 Jean Fromentin. The rotating normal form of braids is regular. *Journal of Algebra*, 501:545–570, 2018. doi:10.1016/j.jalgebra.2018.01.001.

12 Frank Garside. The braid group and other groups. *The Quarterly Journal of Mathematics*, 20(1):235–254, 1969. doi:10.1093/qmath/20.1.235.

13 Volker Gebhardt and Juan González-Meneses. Generating random braids. *Journal of Combinatorial Theory, Series A*, 120(1):111–128, 2013. doi:10.1016/j.jcta.2012.07.003.

14 Jean Michel. A note on words in braid monoids. *Journal of Algebra*, 215(1):366–377, 1999. doi:10.1006/jabr.1998.7723.