Wheeler Bisimulations

Nicola Cotumaccio
University of Helsinki, Finland

Abstract

Over the years, bisimulations have emerged as a pervasive paradigm, finding applications in numerous areas, including concurrency theory, model checking, automata theory, logic, programming languages and category theory. In this paper, we establish a connection between bisimulations and data compression. More precisely, we study the relationship between bisimulations and Wheeler automata (Alanko et al., SODA 2020), a class of automata that has received considerable attention in recent years. The standard notion of bisimulation is not appropriate, so we introduce *Wheeler bisimulations*, that is, bisimulations that respect the convex structure of the considered Wheeler automata. We show that Wheeler bisimilarity induces a *unique* minimal Wheeler NFA (analogously to standard bisimulations). In particular, in the deterministic case, we retrieve the minimal Wheeler deterministic automaton of a given language. We also show that the minimal Wheeler NFA induced by Wheeler bisimulations can be built in linear time. This is in contrast with standard bisimulations, for which the corresponding minimal NFA can be built in $O(m \log n)$ time (where m is the number of edges and n is the number of states) by adapting Paige-Tarjan partition refinement algorithm. Compared to previous state-reduction techniques, our bisimulation-induced construction is the first for which (i) we obtain a *canonical* Wheeler NFA and (ii) the resulting Wheeler NFA can be built in linear time.

2012 ACM Subject Classification Theory of computation → Regular languages; Theory of computation → Pattern matching; Theory of computation → Concurrent algorithms

Keywords and phrases Wheeler automata, bisimulation, minimal automata

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2026.27

Related Version *Full Version*: `https://arxiv.org/abs/2602.07964`

Funding Funded by the Helsinki Institute for Information Technology (HIIT).

Acknowledgements I am grateful to Luca Aceto for his insightful comments and valuable feedback. I also thank the anonymous reviewers for carefully reviewing the paper, and in particular for suggesting a simpler proof of Lemma 2 and pointing out a mistake in an earlier proof of Theorem 29.

1 Introduction

Bisimulations were first introduced by Milner and Park in the early 1980s to model processes that cannot be distinguished by an external agent [52, 55]. Over the years, bisimulations have emerged as a pervasive paradigm, finding applications in numerous areas, including concurrency theory, model checking, automata theory, logic, programming languages and category theory. We refer the reader to standard books on the topic [7, 58, 47, 2]. Here we mainly focus on bisimulations between *automata* (see in particular [47]), but a similar theory of bisimulations can be developed for other labeled systems [7] (see also [64, 13] for Petri nets and [37] for Kripke structures).

Let $\mathcal{A} = (Q, E, s, F)$ be an NFA, with $|Q| = n$ and $|E| = m$. Then, up to isomorphism, there exists a unique minimal NFA bisimilar to $\mathcal{A}$ [47]. The algorithmic problem of computing the minimal bisimilar NFA has a long history. In the deterministic setting, it corresponds to the problem of minimizing a DFA, which can be solved using, e.g., Moore algorithm [53] or Hopcroft algorithm [43]. In the general setting, the problem can be solved using, e.g., Kanellakis-Smolka algorithm [45] or Paige-Tarjan algorithm [54]. The most efficient algorithms are Valmari and Lehtinen's variants of Hopcroft algorithm and Paige-Tarjan algorithm, both running in $O(m \log n)$ time (see [63, 62]). The bound $O(m \log n)$ is achieved under the standard assumption that the edge labels can be sorted in linear time [62, Section 4.3].

In this paper, we establish a connection between bisimulations and data compression. More precisely, we study the relationship between bisimulations and *Wheeler automata* [4, 42], a class of automata that has received considerable attention in recent years (we refer the reader to [31] for a survey). A Wheeler automaton is a (conventional) NFA $\mathcal{A}$ endowed with a total order $\leq$ on the set of all states. Intuitively, if for some state u and v we have $u < v$, then the strings that reach state u must be co-lexicographically smaller than the strings that reach state v, up to intersections (see Section 3 for details).

The idea of ordering the states of an automaton dates back at least to the seventies, when Shyr and Thierrin introduced *ordered automata* [59]. Wheeler automata are perhaps the most elegant and successful implementation of this paradigm:

- A Wheeler automaton can be stored using only $(e \log \sigma)(1 + o(1)) + O(e)$ bits, where e is the number of edges and σ is the size of the alphabet [42]. At the very least we need $e \log \sigma$ bits to store the edge labels (we need $\log \sigma$ bits per edge), so intuitively the topological complexity of a Wheeler automaton can be stored succinctly.
- We can decide whether a string of length m is accepted by a Wheeler automaton in time $O(m \log \log \sigma)$ [42]. For general automata, the time complexity is $\Omega(me^{1-\epsilon})$, where $\epsilon > 0$ is an arbitrarily small constant (under the Orthogonal Vector hypothesis) [40]. Remarkably, the algorithm running in $O(m \log \log \sigma)$ time does not need to access the explicit representation of a Wheeler automaton, but only its compressed representation (consisting of $(e \log \sigma)(1 + o(1)) + O(e)$ bits).

In other words, Wheeler automata extend the most important data structures for solving pattern matching queries on compressed texts (based on the *suffix array* [51], the *Burrows-Wheeler transform* [18] and the *FM-index* [41]) from strings to automata. In particular, Wheeler automata are closely related to Bruijn graphs, which are used in bioinformatics to perform Eulerian sequence assembly (see [16]).

The regular languages recognized by Wheeler automata are called *Wheeler languages*. The class of Wheeler languages is one of the richest subclasses of regular languages:

- If a regular language is recognized by some Wheeler non-deterministic automaton, it is also recognized by some Wheeler deterministic automaton [5].
- Every Wheeler language admits a unique minimal Wheeler deterministic automaton (which, in general, is larger than the standard minimal DFA) [5], and a Wheeler deterministic automaton can be minimized efficiently [3].
- Every Wheeler language admits an algebraic characterization based on the *Myhill-Nerode theorem* [4].
- It is possible to efficiently detect if a regular language is Wheeler by looking at the cycle structure of its (standard) minimal automaton [10].

Among other topics, previous work has investigated string-labeled Wheeler NFAs [25, 30, 26], the state complexity of Wheeler languages [36], their relationship with locally testable languages [9] and circular strings [29, 27], and more sophisticated applications in data compression [21, 33, 6, 24].

1.1 Our Contribution

In the deterministic setting, both regular languages and Wheeler languages admit a *unique* minimal automaton up to isomorphism. Note that, in general, the minimal Wheeler automaton can be larger than the standard minimal automaton [5] (we remark that different trade-offs between the size and the compressibility of a DFA recognizing a given language can be obtained by extending the ideas behind Wheeler automata to arbitrary automata [32, 34]). In the non-deterministic setting, a regular language can admit multiple minimal automata, and the problem of building such a minimal automaton is notoriously PSPACE-complete [60]. By introducing additional constraints, bisimulations make it possible to recover the uniqueness of a minimal automaton in the non-deterministic setting: two bisimilar automata recognize the same language, but two automata that recognize the same language need not be bisimilar [47].

In this paper, we study the minimality problem for Wheeler languages in the non-deterministic setting. We start our investigation by observing that, in general, Wheeler languages are not recognized by a unique minimal Wheeler NFA (Example 6). Then, we study the relationship between Wheeler languages and nondeterminism through the lens of bisimulations. The standard notion of bisimulation is not appropriate (see Example 8), so we introduce *Wheeler bisimulations*, that is, bisimulations that respect the convex structure of the considered Wheeler automata (Definition 9). We show that Wheeler bisimilarity induces a *unique* minimal Wheeler NFA (analogously to standard bisimulations), see Theorem 16 and Corollary 17. In particular, in the deterministic case, we retrieve the minimal Wheeler deterministic automaton of a given language (Theorem 20 and Corollary 21).

We also study Wheeler bisimulations from an algorithmic perspective. We show that the minimal Wheeler NFA induced by Wheeler bisimulations can be computed in linear time (Theorem 22). Consequently, in the Wheeler setting we can do better than in the conventional setting (where the best algorithm runs in $O(m \log n)$ time). We remark that, in general, the minimal Wheeler NFA induced by Wheeler bisimulations can be distinct from the minimal NFA induced by conventional bisimulations. We achieve linear time under the usual assumption that the edge labels can be sorted in linear time, and we complete the picture with a lower bound showing that, in the comparison model, there exists no linear-time algorithm for the same problem (Theorem 29). In linear time we can also decide whether there exists a Wheeler bisimulation between two Wheeler NFAs (Corollary 28).

We can informally summarize our main results as follows.

▶ **Theorem 1.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA. Then, up to isomorphism, there exists a unique minimal Wheeler NFA $(\mathcal{A}_*, \leq_*)$ for which there exists a Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}_*, \leq_*)$. Moreover:*

- *$(\mathcal{A}_*, \leq_*)$ can be built from $(\mathcal{A}, \leq)$ in linear time.*
- *If $(\mathcal{A}, \leq)$ is a Wheeler DFA, then $(\mathcal{A}_*, \leq_*)$ is the minimal Wheeler DFA equivalent to $(\mathcal{A}, \leq)$.*

In linear time we can also decide whether there exists a Wheeler bisimulation between two given Wheeler NFAs.

Consider a Wheeler language $\mathcal{L}$. We know that every Wheeler NFA recognizing $\mathcal{L}$ admits a compressed encoding. To obtain an even more compressed representation of $\mathcal{L}$, we should select a Wheeler NFA with as few states as possible. Several papers have addressed this challenge, typically by devising procedures to collapse some states in a given Wheeler NFA recognizing $\mathcal{L}$ [8, 23, 12, 11]. Remarkably, our bisimulation-induced construction is the first for which (i) we obtain a *canonical* Wheeler NFA and (ii) the resulting Wheeler NFA can be built in linear time. From a broader perspective, bisimulations confirm that Wheeler languages form one of the richest and most robust subclasses of regular languages.

The paper is organized as follows. In Section 2, we discuss notation and preliminary results. In Section 3 we introduce Wheeler automata, and in Section 4 we define Wheeler bisimulations. In Section 5 we prove that Wheeler bisimilarity induces a unique minimal

Wheeler NFA, and in Section 6 we study the algorithmic aspects of Wheeler bisimulations. In Section 7 we present our conclusions and we discuss future work. Due to space constraints, some proofs and some auxiliary results are deferred to the full version on arXiv.

2 Preliminaries

2.1 Relations

Let Q and Q' be sets, and let $R \subseteq Q \times Q'$ be a relation from Q to Q'. For every $u \in Q$ and for every $u' \in Q'$, we write $u \ R \ u'$ if and only if $(u, u') \in R$. For every $u \in Q$, let $R(u) = \{u' \in Q \mid u \ R \ u'\}$, and for every $U \subseteq Q$, let $R(U) = \bigcup_{u \in U} R(u)$. If R is a function from Q to Q' and $u \in Q$, we identify $R(u)$ and its unique element.

If $R \subseteq Q \times Q'$ is a relation, let $R^{-1} \subseteq Q' \times Q$ be the inverse relation. If $R \subseteq Q \times Q'$ and $R' \subseteq Q' \times Q''$ are relations, let $R' \circ R \subseteq Q \times Q''$ be the composition of the two relations. Let $id_Q \subseteq Q \times Q$ be the identity function from Q to Q.

An equivalence relation on Q is typically denoted by $\equiv$ or $\sim$. For every $u \in Q$, let $[u]_{\equiv} = \{v \in Q \mid u \equiv v\}$. Then, $\{[u]_{\equiv} \mid u \in Q\}$ is a partition of Q, denoted by $\mathcal{P}_{\equiv}$. Every element of $\mathcal{P}_{\equiv}$ is an $\equiv$-equivalence class. A total order on Q is typically denoted by $\leq$. For every $u, v \in Q$, we write $u < v$ if $(u \leq v) \wedge (u \neq v)$.

2.2 Convexity

To introduce Wheeler bisimulations (Definition 9), we will need the notion of *convexity*. Let Q be a set, and let $\leq$ be a total order on Q. We say that a subseteq $C \subseteq Q$ is $\leq$-*convex* if for every $u, v, w \in Q$ such that $u \leq v \leq w$ and $u, w \in C$ we have $v \in C$. In particular, $\emptyset$ and Q are $\leq$-convex, and $\{u\}$ is $\leq$-convex for every $u \in Q$.

It is easy to check that, if C_1 and C_2 are $\leq$-convex, then $C_1 \cap C_2$ is $\leq$-convex. However, in general, $C_1 \cup C_2$ need not be $\leq$-convex. For example, if $Q = \{1, 2, 3, 4, 5\}$, $\leq$ is the total order on Q such that $1 < 2 < 3 < 4 < 5$, $C_1 = \{1, 2\}$ and $C_2 = \{4, 5\}$, then C_1 and C_2 are $\leq$-convex, but $C_1 \cup C_2$ is not $\leq$-convex. Nonetheless, if $C_1 \cap C_2 \neq \emptyset$, then $C_1 \cup C_2$ must be $\leq$-convex, as shown in the following lemma.

▶ **Lemma 2.** *Let Q be a set, and let $\leq$ be a total order on Q. Let C_1 and C_2 be $\leq$-convex sets. If $C_1 \cap C_2 \neq \emptyset$, then $C_1 \cup C_2$ is $\leq$-convex.*

Proof. Since $C_1 \cap C_2 \neq \emptyset$, we can pick $z \in C_1 \cap C_2$. Let $u, v, w \in Q$ such that $u \leq v \leq w$ and $u, w \in C_1 \cup C_2$. We must prove that $v \in C_1 \cup C_2$. Assume that $v \leq z$ (the case $z < v$ is analogous). Then, $u \leq v \leq z$. Assume that $u \in C_1$ (the case $u \in C_2$ is analogous). From $u \leq v \leq z$ and $u, z \in C_1$ we obtain $v \in C_1$ because C_1 is $\leq$-convex, so we conclude $v \in C_1 \cup C_2$. ◀

Lastly, we extend the notion of convexity to partitions and equivalence relations. Let Q be a set, and let $\leq$ be a total order on Q. A partition $\mathcal{P}$ of Q is $\leq$-convex if every $U \in \mathcal{P}$ is $\leq$-convex. An equivalence relation $\equiv$ on Q is $\leq$-convex if the partition $\mathcal{P}_{\equiv}$ of Q is $\leq$-convex (equivalently, if $[u]_{\equiv}$ is $\leq$-convex for every $u \in Q$).

2.3 Automata

Let Σ be a fixed finite alphabet. We denote by Σ^* the set of all finite strings over Σ, where $\epsilon \in \Sigma^*$ is the empty string. Given a language $\mathcal{L} \subseteq \Sigma^*$, we denote by $\text{Pref}(\mathcal{L})$ the set of all prefixes of some element of $\mathcal{L}$, that is, $\text{Pref}(\mathcal{L}) = \{\alpha \in \Sigma^* \mid (\exists \alpha' \in \Sigma^*)(\alpha\alpha' \in \mathcal{L})\}$.

A non-deterministic finite automaton (NFA) is a 4-tuple $\mathcal{A} = (Q, E, s, F)$, where Q is the finite set of states, $E \subseteq Q \times Q \times \Sigma$ is the set of edges (also known as transitions), $s \in Q$ is the (unique) initial state and $F \subseteq Q$ is the set of final states. We denote by $\mathcal{L}(\mathcal{A})$ the language recognized by $\mathcal{A}$. Throughout the paper, we assume that, for every $u \in Q$, we have that u is reachable in $\mathcal{A}$ (that is, there is a walk from the initial state s to u) and u is co-reachable in $\mathcal{A}$ (that is, there is a walk from u to a final state, and in particular u is co-reachable if $u \in F$). These are standard assumptions in automata theory because, if a state u is not reachable or is not co-reachable, then it can be removed without changing $\mathcal{L}(\mathcal{A})$ (as long as the set of states does not become empty). Note that, under these assumptions, the empty language is not recognized by any NFA.

Let $\mathcal{A} = (Q, E, s, F)$ be an NFA. For every $u \in Q$, we denote by $I_u^{\mathcal{A}}$ the set of all $\alpha \in \Sigma^*$ such that α can be read following some walk from s to u (in particular, $\epsilon \in I_s^{\mathcal{A}}$). Note that $\mathcal{L}(\mathcal{A}) = \bigcup_{u \in F} I_u^{\mathcal{A}}$. Moreover, we have $I_u^{\mathcal{A}} \neq \emptyset$ for every $u \in Q$ because every $u \in Q$ is reachable in $\mathcal{A}$, and $\mathrm{Pref}(\mathcal{L}(\mathcal{A})) = \bigcup_{u \in Q} I_u^{\mathcal{A}}$ because every $u \in Q$ is co-reachable in $\mathcal{A}$.

An NFA $\mathcal{A} = (Q, E, s, F)$ is a deterministic finite automaton (DFA) if for every $u \in Q$ and for every $a \in \Sigma$ there exists at most one $v \in Q$ such that $(u, v, a) \in E$ (so in this paper a DFA needs not be total). If $\mathcal{A} = (Q, E, s, F)$ is a DFA, then for every $u, v \in Q$ such that $u \neq v$ we have $I_u^{\mathcal{A}} \cap I_v^{\mathcal{A}} = \emptyset$.

Let $\mathcal{A} = (Q, E, s, F)$ and $\mathcal{A}' = (Q', E', s', F')$ be two NFAs. A bijective function ϕ from Q to Q' is an isomorphism from $\mathcal{A}$ to $\mathcal{A}'$ if (i) for every $u, v \in Q$ and for every $a \in \Sigma$ we have $(u, v, a) \in E$ if and only if $(\phi(u), \phi(v), a) \in E'$, (ii) $\phi(s) = s'$ and (iii) for every $u \in Q$, $u \in F$ if and only if $\phi(u) \in F'$.

When we consider a class C of NFAs, we say that an element $\mathcal{A}$ of C is a minimal element of C if every element of C has at least as many states as $\mathcal{A}$.

3 Wheeler Automata

As customary in the literature on Wheeler languages (e.g., [4, 32, 21]), we fix a total order $\preceq$ on Σ. In our examples, we always assume that Σ is the English alphabet and $\preceq$ is the usual total order on Σ such that $a \prec b$, $b \prec c$, $c \prec d$, and so on. Moreover, we extend $\preceq$ to Σ^* *co-lexicographically*, that is, for every $\alpha \in \Sigma^*$ we have $\alpha \prec \beta$ if and only if the reverse string α^R is smaller than the reverse string β^R in the standard dictionary order. Formally, let γ be the longest common suffix of α and β (possibly, $\gamma = \epsilon$), and let $\alpha', \beta' \in \Sigma^*$ such that $\alpha = \alpha'\gamma$ and $\beta = \beta'\gamma$. Then, we have $\alpha \prec \beta$ if and only $\beta' \neq \epsilon$ and one of the following is true: (i) $\alpha' = \epsilon$ or (ii) $\alpha' \neq \epsilon$ and $a \prec b$, where a is the last character of α' and b is the last character of β'. For example, we have $a \prec aa$ and $dba \prec ca$.

Let us recall the definition of Wheeler automata.

▶ **Definition 3.** *Let $\mathcal{A} = (Q, E, s, F)$ be an NFA, and let $\leq$ be a total order on Q. We say that $(\mathcal{A}, \leq)$ is a* Wheeler NFA *if:*

- *(Axiom 1) For every $u \in Q$, $s \leq u$.*
- *(Axiom 2) For every $u, v, u', v' \in Q$ and for every $a, b \in \Sigma$, if $(u, v, a), (u', v', b) \in E$ and $v < v'$, then $a \preceq b$.*
- *(Axiom 3) For every $u, v, u', v' \in Q$ and for every $a \in \Sigma$, if $(u, v, a), (u', v', a) \in E$ and $v < v'$, then $u \leq u'$.*

We say that $\leq$ is a Wheeler order *on $\mathcal{A}$. If $\mathcal{A}$ is a DFA, we say that $(\mathcal{A}, \leq)$ is a* Wheeler DFA. *Moreover, $\mathcal{L} \subseteq \Sigma^*$ is a* Wheeler language *if there exists a Wheeler NFA $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.*

See Figure 1 for an example of a Wheeler NFA and for a graphical interpretation of Axiom 3 (if we list all states consistently with the Wheeler order, we duplicate the list, and we draw the edges of the Wheeler NFA starting from the bottom list and reaching the top list, then Axiom 3 ensures that equally labeled edges do not cross). This graphical interpretation is useful to follow some proofs (see for example Lemma 24).

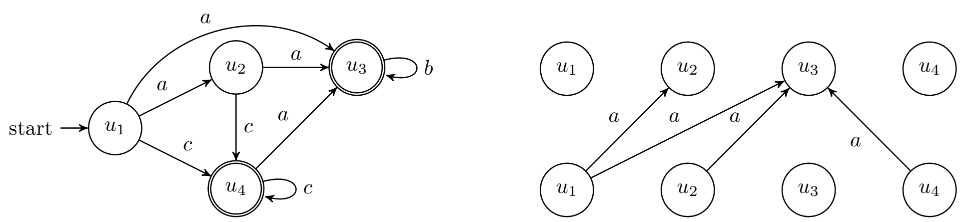

Figure 1 *Left:* A Wheeler NFA. The states are numbered following the Wheeler order. *Right:* Edges labeled with the same character (e.g., a) do not cross.

Let us recall some properties of Wheeler NFAs.

▶ **Lemma 4** ([5, Lemma 3.7]). *Let $\mathcal{L} \subseteq \Sigma^*$ be a Wheeler language. Then, there exists a Wheeler DFA $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.*

In other words, every language recognized by a Wheeler NFA is also recognized by a Wheeler DFA.

▶ **Lemma 5** ([5, Lemma 3.2]). *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Let $u, v \in Q$, and let $\alpha \in I_u^{\mathcal{A}}$ and $\beta \in I_v^{\mathcal{A}}$ such that $\{\alpha, \beta\} \not\subseteq I_u^{\mathcal{A}} \cap I_v^{\mathcal{A}}$. Then, $u < v$ if and only if $\alpha \prec \beta$.*

In other words, a Wheeler order must be consistent with the strings reaching each state, up to intersections. For example, in Figure 1 we have $a \in I_{u_2}^{\mathcal{A}}$, $aa \in I_{u_3}^{\mathcal{A}}$, $\{a, aa\} \not\subseteq I_{u_2}^{\mathcal{A}} \cap I_{u_3}^{\mathcal{A}}$, $2 < 3$ and, consistently, $a \prec aa$. If we interpret the strings in $I_u^{\mathcal{A}}$ as the set of all "prefixes" of u, then Lemma 5 intuitively states that a Wheeler order must (co-lexicographically) sort the prefixes of the states. This explains why Wheeler automata can be seen as a non-trivial extension of *prefix arrays* and *suffix arrays* from strings to automata, which was the original motivation for studying Wheeler automata [42]. The condition $\{\alpha, \beta\} \not\subseteq I_u^{\mathcal{A}} \cap I_v^{\mathcal{A}}$ of Lemma 5 implies that the order between two states is not disambiguated by the strings reaching both states. For example, consider the NFA $\mathcal{A} = (Q, E, s, F)$, where $Q = \{s, u_1, u_2, u_3\}$ consists of four pairwise distinct states, $E = \{(s, u_1, a), (s, u_1, b), (u_1, u_2, c), (u_1, u_3, c)\}$ and $F = \{u_2, u_3\}$. Let $\leq$ be the total order on Q such that $s < u_1 < u_2 < u_3$. Then, $(\mathcal{A}, \leq)$ is a Wheeler NFA. Notice that $bc \in I_{u_2}^{\mathcal{A}}$, $ac \in I_{u_3}^{\mathcal{A}}$, $u_2 < u_3$ and $ac \prec bc$. This does not contradict Lemma 5 because $\{ac, bc\} \subseteq I_{u_2}^{\mathcal{A}} \cap I_{u_3}^{\mathcal{A}}$.

Recall that, if $\mathcal{A}$ is a DFA, then $u \neq v$ implies $I_u^{\mathcal{A}} \cap I_v^{\mathcal{A}} = \emptyset$. This means that, if $(\mathcal{A}, \leq)$ is a Wheeler DFA, Lemma 5 can be restated as follows: let $u, v \in Q$ such that $u \neq v$, and let $\alpha \in I_u^{\mathcal{A}}$ and $\beta \in I_v^{\mathcal{A}}$; then, $u < v$ if and only if $\alpha \prec \beta$. In particular, if $\mathcal{A}$ is a DFA, then there exists *at most* one Wheeler order $\leq$ on $\mathcal{A}$, because the order between two states $u, v \in Q$ such that $u \neq v$ is uniquely determined by any $\alpha \in I_u^{\mathcal{A}}$ and $\beta \in I_v^{\mathcal{A}}$. In other words, if $(\mathcal{A}, \leq)$ is a Wheeler DFA, then $\leq$ is uniquely determined by $\mathcal{A}$. In general, this is not true for Wheeler NFAs. For example, if we consider the NFA $\mathcal{A} = (Q, E, s, F)$, where $Q = \{s, u, v\}$ consists of three pairwise distinct states, $E = \{(s, u, a), (s, v, a)\}$ and $F = \{u, v\}$, then both total orders in which s comes first are Wheeler orders on $\mathcal{A}$.

Every Wheeler language is star-free [5], so $(aa)^*$ is not Wheeler because $(aa)^*$ is the classical example of a regular language that is not star-free [61]. A more interesting example is the regular language $\mathcal{L} = ae^*b + ce^*d$ on the alphabet $\Sigma = \{a, b, c, d, e\}$. Notice that $\mathcal{L}$ is star-free because e^* is the complement of $\Sigma^*(a + b + c + d)\Sigma^*$, where Σ^* is the complement of $\emptyset$. However, $\mathcal{L}$ is not Wheeler, as proved next (we adapt [5, Example 3]). If $\mathcal{L}$ were Wheeler, then $\mathcal{L}$ would be recognized by a Wheeler DFA $(\mathcal{A}, \leq)$ by Lemma 4. Let $\alpha_i = ae^i$ if i is odd and $\alpha_i = ce^i$ if i is even. For every i we have $\alpha_i \in \mathrm{Pref}(\mathcal{L})$, so there exists a (unique) state u_i of $\mathcal{A}$ such that $\alpha_i \in I^{\mathcal{A}}_{u_i}$. Notice that $\alpha_1 \prec \alpha_2 \prec \alpha_3 \prec \dots$, so by Lemma 5 we have $u_1 \leq u_2 \leq u_3 \leq \dots$, and the finiteness of the number of states implies that $u_i = u_{i+1} = u_{i+2} = \dots$ for some i. We can assume that i is odd. Then, $\alpha_i, \alpha_{i+1} \in I^{\mathcal{A}}_{u_i}$ (the two strings reach the same state), so $\alpha_i b \in \mathcal{L}$ if and only if $\alpha_{i+1} b \in \mathcal{L}$. This is a contradiction because $\alpha_i b = ae^i b \in \mathcal{L}$ and $\alpha_{i+1} b = ce^{i+1} b \notin \mathcal{L}$.

4 Wheeler Bisimulations

Let $\mathcal{L} \subseteq \Sigma^*$ be a Wheeler language. As mentioned in the introduction, up to isomorphism there exists a unique (state)-minimal Wheeler DFA $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$ [5]. The first result in the paper is a simple example showing that this is not true for Wheeler NFAs (analogously, in general there is not a unique minimal NFA recognizing a given regular language).

▶ **Example 6.** Consider the regular language $\mathcal{L} = aa^*$. No (Wheeler) NFA with one state can recognize $\mathcal{L}$, and Figure 2 shows two non-isomorphic Wheeler NFAs with two states recognizing $\mathcal{L}$.

Figure 2 Two non-isomorphic minimal Wheeler NFAs recogning $\mathcal{L} = aa^*$ (see Example 6). The states of each NFA are numbered following the corresponding Wheeler order.

Our goal is to introduce a meaningful notion of minimality for Wheeler NFAs that ensures uniqueness up to isomorphism. To this end, let us recall the standard definition of bisimulation (see, e,g., [47, 7]).

▶ **Definition 7.** *Let $\mathcal{A}$ and $\mathcal{A}'$ be NFAs, with $\mathcal{A} = (Q, E, s, F)$ and $\mathcal{A}' = (Q', E', s', F')$. A relation $R \subseteq Q \times Q'$ is a* bisimulation *from $\mathcal{A}$ to $\mathcal{A}'$ if:*

- *For every $u, v \in Q$, $u' \in Q'$ and $a \in \Sigma$, if $u \mathrel{R} u'$ and $(u, v, a) \in E$, then there exists $v' \in Q'$ such that $v \mathrel{R} v'$ and $(u', v', a) \in E'$.*
- *For every $u \in Q$, $u', v' \in Q'$ and $a \in \Sigma$, if $u \mathrel{R} u'$ and $(u', v', a) \in E'$, then there exists $v \in Q$ such that $v \mathrel{R} v'$ and $(u, v, a) \in E$.*
- *We have $s \mathrel{R} s'$.*
- *For every $u \in Q$ and $u' \in Q$ such that $u \mathrel{R} u'$, we have $u \in F$ if and only if $u' \in F'$.*

Let us recall why bisimulations lead to a notion of uniqueness for (standard) NFAs. Fix an NFA $\mathcal{A}$, and consider the class of all NFAs $\mathcal{A}'$ such that there exists a bisimulation from $\mathcal{A}$ to $\mathcal{A}'$. Then, all the *minimal* NFAs in this class (that is, all the NFAs in the class having the minimum number of states) must be isomorphic [47, 7]. In other words, up to isomorphism, there exists only one NFA $\mathcal{A}' = (Q', E', s', F')$ for which (i) there exists a bisimulation from $\mathcal{A}$ to $\mathcal{A}'$ and (ii) for every NFA $\mathcal{A}'' = (Q'', E'', s'', F'')$ such that $|Q''| < |Q'|$, there exists no bisimulation from $\mathcal{A}$ to $\mathcal{A}''$.

It is tempting to directly use the notion of bisimulation on Wheeler NFAs to identify a unique minimal Wheeler NFA. However, the next example shows that such an attempt fails.

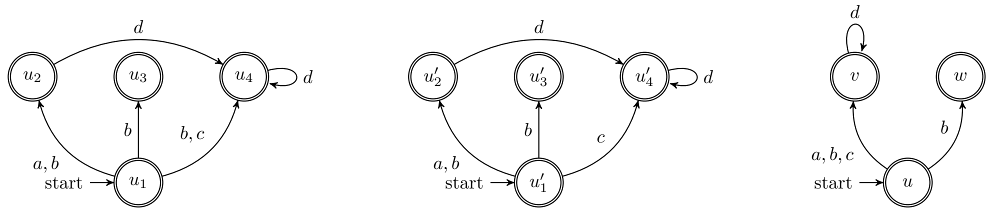

Figure 3 The three NFAs used in Example 8. On the left, the Wheeler NFA $(\mathcal{A}, \leq)$. In the center, the Wheeler NFA $(\mathcal{A}', \leq')$. On the right, the NFA $\mathcal{A}''$. The states of $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ are numbered following the corresponding Wheeler orders.

▶ **Example 8.** Consider the Wheeler NFAs $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ in Figure 3, where $\mathcal{A} = (Q, E, s, F)$ and $\mathcal{A}' = (Q', E', s', F')$. Let $R \subseteq Q \times Q'$ be the relation described by the following pairs: $u_1 \; R \; u'_1$, $u_2 \; R \; u'_2$, $u_3 \; R \; u'_3$, $u_4 \; R \; u'_4$, $u_4 \; R \; u'_2$. It is easy to check that R is a bisimulation from $\mathcal{A}$ to $\mathcal{A}'$. Moreover, id_Q is a bisimulation from $\mathcal{A}$ to $\mathcal{A}$.

We want to show that every Wheeler NFA $(\mathcal{A}^{\#}, \leq^{\#})$ for which there exists a bisimulation from $\mathcal{A}$ to $\mathcal{A}^{\#}$ must have at least four states. Since $\mathcal{A}$ and $\mathcal{A}'$ are non-isomorphic and have four states, the notion of bisimulation does not lead to a *unique* minimal Wheeler NFAs.

The NFA $\mathcal{A}'' = (Q'', E'', s'', F'')$ in Figure 3 is a minimal NFA such that there exists a bisimulation from $\mathcal{A}$ to $\mathcal{A}''$, and from the theory of bisimulation we know that $\mathcal{A}''$ is unique up to isomorphism [47, 7]. Since $\mathcal{A}''$ has three states, we only need to prove that there exists no total order $\leq''$ on Q'' such that $(\mathcal{A}'', \leq'')$ is a Wheeler NFA, because this implies that every Wheeler NFA $(\mathcal{A}^{\#}, \leq^{\#})$ for which there exists a bisimulation from $\mathcal{A}$ to $\mathcal{A}^{\#}$ must have at least four states. Suppose for the sake of a contradiction that there exists a total order $\leq''$ on Q such that $(\mathcal{A}'', \leq'')$ is a Wheeler NFA. Let us show that both $v < w$ and $w < v$ lead to a contradiction. We cannot have $v < w$ because from $(u, v, c), (u, w, b) \in E''$ we would obtain $c \preceq b$ by Axiom 2. We cannot have $w < v$ because from $(u, w, b), (u, v, a) \in E''$ we would obtain $b \preceq a$ again by Axiom 2.

We conclude that, in general, the standard notion of bisimulation does not induce a unique minimal Wheeler NFA.

In view of Example 8, to retrieve a unique minimal Wheeler NFA, we now define *Wheeler bisimulations*. The main idea is that we should only consider bisimulations that respect the convex structure of the considered Wheeler NFAs.

▶ **Definition 9.** *Let $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ be Wheeler NFAs, with $\mathcal{A} = (Q, E, s, F)$ and $\mathcal{A}' = (Q', E', s', F')$. A relation $R \subseteq Q \times Q'$ is a* Wheeler bisimulation *from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$ if:*

- *(Property 1) R is a bisimulation from $\mathcal{A}$ to $\mathcal{A}'$.*
- *(Property 2) For every $C \subseteq Q$, if C is $\leq$-convex, then $R(C)$ is $\leq'$-convex.*
- *(Property 3) For every $C' \subseteq Q'$, if C' is $\leq'$-convex, then $R^{-1}(C')$ is $\leq$-convex.*

Notice that the bisimulation R defined in Example 8 is not a Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$, because $R(\{u_4\}) = \{u'_2, u'_4\}$, $\{u_4\}$ is $\leq$-convex, but $\{u'_2, u'_4\}$ is not $\leq'$-convex.

Consider the class $\mathfrak{C}$ of all Wheeler NFAs, and for all Wheeler NFAs $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ let $(\mathcal{A}, \leq) \sim_{\mathfrak{C}} (\mathcal{A}', \leq')$ if and only if there exists a Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$. It is not difficult to see that $\sim_{\mathfrak{C}}$ is an equivalence relation on $\mathfrak{C}$. Consequently, our goal is to prove that, in every $\sim_{\mathfrak{C}}$-equivalence class, all Wheeler NFAs having the minimum number of states must be isomorphic.

Let $\mathcal{L} \subseteq \Sigma^*$. Note that the class of all Wheeler NFAs $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$ may intersect infinitely many $\sim_{\mathfrak{C}}$-equivalence classes, as shown in the next example.

▶ **Example 10.** Consider the language $\mathcal{L} = a^*$. Fix $k \geq 3$. Let $\mathcal{A}^{(k)} = (Q^{(k)}, E^{(k)}, s^{(k)}, F^{(k)})$ be the NFA defined as follows. Let $Q^{(k)} = \{u_1^{(k)}, u_2^{(k)}, \dots, u_k^{(k)}\}$ be a set consisting of k pairwise distinct states, $E^{(k)} = \{(u_i^{(k)}, u_{i+1}^{(k)}, a) \mid 1 \leq i \leq k-1\} \cup \{(u_1^{(k)}, u_1^{(k)}, a)\}$, $s^{(k)} = u_1^{(k)}$ and $F^{(k)} = \{u_1^{(k)}, u_k^{(k)}\}$. Moreover, let $\leq^{(k)}$ be the total order on $Q^{(k)}$ such that $u_1^{(k)} < u_2^{(k)} < \dots < u_k^{(k)}$. It is immediate to check that $(\mathcal{A}^{(k)}, \leq^{(k)})$ is a Wheeler NFA such that $\mathcal{L}(\mathcal{A}^{(k)}) = \mathcal{L}$ (see Figure 4).

Fix $3 \leq h < k$. Let us prove that there exists no Wheeler bisimulation from $(\mathcal{A}^{(h)}, \leq^{(h)})$ to $(\mathcal{A}^{(k)}, \leq^{(k)})$. Suppose for the sake of a contradiction that there exists a Wheeler bisimulation $R \subseteq Q^{(h)} \times Q^{(k)}$ from $\mathcal{A}^{(h)}$ to $\mathcal{A}^{(k)}$.

Let us prove that $u_i^{(h)} \ R \ u_i^{(k)}$ for every $1 \leq i \leq h$. We proceed by induction on i. Since R is a bisimulation from $\mathcal{A}^{(h)}$ to $\mathcal{A}^{(k)}$ by Property 1, we have $s^{(u)} \ R \ s^{(k)}$, or equivalently, $u_1^{(h)} \ R \ u_1^{(k)}$. Now assume that $2 \leq i \leq h$. By the inductive hypothesis, we know that $u_{i-1}^{(h)} \ R \ u_{i-1}^{(k)}$. We know that R is a bisimulation from $\mathcal{A}^{(h)}$ to $\mathcal{A}^{(k)}$, so from $(u_{i-1}^{(h)}, u_i^{(h)}, a) \in E^{(h)}$ we obtain that there exists $u' \in Q^{(k)}$ such that $u_i^{(h)} \ R \ u'$ and $(u_{i-1}^{(k)}, u', a) \in E^{(k)}$. If $i \geq 3$, we must necessarily have $u' = u_i^{(k)}$ by the definition of $E^{(k)}$. If $i = 2$, by the definition of $E^{(k)}$ we have $(u' = u_2^{(k)}) \vee (u' = u_1^{(k)})$. However, the case $u' = u_1^{(k)}$ leads to a contradiction because R is a bisimulation from $\mathcal{A}^{(h)}$ to $\mathcal{A}^{(k)}$ and we would obtain $u_2^{(h)} \ R \ u_1^{(k)}$, with $u_2^{(h)} \notin F^{(h)}$ (because $h \geq 3$) and $u_1^{(k)} \in F^{(k)}$. We conclude that in both cases we must have $u' = u_i^{(k)}$, so from $u_i^{(h)} \ R \ u'$ we obtain $u_i^{(h)} \ R \ u_i^{(k)}$.

In particular, we have $u_h^{(h)} \ R \ u_h^{(k)}$. This is a contradiction because R is a bisimulation from $\mathcal{A}^{(h)}$ to $\mathcal{A}^{(k)}$, $u_h^{(h)} \notin F^{(h)}$ and $u_h^{(k)} \in F^{(k)}$ (because $3 \leq h < k$).

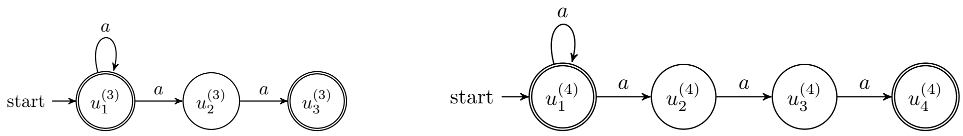

Figure 4 The Wheeler NFAs $(\mathcal{A}^{(3)}, \leq^{(3)})$ and $(\mathcal{A}^{(4)}, \leq^{(4)})$ used in Example 10.

5 Minimality

Let $\mathcal{A}$ be an NFA. An *autobisimulation* on $\mathcal{A}$ is a bisimulation from $\mathcal{A}$ to $\mathcal{A}$. It is well known that every NFA $\mathcal{A}$ admits a *maximal* autobisimulation which can be used to quotient $\mathcal{A}$, and the quotient NFA is the *minimal* NFA bisimilar to $\mathcal{A}$ [47, 7].

In this section, our goal is to extend these results to Wheeler NFAs and Wheeler bisimulations. We define *Wheeler autobisimulations* slightly differently from conventional bisimulations. The key idea is to add one more requirement, namely, *reflexivity*. The intuition is that, in this way, we can use Lemma 2 to infer that the union of Wheeler autobisimulations *must* be a Wheeler autobisimulation.

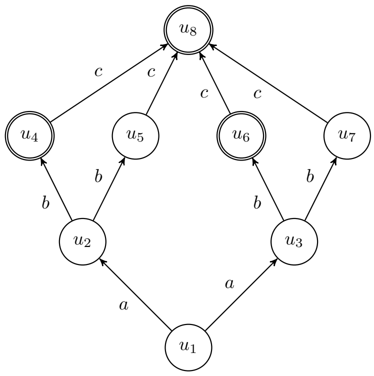

Figure 5 The Wheeler NFA $(\mathcal{A}, \leq)$ used in Example 13. The states are numbered following the Wheeler order $\leq$.

▶ **Definition 11.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. A relation $R \subseteq Q \times Q$ is a* Wheeler autobisimulation *on $(\mathcal{A}, \leq)$ if R is a reflexive Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}, \leq)$.*

We can now show that every Wheeler NFA $(\mathcal{A}, \leq)$ admits a maximum Wheeler autobisimulation $\equiv_{\mathcal{A},\leq}$. Crucially, $\equiv_{\mathcal{A},\leq}$ is always $\leq$-convex.

▶ **Theorem 12.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Then, there exists a (unique) maximum Wheeler autobisimulation $\equiv_{\mathcal{A},\leq}$ on $(\mathcal{A}, \leq)$, that is, a Wheeler autobisimulation $\equiv_{\mathcal{A},\leq}$ on $(\mathcal{A}, \leq)$ that contains every Wheeler autobisimulation on $(\mathcal{A}, \leq)$. Moreover, $\equiv_{\mathcal{A},\leq}$ is a $\leq$-convex equivalence relation on Q.*

Fix a Wheeler NFA $(\mathcal{A}, \leq)$. Consider the maximum Wheeler autobisimulation $\equiv_{\mathcal{A},\leq}$ on $(\mathcal{A}, \leq)$, and consider the conventional maximum autobisimulation $\equiv_{\mathcal{A}}$ on $\mathcal{A}$. Both $\equiv_{\mathcal{A},\leq}$ and $\equiv_{\mathcal{A}}$ are always equivalence relations, and $\equiv_{\mathcal{A},\leq}$ is always $\leq$-convex (see Theorem 12 and [47]). In particular, $\equiv_{\mathcal{A},\leq}$ is a conventional autobisimulation (by Property 1), so by the maximality of $\equiv_{\mathcal{A}}$ we obtain that every $\equiv_{\mathcal{A},\leq}$-equivalence class is contained in some $\equiv_{\mathcal{A}}$-equivalence class. However, the next example shows that, in general, $\equiv_{\mathcal{A},\leq}$ cannot be obtained from $\equiv_{\mathcal{A}}$ by simply splitting each $\equiv_{\mathcal{A}}$-equivalence class into its $\leq$-convex components.

▶ **Example 13.** Let $(\mathcal{A}, \leq)$ be the Wheeler NFA in Figure 5. The $\equiv_{\mathcal{A}}$-equivalence classes are $\{u_1\}$, $\{u_2, u_3\}$, $\{u_4, u_6\}$, $\{u_5, u_7\}$, $\{u_8\}$ and the $\equiv_{\mathcal{A},\leq}$-equivelence classes are $\{u_1\}$, $\{u_2\}$, $\{u_3\}$, $\{u_4\}$, $\{u_5\}$, $\{u_6\}$, $\{u_7\}$, $\{u_8\}$, so the $\equiv_{\mathcal{A}}$-equivalence class $\{u_2, u_3\}$ must be split into $\{u_2\}$ and $\{u_3\}$ even if the set $\{u_2, u_3\}$ is $\leq$-convex.

Let us show how to define the quotient Wheeler NFA $(\mathcal{A}_*, \leq_*)$ starting from the Wheeler NFA $(\mathcal{A}, \leq)$. The key point is that the quotient NFA $\mathcal{A}_*$ can always be endowed with a Wheeler order $\leq_*$.

▶ **Lemma 14.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Let $Q_* = \mathcal{P}_{\equiv_{\mathcal{A},\leq}}$, and let $\leq_*$ be the relation from Q_* to Q_* such that, for every $U, V \in Q_*$, we have $U \leq_* V$ if and only if there exist $u \in U$ and $v \in V$ such that $u \leq v$.*

1. *Let $U, V \in Q_*$ be such that $U <_* V$. Then, for every $u \in U$ and for every $v \in V$ we have $u < v$.*
2. *$\leq_*$ is a total order on Q_*.*

3. *Let* $\mathcal{A}_* = (Q_*, E_*, s_*, F_*)$, *where:*
   - $E_* = \{(U, V, a) \in Q_* \times Q_* \times \Sigma \mid (\exists u \in U)(\exists v \in V)((u, v, a) \in E)\}$.
   - $s_* = [s]_{\equiv_{\mathcal{A},\leq}}$.
   - $F_* = \{U \in Q_* \mid (\exists u \in U)(u \in F)\}$.

   Then, $(\mathcal{A}_*, \leq_*)$ *is a Wheeler NFA.*
4. *If* $\mathcal{A}$ *is a DFA, then* $\mathcal{A}_*$ *is a DFA.*

The following lemma shows that, if there exists a Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$, then the corresponding quotients must be isomorphic, and the isomorphism is witnessed by a bisimulation R that respects the convex structure of each quotient.

▶ **Lemma 15.** *Consider a* $\sim_{\mathfrak{C}}$*-equivalence class* C*, let* $(\mathcal{A}, \leq)$ *and* $(\mathcal{A}', \leq')$ *be two elements of* C*, and let* $(\mathcal{A}_*, \leq_*)$ *and* $(\mathcal{A}'_*, \leq'_*)$ *be the corresponding Wheeler NFAs built in Lemma 14, with* $\mathcal{A}_* = (Q_*, E_*, s_*, F_*)$ *and* $\mathcal{A}'_* = (Q'_*, E'_*, s'_*, F'_*)$*. Let* $Q_* = \{U_1, U_2, \ldots, U_n\}$ *and* $Q'_* = \{U'_1, U'_2, \ldots, U'_{n'}\}$*, where* $U_1 <_* U_2 <_* \cdots <_* U_n$ *and* $U'_1 <'_* U'_2 <'_* \cdots <'_* U'_{n'}$*. Then,* $n = n'$*. Moreover, let* ϕ *be the function from* Q_* *to* Q'_* *such that* $\phi(U_i) = U'_i$ *for every* $1 \leq i \leq n$*. Then, (i)* ϕ *is a Wheeler bisimulation from* $(\mathcal{A}_*, \leq_*)$ *to* $(\mathcal{A}'_*, \leq'_*)$*, (ii)* ϕ *is the* unique *Wheeler bisimulation from* $(\mathcal{A}_*, \leq_*)$ *to* $(\mathcal{A}'_*, \leq'_*)$*, and (iii)* ϕ *is an isomorphism from* $\mathcal{A}_*$ *to* $\mathcal{A}'_*$*.*

Proof (Sketch). One can show that there must exist a Wheeler bisimulation R from $(\mathcal{A}_*, \leq_*)$ to $(\mathcal{A}'_*, \leq'_*)$, and R must be an isomorphism (which implies $n = n'$). To conclude the proof, we only need to prove that we must have $R(U_i) = U'_i$ for every $1 \leq i \leq n$. We proceed by induction on i. For $i = 1$, we have $U_1 = s_*$ and $U'_1 = s'_*$ by Axiom 1, and we know that $R(s_*) = s'_*$ because R is an isomorphism from $\mathcal{A}_*$ to $\mathcal{A}'_*$, so $R(U_i) = R(s_*) = s'_* = U'_1$. Now assume that $2 \leq i \leq n$. By the inductive hypothesis, we know that $R(U_j) = U'_j$ for every $1 \leq j \leq i - 1$. We must show that $R(U_i) = U'_i$. Let $1 \leq k \leq n$ be such that $R(U_i) = U'_k$. We must prove that $k = i$. We only need to prove that we cannot have $k < i$ and we cannot have $i < k$. We cannot have $k < i$ because by the inductive hypothesis we would have $R(U_k) = U'_k$, so we would have $\{U_k, U_i\} \subseteq R^{-1}(U'_k)$ and R would not be injective. Moreover, we cannot have $i < k$ because by the inductive hypothesis we would have $R(\{U_1, U_2, \ldots, U_{i-2}, U_{i-1}, U_i\}) = \{U'_1, U'_2, \ldots, U'_{i-2}, U'_{i-1}, U'_k\}$, which would be a contradiction because R is a Wheeler bisimulation from $(\mathcal{A}_*, \leq_*)$ to $(\mathcal{A}'_*, \leq'_*)$, $\{U_1, U_2, \ldots, U_{i-2}, U_{i-1}, U_i\}$ is $\leq_*$-convex, but $\{U'_1, U'_2, \ldots, U'_{i-2}, U'_{i-1}, U'_k\}$ would not be $\leq'_*$-convex. ◀

We can now prove that, if we consider the $\sim_{\mathfrak{C}}$-equivalence class C of a Wheeler NFA $(\mathcal{A}, \leq)$, then $(\mathcal{A}_*, \leq_*)$ is a minimal element of C and, up to isomorphism, is the *unique* minimal element of C (Theorem 16 and Corollary 17).

▶ **Theorem 16.** *Consider a* $\sim_{\mathfrak{C}}$*-equivalence class* C*, and let* $(\mathcal{A}, \leq)$ *and* $(\mathcal{A}', \leq')$ *be two minimal elements of* C*. Then, there exists an isomorphism from* $\mathcal{A}$ *to* $\mathcal{A}'$*.*

▶ **Corollary 17.** *Consider a* $\sim_{\mathfrak{C}}$*-equivalence class* C*, let* $(\mathcal{A}, \leq)$ *be an element of* C*, and let* $(\mathcal{A}_*, \leq_*)$ *be the Wheeler NFA of Lemma 14. Then,* $(\mathcal{A}_*, \leq_*)$ *is a minimal element of* C*.*

We now show that every Wheeler language admits a unique minimal Wheeler DFA (up to isomorphism). This result was already proved in [4] using a different method; here, we use Wheeler bisimulations. In addition, we will show that *all* Wheeler DFAs $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$ must be in the same $\sim_{\mathfrak{C}}$-equivalence class $T_{\mathcal{L}}$, and the minimal element of $T_{\mathcal{L}}$ (unique up to isomorphism by Theorem 16) is the minimal Wheeler DFA. Note that a similar result holds for standard bisimulations: all DFAs recognizing a given regular language must be in the same bisimulation class, and the miminal element of this bisimulation class is the minimal DFA of the language [39].

To prove our results, we will use the following lemma. Intuitively, *in the deterministic case*, equivalent Wheeler automata are always related through a Wheeler bisimulation.

▶ **Lemma 18.** *Let $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ be Wheeler DFAs, with $\mathcal{A} = (Q, E, s, F)$ and $\mathcal{A}' = (Q', E', s', F')$. If $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$, there exists a Wheeler bisimulation $R \subseteq Q \times Q'$ from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$.*

Proof. Let $R \subseteq Q \times Q'$ be the relation such that, for every $u \in Q$ and for every $u' \in Q'$, we have $u \; R \; u'$ if and only if $I_u^{\mathcal{A}} \cap I_{u'}^{\mathcal{A}'} \neq \emptyset$. Let us prove that R is a Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$.

- Property 1 is true because R is a bisimulation from $\mathcal{A}$ to $\mathcal{A}'$ (see [39]).
- Let us prove Property 2. Let $C \subseteq Q$ be a $\leq$-convex set. We must prove that $R(C)$ is $\leq'$-convex. Fix $u', v', w' \in Q'$ such that $u' \leq' v' \leq w'$ and $u', w' \in R(C)$. We must prove that $v' \in R(C)$. If $(v' = u') \vee (v' = w')$, the conclusion is immediate, so we can assume $u' <' v' <' w'$.
  Since $u', w' \in R(C)$, there exist $u, w \in C$ such that $u \; R \; u'$ and $w \; R \; w'$. This means that there exist $\alpha \in I_u^{\mathcal{A}} \cap I_{u'}^{\mathcal{A}'}$ and $\gamma \in I_w^{\mathcal{A}} \cap I_{w'}^{\mathcal{A}'}$. We know that $I_{v'}^{\mathcal{A}'} \neq \emptyset$, so we can pick $\beta \in I_{v'}^{\mathcal{A}'}$. In particular, $\beta \in \mathrm{Pref}(\mathcal{L}(\mathcal{A}'))$, so $\beta \in \mathrm{Pref}(\mathcal{L}(\mathcal{A}))$ (we have $\mathrm{Pref}(\mathcal{L}(\mathcal{A})) = \mathrm{Pref}(\mathcal{L}(\mathcal{A}'))$ because $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$), and we conclude that there exists $v \in Q$ such that $\beta \in I_v^{\mathcal{A}}$.
  Let us prove that $\alpha \prec \beta \prec \gamma$. We only prove that $\alpha \prec \beta$ because one can prove analogously that $\beta \prec \gamma$. Since $\mathcal{A}'$ is a DFA and $u' \neq v'$, we have $I_{u'}^{\mathcal{A}'} \cap I_{v'}^{\mathcal{A}'} = \emptyset$. We have $u' <' v'$, $\alpha \in I_{u'}^{\mathcal{A}'}$ and $\beta \in I_{v'}^{\mathcal{A}'}$, so by Lemma 5 we obtain $\alpha \prec \beta$.
  Let us prove that $u \leq v \leq w$. We only prove that $u \leq v$ because one can prove analogously that $v \leq w$. Assume that $u \neq v$. We need to prove that $u < v$. Since $\mathcal{A}$ is a DFA and $u \neq v$, we have $I_u^{\mathcal{A}} \cap I_v^{\mathcal{A}} = \emptyset$. We have $\alpha \prec \beta$, $\alpha \in I_u^{\mathcal{A}}$ and $\beta \in I_v^{\mathcal{A}}$, so by Lemma 5 we obtain $u < v$.
  We know that C is $\leq$-convex, so from $u \leq v \leq w$ and $u, w \in C$ we conclude $v \in C$. We have $v \; R \; v'$ because $\beta \in I_v^{\mathcal{A}} \cap I_{v'}^{\mathcal{A}'}$, so $v' \in R(C)$.
- Let us prove Property 3. Let $C' \subseteq Q'$ be a $\leq'$-convex set. We must prove that $R^{-1}(C')$ is $\leq$-convex. The proof is analogous to the previous point. ◀

Fix a Wheeler language $\mathcal{L} \subseteq \Sigma^*$. We define $D_{\mathcal{L}}$ and $T_{\mathcal{L}}$ as follows.

- Let $D_{\mathcal{L}}$ be the class of all Wheeler DFAs $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.
- By Lemma 4, there exists some Wheeler DFA $(\mathcal{A}, \leq)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Let $T_{\mathcal{L}}$ be the $\sim_{\mathfrak{C}}$-equivalence class to which $(\mathcal{A}, \leq)$ belongs. By Lemma 18, the definition of $T_{\mathcal{L}}$ does not depend on the choice of $(\mathcal{A}, \leq)$. This means that, for every Wheeler DFA $(\mathcal{A}', \leq')$, if $\mathcal{L}(\mathcal{A}') = \mathcal{L}$, then $(\mathcal{A}', \leq')$ belongs to $T_{\mathcal{L}}$. Notice that this is consistent with Example 10 because the Wheeler NFAs $\mathcal{A}^{(k)}$'s used in Example 10 are not Wheeler DFAs.

In particular, we have $\emptyset \subsetneqq D_{\mathcal{L}} \subseteq T_{\mathcal{L}}$.

The key result is that $D_{\mathcal{L}}$ and $T_{\mathcal{L}}$ have the same minimal elements, as shown in the next lemma.

▶ **Lemma 19.** *Let $\mathcal{L} \subseteq \Sigma^*$ be a Wheeler language, and let $(\mathcal{A}, \leq)$ be a Wheeler NFA. Then, $(\mathcal{A}, \leq)$ is a minimal element of $D_{\mathcal{L}}$ if and only if it is a minimal element of $T_{\mathcal{L}}$. In particular, every minimal element of $T_{\mathcal{L}}$ is a Wheeler DFA.*

We can now prove that every Wheeler language admits a unique minimal Wheeler DFA (up to isomorphism).

▶ **Theorem 20.** *Let $\mathcal{L} \subseteq \Sigma^*$ be a Wheeler language, and let $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ be two minimal elements of $D_\mathcal{L}$. Then, there exists an isomorphism from $\mathcal{A}$ to $\mathcal{A}'$.*

Proof. By Lemma 19, $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ are two minimal elements of $T_\mathcal{L}$. By Theorem 16, there exists an isomorphism from $\mathcal{A}$ to $\mathcal{A}'$. ◀

Lastly, the minimal Wheeler DFA of a Wheeler language can be obtained by quotienting any Wheeler DFA that recognizes the same language.

▶ **Corollary 21.** *Let $\mathcal{L} \subseteq \Sigma^*$ be a Wheeler language, let $(\mathcal{A}, \leq)$ be a Wheeler DFA such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$, and let $(\mathcal{A}_*, \leq_*)$ be the Wheeler automaton built in Lemma 14. Then, $(\mathcal{A}_*, \leq_*)$ is a minimal element of $D_\mathcal{L}$.*

Proof. By Corollary 17, $(\mathcal{A}_*, \leq_*)$ is a minimal element of $T_\mathcal{L}$, so by Lemma 19 $(\mathcal{A}_*, \leq_*)$ is a minimal element of $D_\mathcal{L}$. ◀

6 Algorithms

Consider a Wheeler NFA $(\mathcal{A}, \leq)$, with $\mathcal{A} = (Q, E, s, F)$, and assume that the edge labels can be sorted in $O(|E|)$ time. The main result of this section is that the quotient Wheeler NFA can be built in linear time.

▶ **Theorem 22.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. We can build the Wheeler NFA $(\mathcal{A}_*, \leq_*)$ of Lemma 14 in $O(|E|)$ time.*

In the special case where $\mathcal{A}$ is a DFA, Theorem 22 states that we can compute the minimal Wheeler DFA in linear time (see Section 5), as already proved in [3]. However, in the non-deterministic setting, the algorithm and the correctness proof become much more involved. At the same time, our algorithm only requires a queue to achieve linear time, while the algorithm in [3] relies on auxiliary data structures.

Let us discuss the main ideas required to prove Theorem 22. Fix a Wheeler NFA $(\mathcal{A}, \leq)$, with $\mathcal{A} = (Q, E, s, F)$, and write $Q = \{u_1, u_2, \dots, u_{|Q|}\}$, where $u_1 < u_2 < \dots < u_{|Q|}$ (in particular, $s = u_1$ by Axiom 1). Recall that the set of all states of the quotient Wheeler NFA is the set of all $\equiv_{\mathcal{A},\leq}$-equivalence classes (see Lemma 14). To prove Theorem 22, the most difficult part is to efficiently compute a suitable representation of $\equiv_{\mathcal{A},\leq}$.

A *bit array* is an array whose entries are equal to zero or one. Let $B_{\mathcal{A},\leq}[2, |Q|]$ be the bit array such that for every $2 \leq i \leq |Q|$ we have $B_{\mathcal{A},\leq}[i] = 1$ if and only if $u_{i-1} \not\equiv_{\mathcal{A},\leq} u_i$. The idea is that $B_{\mathcal{A},\leq}[2, |Q|]$ stores all the information required to compute $\equiv_{\mathcal{A},\leq}$ because $\equiv_{\mathcal{A},\leq}$ is $\leq$-convex by Theorem 12. More precisely, we have the following result.

▶ **Lemma 23.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Then, for every $1 \leq i, i' \leq |Q|$, we have $u_i \equiv_{\mathcal{A},\leq} u_{i'}$ if and only if for every $\min\{i, i'\} + 1 \leq k \leq \max\{i, i'\}$ we have $B_{\mathcal{A},\leq}[k] = 0$.*

Proof. Recall that $\equiv_{\mathcal{A},\leq}$ is a $\leq$-convex equivalence relation on Q by Theorem 12. Fix $1 \leq i, i' \leq |Q|$. We must prove that $u_i \equiv_{\mathcal{A},\leq} u_{i'}$ if and only if for every $\min\{i, i'\} + 1 \leq k \leq \max\{i, i'\}$ we have $B_{\mathcal{A},\leq}[k] = 0$.

($\Leftarrow$) For every $\min\{i, i'\} + 1 \leq k \leq \max\{i, i'\}$, we have $B_{\mathcal{A},\leq}[k] = 0$, so $u_{k-1} \equiv_{\mathcal{A},\leq} u_k$. Since $\equiv_{\mathcal{A},\leq}$ is transitive, we have $u_{\min\{i,i'\}} \equiv_{\mathcal{A},\leq} u_{\max\{i,i'\}}$. We know that $\equiv_{\mathcal{A},\leq}$ is symmetric, so $u_i \equiv_{\mathcal{A},\leq} u_{i'}$.

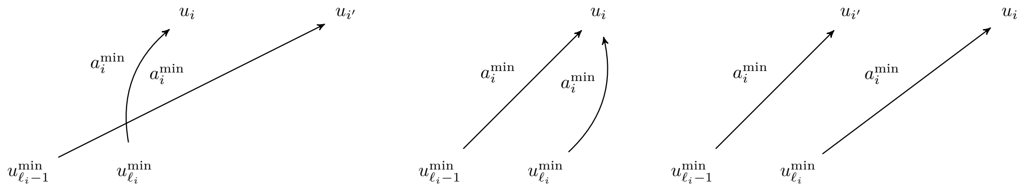

Figure 6 The three cases considered in the proof of Lemma 24. *Left:* the case $i < i'$ would imply that equally labeled edges cross (see Figure 1). *Center:* the case $i' = i$ would contradict the minimality of $\ell_i^{\min}$. *Right:* the case $i' < i$ would imply $B_{\mathcal{A},\le}[i] \neq 1$ because $\equiv_{\mathcal{A},\le}$ is $\le$-convex.

($\Rightarrow$) Since $u_i \equiv_{\mathcal{A},\le} u_{i'}$ and $\equiv_{\mathcal{A},\le}$ is symmetric, we obtain $u_{\min\{i,i'\}} \equiv_{\mathcal{A},\le} u_{\max\{i,i'\}}$. For every $\min\{i,i'\}+1 \le k \le \max\{i,i'\}$, we have $u_{\min\{i,i'\}} \equiv_{\mathcal{A},\le} u_k$ because $\min\{i,i'\}+1 \le k \le \max\{i,i'\}$ and $\equiv_{\mathcal{A},\le}$ is $\le$-convex by Theorem 12. We know that $\equiv_{\mathcal{A},\le}$ is an equivalence relation on Q, so $u_{k-1} \equiv_{\mathcal{A},\le} u_k$ for every $\min\{i,i'\}+1 \le k \le \max\{i,i'\}$, and we conclude $B_{\mathcal{A},\le}[k] = 0$ for every $\min\{i,i'\}+1 \le k \le \max\{i,i'\}$. ◀

Lemma 23 suggests that computing $B_{\mathcal{A},\le}[2,|Q|]$ is the key step required to prove Theorem 22. To this end, let us define $a_i^{\min}$, $\ell_i^{\min}$, $a_i^{\max}$, $\ell_i^{\max}$ and $Out(i)$ for every $1 \le i \le |Q|$.

- For $2 \le i \le |Q|$, let $a_i^{\min}$ be the smallest character labeling some edge reaching u_i. In other words, let $a_i^{\min}$ be the smallest $c \in \Sigma$ (with respect to the total order $\preceq$) such that $(u_j, u_i, c) \in E$ for some $1 \le j \le |Q|$. Note that $a_i^{\min}$ is well defined because we know that u_i must have at least one incoming edge (because u_i is reachable from the initial state s). If u_1 has at least one incoming edge, define $a_1^{\min}$ in the same way; otherwise, let $a_1^{\min} = \perp$.
- For $2 \le i \le |Q|$, let $\ell_i^{\min}$ be the smallest $1 \le j \le |Q|$ such that $(u_j, u_i, a_i^{\min}) \in E$. If $a_1^{\min} \neq \perp$, define $\ell_1^{\min}$ in the same way; if $a_1^{\min} = \perp$, define $\ell_1^{\min} = \perp$.
- For $1 \le i \le |Q|$, define $a_i^{\max}$ and $\ell_i^{\max}$ in the same way as $a_i^{\min}$and $\ell_i^{\min}$, but replacing "smallest" with "largest".
- For $1 \le i \le |Q|$, let $Out(i)$ be the set of all characters labeling some edge leaving u_i. In other words, let $Out(i)$ be the set of all $a \in \Sigma$ for which there exists $1 \le j \le |Q|$ such that $(u_i, u_j, a) \in E$.

Let us discuss some situations in which we have $B_{\mathcal{A},\le}[i] = 1$.

▶ **Lemma 24.** *Let $(\mathcal{A}, \le)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Fix $2 \le i \le |Q|$ (in particular, $\ell_i^{\min} \neq \perp$).*

1. *If $Out(i-1) \neq Out(i)$, then $B_{\mathcal{A},\le}[i] = 1$.*
2. *If $(u_{i-1} \in F \wedge u_i \notin F) \vee (u_{i-1} \notin F \wedge u_i \in F)$, then $B_{\mathcal{A},\le}[i] = 1$.*
3. *Assume that $B_{\mathcal{A},\le}[i] = 1$. If $\ell_i^{\min} \ge 2$, then $B_{\mathcal{A},\le}[\ell_i^{\min}] = 1$, and if $(\ell_{i-1}^{\max} \neq \perp) \wedge (\ell_{i-1}^{\max} \le |Q|-1)$, then $B_{\mathcal{A},\le}[\ell_{i-1}^{\max}+1] = 1$.*

Proof (Sketch). The first two claims follow immediately because $\equiv_{\mathcal{A},\le}$ is a bisimulation from $\mathcal{A}$ to $\mathcal{A}$. We only focus on the first part of the third claim.

Assume that $B_{\mathcal{A},\le}[i] = 1$ and $\ell_i^{\min} \ge 2$. We know that $\ell_i^{\min} \neq \perp$ (which implies $a_i^{\min} \neq \perp$), and $B_{\mathcal{A},\le}[\ell_i^{\min}]$ is well defined because $\ell_i^{\min} \ge 2$. Suppose for the sake of a contradiction that $B_{\mathcal{A},\le}[\ell_i^{\min}] \neq 1$. This means that $u_{\ell_i^{\min}-1} \equiv_{\mathcal{A},\le} u_{\ell_i^{\min}}$. Since $\equiv_{\mathcal{A},\le}$ is a bisimulation from $\mathcal{A}$ to $\mathcal{A}$ and $(u_{\ell_i^{\min}}, u_i, a_i^{\min}) \in E$, there exists $1 \le i' \le |Q|$ such that $u_{i'} \equiv_{\mathcal{A},\le} u_i$ and $(u_{\ell_i^{\min}-1}, u_{i'}, a_i^{\min}) \in E$. We distinguish three cases, and we show that of all them lead to a contradiction (see Figure 6).

- Case 1: $i < i'$. From $(u_{\ell_i^{\min}}, u_i, a_i^{\min}), (u_{\ell_i^{\min}-1}, u_{i'}, a_i^{\min}) \in E$ and $i < i'$ we conclude $\ell_i^{\min} \leq \ell_i^{\min} - 1$ by Axiom 3, which is a contradiction.
- Case 2: $i' = i$. We have $(u_{\ell_i^{\min}-1}, u_i, a_i^{\min}) \in E$, so by the minimality of $\ell_i^{\min}$ we conclude $\ell_i^{\min} \leq \ell_i^{\min} - 1$, which is a contradiction.
- Case 3: $i' < i$. We know that $\equiv_{\mathcal{A},\leq}$ is $\leq$-convex (by Theorem 12), $i' \leq i-1 < i$ and $u_{i'} \equiv_{\mathcal{A},\leq} u_i$, so we obtain that $u_{i-1} \equiv_{\mathcal{A},\leq} u_i$ and we conclude that $B_{\mathcal{A},\leq}[i] \neq 1$, which is a contradiction. ◀

We are now ready to compute $B_{\mathcal{A},\leq}[2, |Q|]$. Informally, we use Algorithm 1 to implement the properties stated in Lemma 24. At the beginning of the algorithm, we initialize $B[2, |Q|]$ to zero, and at the end of the algorithm, we will have $B[2, |Q|] = B_{\mathcal{A},\leq}[2, |Q|]$. We compute the bit array $Z[2, |Q|]$ such that, for every $2 \leq i \leq |Q|$, we have $Z[i] = 1$ if and only if $Out(i-1) \neq Out(i)$. Then, Lines 8-13 implement the first and the second properties of Lemma 24, and Lines 14-24 propagate the third property of Lemma 24.

Algorithm 1 The input is a Wheeler NFA $(\mathcal{A}, \leq)$, with $\mathcal{A} = (Q, E, s, F)$. The output is $B_{\mathcal{A},\leq}[2, |Q|]$. The algorithm uses a queue K and two arrays $Z[2, |Q|]$ and $B[2, |Q|]$.

1: **for** $i \leftarrow 1, 2, \ldots, |Q|$ **do** ▷ Initialization
2:     Compute $\ell_i^{\min}$ and $\ell_i^{\max}$.
3: **end for**
4: compute $Z[2, |Q|]$ ▷ $Z[i] = 1$ if $Out(i-1) \neq Out(i)$ and $Z[i] = 0$ if $Out(i-1) = Out(i)$
5: initialize $B[2, |Q|]$ to zero
6: initialize an empty queue K
7:
8: **for** $i \leftarrow 2, 3, \ldots, |Q|$ **do** ▷ Main part
9:     **if** $(u_{i-1} \in F \wedge u_i \notin F) \vee (u_{i-1} \notin F \wedge u_i \in F) \vee Z[i] = 1$ **then**
10:         $Enqueue(K, i)$
11:         $B[i] \leftarrow 1$
12:     **end if**
13: **end for**
14: **while** K is nonempty **do**
15:     $i \leftarrow Dequeue(K)$
16:     **if** $(\ell_i^{\min} \geq 2) \wedge (B[\ell_i^{\min}] = 0)$ **then**
17:         $Enqueue(K, \ell_i^{\min})$
18:         $B[\ell_i^{\min}] \leftarrow 1$
19:     **end if**
20:     **if** $(\ell_{i-1}^{\max} \neq \perp) \wedge (\ell_{i-1}^{\max} \leq |Q| - 1) \wedge (B[\ell_{i-1}^{\max} + 1] = 0)$ **then**
21:         $Enqueue(K, \ell_{i-1}^{\max} + 1)$
22:         $B[\ell_{i-1}^{\max} + 1] \leftarrow 1$
23:     **end if**
24: **end while**
25: **return** $B[2, |Q|]$

Let us show that Algorithm 1 correctly computes $B_{\mathcal{A},\leq}[2, |Q|]$.

▶ **Lemma 25.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. On input $(\mathcal{A}, \leq)$, Algorithm 1 returns the array $B_{\mathcal{A},\leq}[2, |Q|]$.*

Proof (Sketch). Fix $2 \leq i \leq |Q|$. We must prove that at the end of the algorithm we have $B[i] = B_{\mathcal{A},\leq}[i]$. If $B[i] = 1$, then we intuitively obtain $B_{\mathcal{A},\leq}[i] = 1$ by Lemma 24. The difficult part is to show that, if $B[i] = 0$, then $B_{\mathcal{A},\leq}[i] = 0$. In other words, we need to show that the properties stated in Lemma 24 are sufficient to detect whether $B[i] = 1$. Intuitively, we proceed as follows. Let $\sim$ be the relation from Q to Q such that for every $1 \leq i, i' \leq |Q|$

we have $u_i \sim u_{i'}$ if and only if for every $\min\{i, i'\} + 1 \leq k \leq \max\{i, i'\}$ we have $B[k] = 0$ at the end of the algorithm. In particular, by the definition of $\sim$, we have $u_{i-1} \sim u_i$ for every $2 \leq i \leq |Q|$ such that $B[i] = 0$ at the end of the algorithm. We only need to prove $\sim$ is a Wheeler autobisimulation on $(\mathcal{A}, \leq)$, because then by the maximality of $\equiv_{\mathcal{A},\leq}$ (see Theorem 12) we have $u_{i-1} \equiv_{\mathcal{A},\leq} u_i$ for every $2 \leq i \leq |Q|$ such that $B[i] = 0$ at the end of the algorithm, and so $B_{\mathcal{A},\leq}[i] = 0$ for every $2 \leq i \leq |Q|$ such that $B[i] = 0$ at the end of the algorithm. We prove that $\sim$ is a Wheeler autobisimulation on $(\mathcal{A}, \leq)$ in the full version. ◀

Let discuss the running time of Algorithm 1 (Lemma 27). We start with a remark.

▶ Remark 26. Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Recall that every $u \in Q$ is reachable in $\mathcal{A}$, so every $u \in Q \setminus \{s\}$ must have at least one incoming edge, which implies $|E| \geq |Q| - 1$. In particular, if an algorithm has time complexity $O(|Q| + |E|)$, we can simply write that it has time complexity $O(|E|)$.

▶ **Lemma 27.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Algorithm 1 can be implemented to run in $O(|E|)$ time on input $(\mathcal{A}, \leq)$.*

Proof (Sketch). The initialization (Lines 1-6) can be implemented in linear time because the edge labels can be sorted in linear time. The main part of the algorithm (Lines 8-25) is executed in linear time because every $2 \leq i \leq |Q|$ is added to the queue K at most once, so we only need $O(1)$ time per state. ◀

Theorem 22 implies that in linear time we can check whether two Wheeler NFAs are in the same $\sim_{\mathfrak{C}}$-equivalence class.

▶ **Corollary 28.** *Let $(\mathcal{A}, \leq)$ and $(\mathcal{A}', \leq')$ be Wheeler NFAs, with $\mathcal{A} = (Q, E, s, F)$ and $\mathcal{A}' = (Q', E', s', F')$. We can determine whether there exists a Wheeler bisimulation from $(\mathcal{A}, \leq)$ to $(\mathcal{A}', \leq')$ in $O(|E| + |E|')$ time.*

Proof (Sketch). By Theorem 22, in linear time we can build the quotient Wheeler NFAs of $(A, \leq)$ and $(\mathcal{A}', \leq')$. Then, Lemma 15 suggests that there exists a Wheeler bisimulation from $(A, \leq)$ to $(\mathcal{A}', \leq')$ if and only if the two quotient Wheeler NFAs are isomorphic. We can check whether the two quotient Wheeler NFAs are isomorphic in linear time because (i) if the two quotient Wheeler NFAs do not have the same number of states, they cannot be isomorphic, and (ii) if the two quotient Wheeler NFAs have the same number of states, then by Lemma 15 they are isomorphic if and only if the (unique) bijection that respects the Wheeler orders of the quotient Wheeler NFAs is an isomorphism. ◀

Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. Theorem 22 states that we can build the quotient Wheeler NFA in linear time, under the assumption that the edge labels can be sorted in linear time. It is natural to wonder if it is possible to achieve linear time even when the edge labels cannot be sorted in linear time. In principle, this does not seem unreasonable: we know the Wheeler order $\leq$, and for every $u, v \in Q$, if $u < v$, then by Axiom 2 all characters labeling some edge reaching u must be smaller than all characters labeling some edge reaching v. In other words, the Wheeler order $\leq$ *inherently* contains information on the mutual order of the edge labels. However, from the Wheeler order we cannot retrieve any information on the mutual order of the characters labeling some edge reaching the *same* state, and we will see that this prevents us from achieving linear time.

More concretely, we consider the *comparison model*, in which the only query allowed to compare the elements of the alphabet Σ is the following: given $a_1, a_2 \in \Sigma$, decide whether $a_1 \preceq a_2$. The model assumes that each such query can be solved in $O(1)$ time. We can show that, in the comparison model, building the quotient NFA requires $\Omega(|E| \log |E|)$ time.

▶ **Theorem 29.** *Let $(\mathcal{A}, \leq)$ be a Wheeler NFA, with $\mathcal{A} = (Q, E, s, F)$. In the comparison model, the time complexity of building the Wheeler NFA $(\mathcal{A}_*, \leq_*)$ of Lemma 14 is $\Theta(|E| \log |E|)$. This is true even if we know that $\mathcal{A}$ is a DFA.*

Theorem 29 is consistent with Theorem 22 because it is well known that the complexity of sorting n elements in the comparison model is not linear but $\Omega(n \log n)$ (see, e.g., [46, 22, 28]).

7 Conclusions and Future Work

In this paper, we have introduced Wheeler bisimulations. We have shown that they induce a unique minimal Wheeler NFA for every Wheeler language, which can be built in linear time.

Definition 7 refers to the most popular notion of bisimulation, often called *strong* bisimulation. However, many variants appear in the literature. The properties listed in Definition 7 ensure that a bisimulation is propagated when following edges in a *forward* fashion, but it is possible to require the same compatibility to hold when edges are followed in a *backward* fashion (or both in a forward fashion and a backward fashion). In addition, several notions of *weak* bisimulation have been proposed (such as stutter bisimulation and normed bisimulation), which typically still induce a unique minimal NFA. Usually, every (strong) bisimulation is a weak bisimulation, so weak bisimulations yield smaller minimal NFAs. For details, we refer the reader to [19, 7]. Moreover, as we have seen in the introduction, bisimulations appear in different domains, so there has been much effort to provide a unified view of bisimulation [56], also via algebraic and co-algebraic approaches [14, 15, 44, 57, 50, 1]. Consequently, our theory of Wheeler bisimulations can be extended in two directions: (i) by modifying the definition of Wheeler bisimulation, and (ii) by switching from Wheeler automata to other formalisms (which should probably capture similar ideas of convexity, see [35] for Petri nets).

The definitions of bisimulation and Wheeler bisimulation are symmetrical: each property must hold even when the roles of the two automata are exchanged. This is not a mandatory requirement: when this symmetry is broken, one works with *simulations* (and not bisimulations), see [7]. It seems reasonable to define *Wheeler simulations* by removing Property 3 in Definition 9 (because Property 2 and Property 3 are symmetrical).

One important point that needs further clarification is the relationship between Wheeler bisimulations and logic. It is well known that the largest bisimulation on an NFA can be characterized through several logics, notably CTL and CTL* ([20], see also [7]). In general, the relationship between Wheeler languages and logic is not well understood. We have seen in the introduction that Wheeler languages enjoy the typical properties of the class of regular languages: determinism and non-determinism have the same expressive power, there exists a unique minimal Wheeler DFA, and Wheeler languages admit a Myhill-Nerode theorem. Nonetheless, no logical characterization of Wheeler languages is known, while regular languages can be characterized via the monadic second-order logic [61]. A characterization of Wheeler languages in terms of syntactic monoids is also missing.

Consider the class of all regular languages. Bisimulations lead to the most interesting notion of uniqueness (up to isomorphism) for NFAs, but other formalisms achieve the same goal, such as residual automata [38], the universal automaton [48], and the átomaton [17] (see [49] for more information and more options). The natural question is whether we can also extend these results to Wheeler automata. We conjecture that this is possible at least for residual automata. Indeed, residual automata are closely related to the Myhill-Nerode theorem, and we know that there exists a Myhill-Nerode theorem for Wheeler automata [5].

References

1 Samson Abramsky. A domain equation for bisimulation. *Information and Computation*, 92(2):161–218, 1991. doi:10.1006/INCO.1991.9999.

2 Luca Aceto, Anna Ingólfsdóttir, Kim Guldstrand Larsen, and Jiri Srba. *Reactive systems: modelling, specification and verification*. Cambridge University Press, 2007.

3 Jarno Alanko, Nicola Cotumaccio, and Nicola Prezza. Linear-time minimization of Wheeler DFAs. In *2022 Data Compression Conference (DCC)*, pages 53–62. IEEE, 2022. doi:10.1109/DCC52660.2022.00013.

4 Jarno Alanko, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Regular languages meet prefix sorting. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 911–930. SIAM, 2020. doi:10.1137/1.9781611975994.55.

5 Jarno Alanko, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Wheeler languages. *Information and Computation*, 281:104820, 2021. doi:10.1016/J.IC.2021.104820.

6 Jarno N Alanko, Davide Cenzato, Nicola Cotumaccio, Sung-Hwan Kim, Giovanni Manzini, and Nicola Prezza. Computing the LCP array of a labeled graph. In *35th Annual Symposium on Combinatorial Pattern Matching*, 2024.

7 Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

8 Ruben Becker, Manuel Cáceres, Davide Cenzato, Sung-Hwan Kim, Bojana Kodric, Francisco Olivares, and Nicola Prezza. Sorting Finite Automata via Partition Refinement. In *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ESA.2023.15.

9 Ruben Becker, Giuseppa Castiglione, Giovanna D'Agostino, Alberto Policriti, Nicola Prezza, Antonio Restivo, and Brian Riccardi. Universally Wheeler languages. In *International Conference on Developments in Language Theory*, pages 45–60. Springer, 2025. doi:10.1007/978-3-032-01475-7_4.

10 Ruben Becker, Davide Cenzato, Sung-Hwan Kim, Bojana Kodric, Alberto Policriti, and Nicola Prezza. Optimal Wheeler language recognition. In *International Symposium on String Processing and Information Retrieval*, pages 62–74. Springer, 2023. doi:10.1007/978-3-031-43980-3_6.

11 Ruben Becker, Nicola Cotumaccio, Sung-Hwan Kim, Nicola Prezza, and Carlo Tosoni. Encoding co-lex orders of finite-state automata in linear space. In *36th Annual Symposium on Combinatorial Pattern Matching*, page 17, 2025.

12 Ruben Becker, Sung-Hwan Kim, Nicola Prezza, and Carlo Tosoni. Indexing finite-state automata using forward-stable partitions. In *International Symposium on String Processing and Information Retrieval*, pages 26–40. Springer, 2024. doi:10.1007/978-3-031-72200-4_3.

13 Eike Best, Raymond Devillers, Astrid Kiehn, and Lucia Pomello. Concurrent bisimulations in Petri nets. *Acta Informatica*, 28(3):231–264, 1991. doi:10.1007/BF01178506.

14 Nick Bezhanishvili, Marcello M Bonsangue, Helle Hvid Hansen, Dexter Kozen, Clemens Kupke, Prakash Panangaden, and Alexandra Silva. Minimisation in logical form. In *Samson Abramsky on Logic and Structure in Computer Science and Beyond*, pages 89–127. Springer, 2023.

15 Filippo Bonchi, Marcello M Bonsangue, Helle H Hansen, Prakash Panangaden, Jan JMM Rutten, and Alexandra Silva. Algebra-coalgebra duality in Brzozowski's minimization algorithm. *ACM Transactions on Computational Logic (TOCL)*, 15(1):1–29, 2014. doi:10.1145/2490818.

16 Alexander Bowe, Taku Onodera, Kunihiko Sadakane, and Tetsuo Shibuya. Succinct de Bruijn graphs. In *International workshop on algorithms in bioinformatics*, pages 225–235. Springer, 2012. doi:10.1007/978-3-642-33122-0_18.

17 Janusz A. Brzozowski and Hellis Tamm. Theory of átomata. *Theoretical Computer Science*, 539:13–27, 2014. doi:10.1016/J.TCS.2014.04.016.

18 Michael Burrows and David J. Wheeler. A block-sorting lossless data compression algorithm, 1994. Technical report.

19 Miroslav Ćirić, Jelena Ignjatović, Milan Bašić, and Ivana Jančić. Nondeterministic automata: equivalence, bisimulations, and uniform relations. *Information Sciences*, 261:185–218, 2014. `doi:10.1016/J.INS.2013.07.029`.

20 Edmund M Clarke and E Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on logic of programs*, pages 52–71. Springer, 1981. `doi:10.1007/BFB0025774`.

21 Alessio Conte, Nicola Cotumaccio, Travis Gagie, Giovanni Manzini, Nicola Prezza, and Marinella Sciortino. Computing matching statistics on Wheeler DFAs. In *2023 Data Compression Conference (DCC)*, pages 150–159. IEEE, 2023. `doi:10.1109/DCC55655.2023.00023`.

22 Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

23 Nicola Cotumaccio. Graphs can be succinctly indexed for pattern matching in $O(|E|^2+|V|^{5/2})$ time. In *Data Compression Conference, DCC 2022, Snowbird, UT, USA, March 22-25, 2022*, pages 272–281. IEEE, 2022.

24 Nicola Cotumaccio. Prefix Sorting DFAs: A Recursive Algorithm. In *34th International Symposium on Algorithms and Computation (ISAAC 2023)*, volume 283 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.ISAAC.2023.22`.

25 Nicola Cotumaccio. A Myhill-Nerode Theorem for Generalized Automata, with Applications to Pattern Matching and Compression. In *41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*, volume 289 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. `doi:10.4230/LIPIcs.STACS.2024.26`.

26 Nicola Cotumaccio. Fast pattern matching with epsilon transitions. In *International Workshop on Combinatorial Algorithms*, pages 242–255. Springer, 2025. `doi:10.1007/978-3-031-98740-3_18`.

27 Nicola Cotumaccio. Improved circular dictionary matching. In *36th Annual Symposium on Combinatorial Pattern Matching*, 2025.

28 Nicola Cotumaccio. On Inverting the Burrows-Wheeler Transform. In *From Strings to Graphs, and Back Again: A Festschrift for Roberto Grossi's 60th Birthday*, volume 132 of *Open Access Series in Informatics (OASIcs)*, pages 17:1–17:8. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/OASIcs.Grossi.17`.

29 Nicola Cotumaccio. Sorting circular suffixes in linear time. In *International Conference on Developments in Language Theory*, pages 294–310. Springer, 2025. `doi:10.1007/978-3-032-01475-7_20`.

30 Nicola Cotumaccio. Generalized Wheeler automata: minimality. In *International Conference on Developments in Language Theory*, pages 88–102. Springer, 2026. `doi:10.1007/978-3-032-28404-4_7`.

31 Nicola Cotumaccio, Giovanna D'Agostino, Daniel Gibney, Alberto Policriti, Nicola Prezza, and Sharma V. Thankachan. Wheeler Graphs and Wheeler Languages. In *The Expanding World of Compressed Data: A Festschrift for Giovanni Manzini's 60th Birthday*, volume 131 of *Open Access Series in Informatics (OASIcs)*, pages 12:1–12:28. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/OASIcs.Manzini.12`.

32 Nicola Cotumaccio, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Co-lexicographically ordering automata and regular languages-part i. *Journal of the ACM*, 70(4):1–73, 2023. `doi:10.1145/3607471`.

33 Nicola Cotumaccio, Travis Gagie, Dominik Köppl, and Nicola Prezza. Space-time trade-offs for the LCP array of Wheeler DFAs. In *International Symposium on String Processing and Information Retrieval*, pages 143–156. Springer, 2023. `doi:10.1007/978-3-031-43980-3_12`.

34 Nicola Cotumaccio and Nicola Prezza. On indexing and compressing finite automata. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2585–2599. SIAM, 2021. `doi:10.1137/1.9781611976465.153`.

35 Nicola Cotumaccio and Catia Trubiani. Convex Petri nets. In *Proceedings of the International Workshop on Petri Nets and Software Engineering*, pages 24–25, 2024.

36 Giovanna D'Agostino, Davide Martincigh, and Alberto Policriti. Ordering regular languages and automata: complexity. *Theoretical Computer Science*, 949:113709, 2023. doi:10.1016/J.TCS.2023.113709.

37 Pierpaolo Degano, Rocco De Nicola, and Ugo Montanari. Universal axioms for bisimulations. *Theoretical Computer Science*, 114(1):63–91, 1993. doi:10.1016/0304-3975(93)90153-K.

38 François Denis, Aurélien Lemay, and Alain Terlutte. Residual finite state automata. *Fundamenta Informaticae*, 51(4):339–368, 2002. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi51-4-02.

39 Joost Engelfriet. Determinancy $\rightarrow$ (observation equivalence $=$ trace equivalence). *Theoretical Computer Science*, 36:21–25, 1985. doi:10.1016/0304-3975(85)90028-3.

40 Massimo Equi, Veli Mäkinen, Alexandru I Tomescu, and Roberto Grossi. On the complexity of string matching for graphs. *ACM Transactions on Algorithms*, 19(3):1–25, 2023. doi:10.1145/3588334.

41 Paolo Ferragina and Giovanni Manzini. Indexing compressed text. *Journal of the ACM*, 52(4):552–581, 2005. doi:10.1145/1082036.1082039.

42 Travis Gagie, Giovanni Manzini, and Jouni Sirén. Wheeler graphs: A framework for BWT-based data structures. *Theoretical computer science*, 698:67–78, 2017. doi:10.1016/J.TCS.2017.06.016.

43 John Hopcroft. An $n \log n$ algorithm for minimizing states in a finite automaton. In *Theory of machines and computations*, pages 189–196. Elsevier, 1971.

44 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. *Journal of Computer and System Sciences*, 81(5):859–879, 2015. doi:10.1016/J.JCSS.2014.12.005.

45 Paris C Kanellakis and Scott A Smolka. CCS expressions, finite state processes, and three problems of equivalence. In *Proceedings of the second annual ACM symposium on Principles of distributed computing*, pages 228–240, 1983. doi:10.1145/800221.806724.

46 Donald E Knuth. *The art of computer programming: sorting and searching, volume 3*. Addison-Wesley Professional, 1998.

47 Dexter C Kozen. *Automata and computability*. Springer Science & Business Media, 2007.

48 Sylvain Lombardy and Jacques Sakarovitch. The universal automaton. *Logic and Automata, History and Perspectives*, 2:457–504, 2008.

49 Hendrik Maarand and Hellis Tamm. Yet another canonical nondeterministic automaton. *Theoretical Computer Science*, 1001:114581, 2024. doi:10.1016/J.TCS.2024.114581.

50 Pasquale Malacaria. Studying equivalences of transition systems with algebraic tools. *Theoretical computer science*, 139(1-2):187–205, 1995. doi:10.1016/0304-3975(94)00047-M.

51 Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993. doi:10.1137/0222058.

52 R Milner. A calculus of communicating systems. *Lecture Notes in Computer Science*, 1980.

53 Edward F Moore et al. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

54 Robert Paige and Robert E Tarjan. Three partition refinement algorithms. *SIAM Journal on computing*, 16(6):973–989, 1987. doi:10.1137/0216062.

55 David Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, pages 167–183, 1981.

56 Markus Roggenbach and Mila Majster-Cederbaum. Towards a unified view of bisimulation: a comparative study. *Theoretical Computer Science*, 238(1-2):81–130, 2000. doi:10.1016/S0304-3975(99)00303-5.

57 Jurriaan Rot. Coalgebraic minimization of automata by initiality and finality. *Electronic Notes in Theoretical Computer Science*, 325:253–276, 2016. doi:10.1016/J.ENTCS.2016.09.042.

58 Davide Sangiorgi. *Introduction to bisimulation and coinduction*. Cambridge University Press, 2011.

59 Huei Jang Shyr and Gabriel Thierrin. Ordered automata and associated languages. *Tamkang J. Math*, 5(1):9–20, 1974.

60 Larry J Stockmeyer and Albert R Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 1–9, 1973. doi:10.1145/800125.804029.

61 Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Springer Science & Business Media, 2012.

62 Antti Valmari. Simple bisimilarity minimization in $O(m \log n)$ time. *Fundamenta Informaticae*, 105(3):319–339, 2010. doi:10.3233/FI-2010-369.

63 Antti Valmari and Petri Lehtinen. Efficient Minimization of DFAs with Partial Transition. In *25th International Symposium on Theoretical Aspects of Computer Science*, volume 1 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 645–656. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008. doi:10.4230/LIPIcs.STACS.2008.1328.

64 Rob Van Glabbeek and Frits Vaandrager. Petri net models for algebraic theories of concurrency. In *International Conference on Parallel Architectures and Languages Europe*, pages 224–242. Springer, 1987. doi:10.1007/3-540-17945-3_13.