The *Rational* Construction of a Wheeler DFA

Giovanni Manzini
Dept. of Computer Science, University of Pisa, Italy

Alberto Policriti
Dept. of Mathematics, Computer Science and Physics, University of Udine, Italy

Nicola Prezza
Dept. of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Italy

Brian Riccardi
Dept. of Informatics, Systems and Communication, University of Milano-Bicocca, Italy

Abstract

Deterministic Finite Wheeler Automata are a natural generalisation to regular languages of the theory of compressed data structures originated by the introduction of the Burrows-Wheeler transform. Indeed, if we can find a Wheeler automaton recognizing a given language $\mathcal{L}$, such automaton can be used to design time and space efficient algorithms for representing and searching $\mathcal{L}$.

In this paper we introduce an alternative representation of Deterministic Wheeler Automata by showing that a natural map between strings and rational numbers in $\mathbb{Q}[0,1)$ can be extended to represent the automaton's states as *intervals* in $\mathbb{Q}[0,1)$. Using this representation it emerges a natural relationship between automata properties and some properties of real numbers. In addition, such representation enables us to formulate problems related to automata in a numerical setting. Although at the moment the numerical approach does not lead to time efficient algorithms, we believe this new perspective deserves further consideration.

As a further demonstration of the convenience of this new representation, we use it to provide a simple proof of an unexpected result on regular languages. More precisely, we compare the size of the smallest *Wheeler* automaton recognizing a given language $\mathcal{L}$ with respect to the size of the smallest automaton, possibly non-Wheeler, recognizing the same language. We show settings in which there can be an exponential gap between the two sizes, and we discuss the implications of this result on the problem of representing regular languages.

2012 ACM Subject Classification Theory of computation → Pattern matching; Theory of computation → Formal languages and automata theory

Keywords and phrases String Matching, Deterministic Finite Automata, Wheeler languages, Graph Indexing, Co-lexicographical Sorting

Digital Object Identifier 10.4230/LIPIcs.CPM.2024.23

Funding *Giovanni Manzini*: Funded by the Italian Ministry of Health, POS 2014-2020, project ID T4-AN-07, CUP I53C22001300001, by INdAM-GNCS Project CUP E53C23001670001 and by PNRR ECS00000017 Tuscany Health Ecosystem, Spoke 6 CUP I53C22000780001.
Alberto Policriti: project funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.4 - Call for tender No. 3138 of 16 December 2021, rectified by Decree n.3175 of 18 December 2021 of Italian Ministry of University and Research funded by the European Union - NextGenerationEU. **Award Number:** project code *CN_*00000033, Concession Decree No. 1034 of 17 June 2022 adopted by the Italian Ministry of University and Research, CUP G23C22001110007, Project title "National Biodiversity Future Center – NBFC".
Nicola Prezza: Funded by the European Union (ERC, REGINDEX, 101039208). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Brian Riccardi: Received grants from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement PANGAIA No. 872539, and from MUR 2022YRB97K, PINC, Pangenome INformatiCs: from Theory to Applications.

1 Introduction

A (deterministic) automaton is a simple version of a Turing Machine, operating just moving from left to right and using the tape just for reading a pattern (no writing). It encodes a (very simple, testing) algorithm, operating either accepting or rejecting its input pattern. A *Wheeler* automaton [1, 6] is an automaton equipped with a *total* order $\prec$ on its set of states and constrained by two simple axioms that, ultimately, cast an order on the entire collection of prefixes of accepted strings. As a matter of fact, a Wheeler automaton operates generalising to a *collection* of strings the computation performed to produce the Burrows-Wheeler transform of a string – that is, a linear and invertible permutation turning a string α into a highly compressible and searchable equivalent [4]. Many important, practical byproducts become available, starting with the ability to store and search the language accepted by a Wheeler automaton in little space and time (see [5]).

A remarkable property of regular languages, not present in other settings, is that a given language can be accepted by different automata with different properties. Hence, a language accepted by a Wheeler automaton can be accepted also by a non-Wheeler automaton. For an automaton $\mathcal{A}$ we define the *width of* $\mathcal{A}$, $\text{width}(\mathcal{A})$, as the minimum *width of a partial order*[1] on $\mathcal{A}$ satisfying Wheeler axioms (details given below). Since (by definition) a Wheeler automaton $\mathcal{A}_w$ admits a total order, it is always $\text{width}(\mathcal{A}_w) = 1$. In [5] it is shown that $\text{width}(\mathcal{A})$ measures the "hardness" of representing and searching $\mathcal{A}$. For example, if $\text{width}(\mathcal{A}) = p$ the automaton can be represented in $\Theta(\log p)$ bits per transition and there exists a linear-space data structure solving regular expression matching in $O(p^2)$ time per matched character.

Let $\mathcal{L}$ be a language accepted by a minimal (in terms of number of states) Deterministic Finite Automaton (DFA) $\mathcal{D}$ as well as by a minimal *Wheeler DFA* (WDFA, see also Section 2) $\mathcal{D}_w$. Since either $\mathcal{D}$ or $\mathcal{D}_w$ can be used to represent the language $\mathcal{L}$ it is worthwhile to compare their effectiveness for this task. To this end, in this paper we consider the problem of bounding the size of $\mathcal{D}_w$ in terms of the size of $\mathcal{D}$ and of the width $\text{width}(\mathcal{D})$. We prove that even for $\text{width}(\mathcal{D}) = 2$, a minimal Wheeler automaton $\mathcal{D}_w$ can have exponentially more states than $\mathcal{D}$. This result has the immediate consequence that the Wheeler automata representation is not always the more effective: it can be algorithmically more convenient to deal with a non-Wheeler automaton with a small width rather than working with a (minimal) Wheeler automaton for the same language.

To provide a simple proof of the above result, we introduce a new general method for *representing* automaton $\mathcal{D}$ (and $\mathcal{D}_w$), proving that the co-lexicographic order of strings and the ordering of a Wheeler automaton can be conveniently presented using rational numbers and convex subsets of rational numbers in $[0, 1)$. This representation provides a different perspective on some properties of automata, highlighting their connection with established properties of real numbers. In addition, it suggests a new view for a number of problems that we illustrate and discuss, concluding by showing that some such problems can also be approached in an *arithmetic way*.

[1] The width of a partial order is the maximum length of any of its anti-chains.

2 Basics

Let $\Sigma = \{a_1, \ldots, a_\sigma\}$ denote a finite ordered alphabet of size σ. We denote by Σ^* the set of finite strings over Σ. The character ϵ denotes the empty string. We assume that the elements of Σ^* are ordered according to the co-lexicographic (co-lex) order defined as follows: given $\alpha, \beta \in \Sigma^*$, we say that α is co-lex smaller than β ($\alpha < \beta$) if and only if α is a suffix of β or there exist $\gamma, \alpha', \beta' \in \Sigma^*$ and $a, b \in \Sigma$ with $a < b$ such that $\alpha = \alpha' a \gamma$ and $\beta = \beta' b \gamma$.

A Deterministic Finite-State Automaton (DFA) $\mathcal{A} = (Q, s, \delta, F)$ consists of a finite set of states Q, an initial state $s \in Q$, a set of final states $F \subseteq Q$, and a transition function $\delta : Q \times \Sigma \to Q$. We extend the transition function to words $\alpha \in \Sigma^*$ as follows: for $a \in \Sigma$, $\alpha \in \Sigma^*$, and $q \in Q$: $\delta(q, a \cdot \alpha) = \delta(\delta(q, a), \alpha)$ and $\delta(q, \epsilon) = q$. For $q \in Q$ we write I_q to denote the set of strings reaching q from the initial state: $I_q = \{\alpha \in \Sigma^* \mid q = \delta(s, \alpha)\}$. The language $\mathcal{L} \subseteq \Sigma^*$ recognised by $\mathcal{A}$ is the set of strings reaching a final state from the initial state: $\mathcal{L}(\mathcal{A}) = \bigcup_{q \in F} I_q$. We denote by $\mathrm{Pref}(\mathcal{L})$ the collection of prefixes of strings in $\mathcal{L}$.

▶ Remark 1. Since we are interested in $\mathcal{L}$ rather than in the structure of $\mathcal{A}$, we tacitly discarded from $\mathcal{A}$ all states that are not relevant for the definition of $\mathcal{L}$. That is, we assume that all states of $\mathcal{A}$ are reachable from the initial state s and reaching at least a final state $f \in F$. These assumptions imply that for all $q \in Q$ it is $I_q \neq \varnothing$, and $I_q \subseteq \mathrm{Pref}(\mathcal{L})$.

Following the literature [1, 2], we assume that the initial state s has no incoming arcs and that $\mathcal{A}$ is *input-consistent*: $(\forall u, v \in Q)(\delta(u, a_1) = \delta(v, a_2) \to a_1 = a_2)$. These assumptions are not too restrictive since any automaton can be converted into an equivalent input-consistent automaton by just multiplying its size by a factor of $|\Sigma|$ (it is sufficient, for each $a \in \Sigma$ and $q \in Q$, to replace q by a copy q_a duplicating out-going arcs and redirecting all a-arcs entering q to q_a – possibly none).

▶ Remark 2. In an input-consistent automaton all δ-arcs reaching a given state are labelled by the same character. Thus we may *shift* labels from arcs to states, obtaining an equivalent *state*-labelled automaton. In the following, we will denote by $\lambda(q) \in \Sigma$ the character labelling state q. For the initial state s, which does not have any incoming arc, we set $\lambda(s) = \#$, where $\# \notin \Sigma$ is smaller than any character in Σ.

▶ Remark 3. If $\mathcal{A} = (Q, s, \delta, F)$ is input-consistent, on the grounds of the above observation the second argument of the transition function δ can be safely ignored assuming that $\delta(q) = q'$ stands for $\delta(q, \lambda(q')) = q'$.

▶ **Definition 4.** *A Wheeler DFA (WDFA) $\mathcal{A} = (Q, s, \delta, F, \prec)$ is a DFA endowed with a binary relation $\prec$ such that $(Q, \prec)$ is a total order having the initial state s as minimum, and the following two (Wheeler) properties are satisfied. Let $v_1 = \delta(u_1)$, and $v_2 = \delta(u_2)$:*

- **i** $v_1 \prec v_2 \;\Rightarrow\; \lambda(v_1) \leqslant \lambda(v_2)$;
- **ii** $(\lambda(v_1) = \lambda(v_2) \wedge v_1 \prec v_2) \Rightarrow u_1 \prec u_2$.

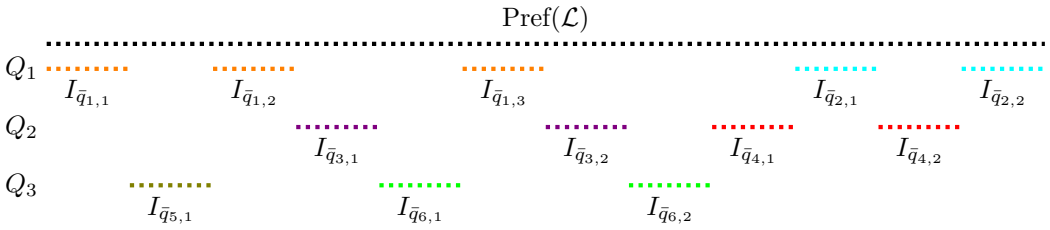
Let $\mathcal{L} \subseteq \Sigma^*$ be a Wheeler language, that is, a language accepted by a deterministic Wheeler automaton. In the rest of the paper we will consider $\mathcal{D} = (Q, s, \delta, F)$ defined as the DFA with the minimum number of states accepting $\mathcal{L}$, and to $\mathcal{D}_w = (Q_w, s, \delta_w, F_w, \prec)$ defined as the WDFA with the minimum number of states accepting $\mathcal{L}$. Uniqueness of $\mathcal{D}$ follows from Myhill-Nerode theorem [8, 9], while uniqueness of $\mathcal{D}_w$ is proven in [2]. By definition it is always $|Q| \leqslant |Q_w|$, but very little else is known about the relative sizes of Q and Q_w; intuitively the ratio $|Q_w|/|Q|$ is the price one has to pay for representing the language $\mathcal{L}$ with a Wheeler automaton.

Definition 4 requires $\prec$ to be a *total* (linear) ordering of the collection of the automaton's states. However, in general, $\mathcal{D}$ does not admit such a total order. Nevertheless $\mathcal{D}$ always admits a *partial* order satisfying (i) and (ii) of Definition 4. Let $p = \mathrm{width}(\mathcal{D})$ (the *width* of

$\mathcal{D}$) be the the minimum number of linear components of a *partial* order satisfying (i) and (ii) of Definition 4. In [5], plenty of arguments are given to illustrate p as a good measure for the distance of $\mathcal{D}$ from being Wheeler.

As established in [2, Lemma 3.4], being $\mathcal{D}_w$ a Wheeler automaton, given $\bar{q} \in Q_w$, the set $I_{\bar{q}} \subseteq \text{Pref}(\mathcal{L})$ of strings reaching $\bar{q}$ from the initial state is an interval $I_{\bar{q}}$ in the linear order $(\text{Pref}(\mathcal{L}), <)$ and the collections of intervals $\{I_{\bar{q}} \mid \bar{q} \in Q_w\}$ constitutes an equivalence relation $\sim_{\mathcal{D}_w}$ partitioning $\text{Pref}(\mathcal{L})$. As a matter of fact, also $\{I_q \mid q \in Q\}$ constitutes an equivalence relation $\sim_{\mathcal{D}}$ partitioning $\text{Pref}(\mathcal{L})$ – even though the I_q's are not, in general, intervals in $(\text{Pref}(\mathcal{L}), <)$ – and $\sim_{\mathcal{D}_w}$ is a *refinement* of $\sim_{\mathcal{D}}$. Hence, for any $\bar{q} \in Q_w$ there exists a unique $q \in Q$, such that $I_{\bar{q}} \subseteq I_q$. In other words, even though I_q for $q \in Q$ might not be an interval, it is always decomposable into a finite collection of intervals $I_{\bar{q}}$'s.

▶ **Example 5.** The following is an example of $\mathcal{D}$ of width 3 where states of $\mathcal{D}_w$ are indexed in such a way that $I_{\bar{q}_{i,j}} \subseteq I_{q_i}$. The partial order of $\mathcal{D}$'s states is: $q_1 \prec q_2; q_3 \prec q_4; q_5 \prec q_6$, with $Q_1 = \{q_1, q_2\}, Q_2 = \{q_3, q_4\}$, and $Q_3 = \{q_5, q_6\}$ being a partition of Q into linearly ordered subsets. Notice that every state of $\mathcal{D}_w$ is reached by interval of strings in $(\text{Pref}(\mathcal{L}), <)$ and any state of $\mathcal{D}$ is reached by a finite collection of intervals of strings.



▶ Remark 6. *Not* being Wheeler for a language means that, for some $q \in Q$, any attempt to produce the previously mentioned decomposition of I_q would result in the introduction of *infinitely many* sub-intervals.

3 The Rational Embedding

In this section we introduce a very simple formal tool, the *rational* embedding, easing the representation and analysis of automata. We begin by embedding Σ^* into $\mathbb{Q}[0, 1)$, the half-open interval of rational numbers between 0 and 1. In what follows, we assume, without loss of generality, that $\Sigma = \{1, 2, \ldots, \sigma\}$ (with the usual order of the integers).

▶ **Definition 7** (The Rational Embedding of Σ^*)**.** *The* Rational Embedding *of* Σ^* *is the map* $\shortparallel : \Sigma^* \to \mathbb{Q}[0, 1)$ *defined as follows. For any* $\alpha = \alpha_1 \ldots \alpha_m \in \Sigma^*$*:*

$$\shortparallel(\alpha) = \sum_{i=1}^{m} \alpha_i \cdot (\sigma + 2)^{-(m-i+1)}.$$

The above embedding sends any non-empty Σ-string to a rational in $(0, 1)$ and the empty word to 0. In the rest of the paper we will always write the values $\shortparallel(\alpha)$ in base $\sigma + 2 = |\Sigma| + 2$; note that by construction the representation will never contain the digit 0 or the digit $\sigma + 1$ to the right of the dot sign.

▶ **Example 8.** Consider the string $\alpha = \alpha_1\alpha_2 \cdots \alpha_m \in \Sigma^*$. The value $\shortparallel$ on α is the rational number $\shortparallel(\alpha)$ written in base $(\sigma + 2)$ as $\shortparallel(\alpha) = 0.\alpha_m \cdots \alpha_2\alpha_1$. Notice that when $\alpha \in \text{Pref}(\mathcal{L})\backslash\{\epsilon\}$, for some $\mathcal{L} = \mathcal{L}(\mathcal{A})$ with $\mathcal{A}$ input-consistent, the most significant digit of $\shortparallel(\alpha)$ is the label of the state reached on $\mathcal{A}$ reading α.

▶ Remark 9. Avoiding 0 and $\sigma+1$ – i.e. the smallest and the largest digits in base $\sigma+2$ – will turn out convenient in order to make the map $\text{ıı}(\cdot)$ injective. This assumption is used in Corollary 21 and, clearly, it does not reduce the overall applicability of the embedding.

The fundamental property of the map $\text{ıı}(\cdot)$, is that the co-lex order on Σ^* corresponds to the order among elements of the rational embeddings of Σ^*. In formulae, denoting by $<$ also the (standard) natural order on $\mathbb{Q}$:

$$\alpha < \beta \text{ (in co-lex order)} \quad \text{if and only if} \quad \text{ıı}(\alpha) < \text{ıı}(\beta) \text{ (as rational numbers)}.$$

Based on the rational embedding of strings we can define the rational embedding of (the states of) a DFA.

▶ **Definition 10.** *Let $I_{\mathbb{Q}[0,1)}$ be the collection of non-empty convex sets of rationals in $\mathbb{Q}[0,1)$:*

$$I_{\mathbb{Q}[0,1)} = \{J \subseteq \mathbb{Q}[0,1) \mid J \neq \varnothing \wedge (\forall a, c \in J)(\forall b \in \mathbb{Q})(a \leqslant b \leqslant c \Rightarrow b \in J)\}.$$

▶ **Definition 11** (The Rational Embedding of a DFA). *The* Rational Embedding *of $\mathcal{A} = (Q, s, \delta, F)$ is the map $I^{\text{ıı}} : Q \to I_{\mathbb{Q}[0,1)}$ defined as follows: for any $q \in Q$,*

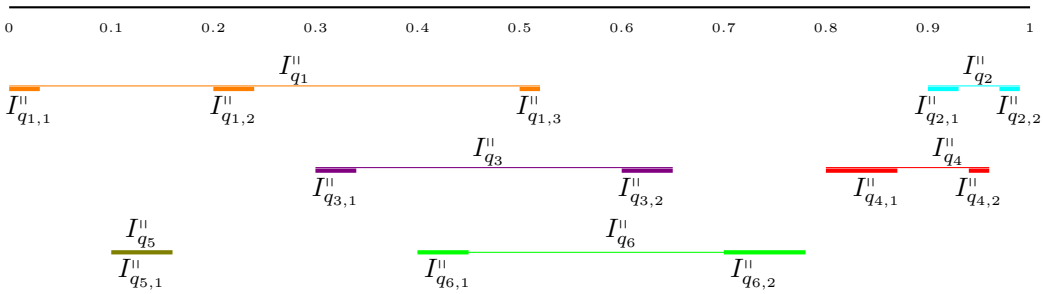
$$I^{\text{ıı}}(q) = \bigcap \{J \in I_{\mathbb{Q}[0,1)} \mid (\forall \alpha \in I_q)(\text{ıı}(\alpha) \in J)\}.$$

In other words, $I^{\text{ıı}}(q)$ is the convex closure (or convex hull) of I_q.

In the following $I^{\text{ıı}}(q)$ will also be denoted by $I^{\text{ıı}}_q$ and we will denote by ℓ_q (respectively r_q) the inf (respectively the sup) of $I^{\text{ıı}}_q$.

Even though determinism guarantees that $q \neq q'$ implies $I_q \cap I_{q'} = \varnothing$, it might be the case that $q \neq q'$ and $I^{\text{ıı}}_q \cap I^{\text{ıı}}_{q'} \neq \varnothing$. However, [2, Theorem 4.3] implies that $\mathcal{A}$ is Wheeler if and only if $q \neq q'$ implies $I^{\text{ıı}}_q \cap I^{\text{ıı}}_{q'} = \varnothing$. This is shown by the following example.

▶ **Example 12.** As already shown in Example 5, given a $\mathcal{D}$-state $q \in Q$ the co-lexicographically ordered words in I_q can be decomposed in a finite sequence of sub-intervals that will constitute the $\mathcal{D}_w$-states. By embedding words and states in $\mathbb{Q}[0,1)$ we simply reproduce this situation on the rationals (as landscape). Below we depict the example, with intervals above referring to states in Q and below to states in Q_w:



Mapping each I_q to a set of real numbers $I^{\text{ıı}}_q$ makes it possible to study automata using tools from elementary calculus. As a first example we show that the notion of accumulation point is related to the concept of *entanglement* introduced in [5, Definition 4.7]. Intuitively, two states q and q' are entangled when there exists an infinite co-lex-monotone sequence of strings reaching alternatively q and q'. Below a formalization of this important notion in a more general setting.

▶ **Definition 13.** *Let $\mathcal{D}$ be a DFA with set of states Q. A subset $Q' \subseteq Q$ is* entangled *if there exists a monotone sequence $(\alpha_i)_{i\in\mathbb{N}}$ in $Pref(\mathcal{L}(\mathcal{D}))$ such that for all $u' \in Q'$ it holds $\delta(s, \alpha_i) = u'$ for infinitely many i's.*

▶ **Definition 14.** *We say that $x \in \mathbb{R}$ is a* left-accumulation point *for a set U if there exists a sequence of elements $u_i \in U$ strictly greater than x and converging to x. Similarly, we say that x is a* right-accumulation point *for U if the elements u_i converging to x are all strictly smaller than x.*

▶ **Lemma 15.** *If a value x is a left-accumulation point (resp. right-accumulation point) for both the sets I''_q and $I''_{q'}$ then q and q' are entangled.*

Proof. If x is a left-accumulation point for both I_q and $I_{q'}$ from elementary calculus we know that there exists an infinite sequence $u_1 > v_1 > \cdots u_i > v_i > \cdots$ converging to x with $u_i \in I''_q$ and $v'_i \in I''_{q'}$. Hence there is a sequence of strings $\alpha_1 \succ \beta_1 \cdots \alpha_i \succ \beta_i \succ \cdots$ with $\alpha_i \in I_q$ and $\beta_i \in I_{q'}$ and the states q and q' are entangled according to Definition 4.7 in [5]. The case when x a right-accumulation point for both I''_q and $I''_{q'}$ is analogous. ◀

▶ **Theorem 16.** *If $\mathcal{D}$ is the minimum DFA accepting $\mathcal{L}$, and x is a left-accumulation point (resp. right-accumulation point) for two distinct sets I''_q and $I''_{q'}$ then $\mathcal{L}$ is not Wheeler.*

Proof. By Lemma 15 the states q and q' are entangled. Since $\mathcal{D}$ is the minimum DFA accepting $\mathcal{L}$ by [5, Theorem 4.21] any DFA recognizing $\mathcal{L}$ has width at least 2. ◀

▶ **Example 17.** Consider the two automata in Figure 1.

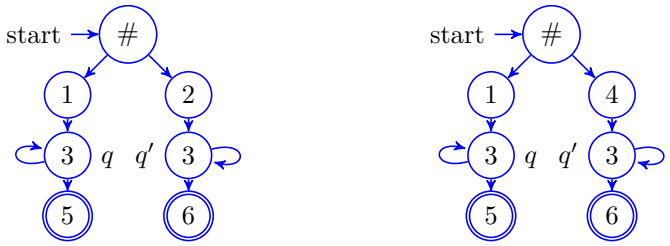


Figure 1 The language accepted by the automaton on the right is Wheeler, while the one accepted by the automaton on the left is not.

The automaton on the right accepts a Wheeler language, while the one on the left does not, the reason being that on the left $0.\bar{3}$ is a right-accumulation point for both I''_q and $I''_{q'}$ because of the sequences $0.31, 0.331, 0.3331, \ldots$ and $0.32, 0.332, 0.3332, \ldots$ (reaching q'): by Theorem 16 the corresponding language is non Wheeler. In the automaton on the right $0.\bar{3}$ is a right-accumulation point for I''_q and a left-accumulation point for $I''_{q'}$ and the automaton is Wheeler with $q \prec q'$. Note that, if in the left automaton we remove states 5 and 6 and make q and q' final, then $0.\bar{3}$ is still a right-accumulation point for both I''_q and $I''_{q'}$ but the resulting automaton is not minimum so Theorem 16 do not apply: indeed the resulting language is Wheeler.

We are particularly interested in the study of the extreme values $\ell_q = \inf I''_q$ and $r_q = \sup I''_q$ as defined in Definition 11. We have the following preliminary results.

▶ **Lemma 18.** *If $\mathcal{L} = \mathcal{L}(\mathcal{D})$ is Wheeler and $\mathcal{D}$ is the minimum DFA accepting $\mathcal{L}$, then for all pairwise distinct $q, q' \in Q$, we have:*

$$\ell_q = \ell_{q'} \;\rightarrow\; (\ell_q \in I''_q \vee \ell_{q'} \in I''_{q'}), \qquad r_q = r_{q'} \;\rightarrow\; (r_q \in I''_q \vee r_{q'} \in I''_{q'}).$$

The same property holds if $\mathcal{D}$ is a WDFA accepting $\mathcal{L}$.

Proof. Assume $\mathcal{D}$ is the minimum DFA for the Wheeler language $\mathcal{L}$. If there exist $q, q' \in Q$ such that $\ell_q = \ell_{q'}$ and $(\ell_q \notin I_q^{\shortparallel} \wedge \ell_{q'} \notin I_{q'}^{\shortparallel})$ then ℓ_q would be a left-accumulation point for both $I_q^{\shortparallel}$ and $I_{q'}^{\shortparallel}$ which is impossible by Theorem 16.

If instead $\mathcal{D}$ is a WDFA accepting $\mathcal{L}$ observe that $\ell_q = \ell_{q'}$ and $(\ell_q \notin I_q^{\shortparallel} \wedge \ell_{q'} \notin I_{q'}^{\shortparallel})$ implies $I_q^{\shortparallel} \cap I_{q'}^{\shortparallel} \neq \varnothing$, which would contradict the hypothesis that $\mathcal{D}$ is Wheeler. The case $r_q = r_{q'}$ is entirely analogous for both kind of automata. ◀

▶ **Lemma 19.** *If $\lambda_q \in \mathcal{L}$ is such that $\shortparallel(\lambda_q) = \ell_q$, then: $\ell_q \in I_q^{\shortparallel}$ if and only if $\lambda_q \in I_q$. The same result holds for r_q as well.*

Proof. See Appendix. ◀

Using Lemma 18 we can order the intervals $I_q^{\shortparallel}$'s according to their left ends ℓ_q's (or their right ends r_q's) breaking ties, when $\ell_q = \ell_{q'}$, by setting $I_q^{\shortparallel}$ less than $I_{q'}^{\shortparallel}$ if $\ell_q \in I_q^{\shortparallel}$ and $\ell_{q'} \notin I_{q'}^{\shortparallel}$ (we cannot have $\ell_q \in I_q^{\shortparallel} \wedge \ell_q \in I_{q'}^{\shortparallel}$ since by Lemma 19 we would have $I_q \cap I'_q \neq \varnothing$). The rationale for this tie-breaking rule is that $\ell_q \in I_q^{\shortparallel}$ ensures that there is an element in $I_q^{\shortparallel}$ which is strictly smaller than all elements in $I_{q'}^{\shortparallel}$. However, we will see below (Corollary 21) that, in fact, breaking ties will never be necessary.

Using the above ordering of the intervals we can derive a procedure to determine the values ℓ_q and r_q, for all $q \in Q$.

▶ **Lemma 20.** *Let $\mathcal{L} = \mathcal{L}(\mathcal{D})$, with $\mathcal{L}$ Wheeler and $\mathcal{D}$ either minimum or Wheeler. For any $q \in Q$ we have:*

$$\ell_q = 0.a_{q,1} \cdots a_{q,h}\overline{a_{q,h+1} \cdots a_{q,h+j}},$$

with $h + j \leqslant |Q|$, and $j = 0$ meaning that ℓ_q is not periodic. Moreover, $j > 0$ if and only if $\ell_q \notin I_q^{\shortparallel}$. An analogous characterisation holds for r_q.

Proof. Let $q_0 = s < q_1 < \ldots < q_n$ be the total order of Q induced by the order of the intervals $I_q^{\shortparallel}$ mentioned above, that is

$$q_i < q_{i'} \overset{\text{def}}{\iff} \left(\ell_{q_i} < \ell_{q_{i'}} \vee (\ell_{q_i} = \ell_{q_{i'}} \wedge \ell_{q_{i'}} \notin I_{q_{i'}}^{\shortparallel})\right). \tag{1}$$

Algorithm 1 determines the digits and the (possible) periodicity of ℓ_q for any state q.

After a call of `left_dd`(q), let $P = \{q_{i_1}, \ldots, q_{i_{k-1}}\}$. It is easily seen by induction that the digits determined $a_{q,1} \cdots a_{q,k-1}$ are, in fact, the first $k-1$ digits of ℓ_q. Upon exit of `left_dd`(q), $k - 1 = h + j < |Q|$ and the algorithm stops in one of the following two cases:

1. $q_{i_k} = s$, or
2. $q_{i_k} = q_{i_{k'}}$, for some $k' \in \{1, \ldots, k-1\}$.

In the first case $h = k - 1$, $j = 0$, and $\ell_q = 0.a_{q,1} \cdots a_{q,h}$, as determined by `left_dd`$(q)$. In the second case $h = k' - 1$, $j = k - k'$, and our claim is that

$$\ell_q = 0.a_{q,1} \cdots a_{q,h}\overline{a_{q,h+1} \cdots a_{q,h+j}}.$$

In fact, in this case it is easy to produce a sequence of strings reaching q whose rational embeddings converge to ℓ_q. Take, for example, β labelling a simple path from s to $q_{i_{k'}}$ and consider the following infinite sequence of words reaching q: for $i \in \mathbb{N}$,

$$\beta(a_{q,h+j} \cdots a_{q,h+1})^i a_{q,h} \cdots a_{q,1}.$$

Algorithm 1 `left_digits_detector(`q`) (left_dd(`q`))`.

$k \leftarrow 1$; // initialise a counter for visited states (and for the digits)
$q_{i_k} \leftarrow q$; // set the first state (and digit)
$P \leftarrow \varnothing$; // P will store visited states
while $q_{i_k} \neq s$ and $q_{i_k} \notin P$ **do** // stop when s or a state in P is reached
  $P \leftarrow P \cup \{q_{i_k}\}$;
  $a_{q,k} \leftarrow \lambda(q_{i_k})$;
  $k \leftarrow k + 1$;
  $i_k \leftarrow \min\left\{k' \mid \delta(q_{k'}) = q_{i_{k-1}}\right\}$; // use the ordering (1)
if $q_{i_k} = s$ **then** // ℓ_q is not periodic
  $h \leftarrow k - 1$;
  $j \leftarrow 0$;
else // q_{i_k} is a previously visited state: set periodicity
  let $k' < k$ such that $q_{i_k} = q_{i_{k'}}$
  $h \leftarrow k' - 1$
  $j \leftarrow k - k'$;

By construction we have that, for any $i \in \mathbb{N}$,

$$\text{ıı}(\beta(a_{q,h+j} \cdots a_{q,h+1})^{i+1} a_{q,h} \cdots a_{q,1}) < \text{ıı}(\beta(a_{q,h+j} \cdots a_{q,h+1})^{i} a_{q,h} \cdots a_{q,1}),$$

and that:

$$\ell_q = \lim_{i \to \infty} \text{ıı}(\beta(a_{q,h+j} \cdots a_{q,h+1})^{i} a_{q,h} \cdots a_{q,1}). \quad \blacktriangleleft$$

Using the characterisation of ℓ_q and r_q provided by Lemma 20 we can strengthen Lemma 18 and prove that different $I_q^{\text{ıı}}$'s have always different left and right limits.

▶ **Corollary 21.** *Let $\mathcal{L} = \mathcal{L}(\mathcal{D})$, with $\mathcal{L}$ Wheeler and $\mathcal{D}$ either minimum or Wheeler. Then, for any pairwise distinct $q, q' \in Q$ we have: $\ell_q \neq \ell_{q'}$ and $r_q \neq r_{q'}$.*

Proof. See Appendix. ◀

The above corollary clarifies that any state $q \in Q$ can be uniquely characterised by a rational number or, equivalently, by a string of at most $|Q| - 1$ characters.

▶ **Theorem 22.** *If $\mathcal{L} = \mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{D}_w)$, with $\mathcal{L}$ Wheeler and $\mathcal{D}$ either minimum or Wheeler, then for all $q \in Q$, we have $\ell_q, r_q \in \mathbb{Q}$.*

Proof. Follows directly from Lemma 20. ◀

▶ Remark 23. We can give examples of automata $\mathcal{D}$ such that, for some $q \in Q$, $I_q^{\text{ıı}}$ includes Cauchy sequences of rational embeddings of strings converging to irrational numbers – as a matter of fact, this easily follows from the fact that the language Σ^* is Wheeler. However, Theorem 22 ensures that such irrational limits of (encodings of) words will never occur as endpoints of $I_q^{\text{ıı}}$'s. In fact, Lemma 20, exploiting the linear order of the reals, used by algorithm `left_dd` to direct the search, shows that – not surprisingly, being $\mathcal{D}$ a finite automaton – a *finite* representation of the bounding elements of $I_q^{\text{ıı}}$'s, can be given.

Corollary 21 immediately implies that the existence of distinct states with equal left (right) limits guarantees non-Wheelerness. The converse of the above result is, in general, not true as illustrated by the following example.

▶ **Example 24.** The automaton in Figure 2 is such that $I^{\shortparallel}_q = [0.51, 0.56]$ and $I^{\shortparallel}_{q'} = [0.52, 0.57]$. However, the value $0.\overline{5}$ is a left-accumulation point for both q and q'. By Theorem 16 the accepted language is not Wheeler even if all the left and right limit are distinct.

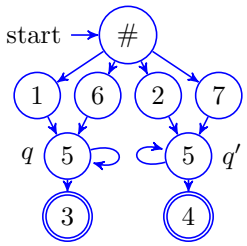


Figure 2 States q and q' in the above automaton are entangled by Lemma 15; however $\ell_q \neq \ell_{q'}$ and $r_q \neq r_{q'}$.

4 On the number of states of the minimum WDFA

Given $\mathcal{D}_w = (Q_w, s_w, \delta_w, F_w, \prec)$ minimum (in the number of states) WDFA accepting a Wheeler language $\mathcal{L} = \mathcal{L}(\mathcal{D})$, with $\mathcal{D} = (Q, s, \delta, F)$ minimum DFA accepting $\mathcal{L}$, we want to study the relationship between the size of $\mathcal{D}_w$ and $\mathcal{D}$.

Consider the collection of intervals $\{I^{\shortparallel}_q \mid q \in Q_w\}$. Since $\mathcal{D}_w$ is Wheeler, as already observed we have that for pairwise distinct $q, q' \in Q_w$, $I^{\shortparallel}_q \cap I^{\shortparallel}_{q'} = \varnothing$. This is, in general, not the case for $\mathcal{D}$ and below we prove that the size of $\mathcal{D}_w$ can be exponential in the size of $\mathcal{D}$, even in case $\text{width}(\mathcal{D}) = 2$.

Below we give a simple example of automaton $\mathcal{D}^1$ accepting a Wheeler language but such that the minimum WDFA $\mathcal{D}^1_w$ has size exponential in the size of $|\mathcal{D}^1|$. Let $\mathcal{D}^1$ be the automaton in Figure 3.

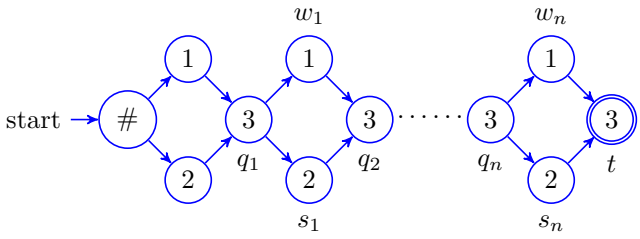
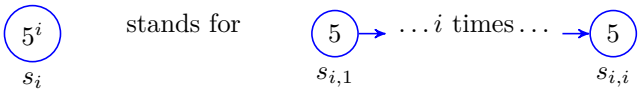


Figure 3 The depicted DFA is accepting a Wheeler (finite) language and the minimum accepting Wheeler DFA accepting the same language has size exponential in n.

$\mathcal{L} = \mathcal{L}(\mathcal{D}^1)$, being finite, is Wheeler [2]. Moreover, any Wheeler automaton accepting $\mathcal{L}$ must have a number of states exponential in n. In fact, given any pair of strings $\alpha, \gamma \in I_t$ such that $\shortparallel(\alpha) < \shortparallel(\gamma)$, it is easy to find a $\beta \in I_{q_n}$ such that $\shortparallel(\alpha) < \shortparallel(\beta) < \shortparallel(\gamma)$. Since there are exponentially many pairwise distinct strings reaching state t, in a Wheeler automaton the set I_t must be partitioned into an exponential number of sub-intervals. Hence, state t must be "split" into exponentially many states of $\mathcal{D}_w$ and the size of $\mathcal{D}_w$ must be exponential in n.

▶ Remark 25. The automaton $\mathcal{D}^1$ of Figure 3 has $\text{width}(\mathcal{D}^1) = n$. Hence, one could think that the explosion in the number of states is exponential in the width of the minimum automaton accepting a given language. Below we show that this is not the case, providing an example of automaton whose width is just 2 but the explosion still occurs.

Consider the DFA $\mathcal{D}^2$ of Figure 4, where, for example, state s_i labeled 5^i stands for a sequence of i states $s_{i,1}, \ldots, s_{i,i}$, all labeled 5. That is:



and, analogously, for states s'_i, w_i and w'_i, for $i \in \{1, \ldots, n\}$. Furthermore, we say that states $s_{i,j}$ and $s'_{i,j}$ are *twins* – the same goes for all other states and their *primed* version.

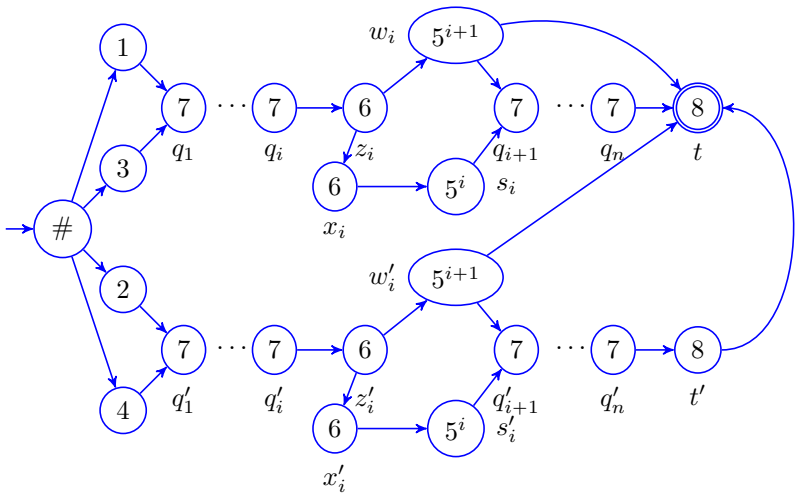


Figure 4 The automaton $\mathcal{D}^2$. The gadget between q_i and q_{i+1}, consisting of states z_i, x_i, w_i, s_i, (and, analogously, the gadget between q'_i and q'_{i+1}) is deployed for $i = 1, \ldots, n-1$. Notice that the i-th copy consists of $2i + 4$ states overall since each label 5^i expands to i states.

Table 1 Left and right limits of $I^{\shortparallel}$ for different kinds of states from automaton $\mathcal{D}^2$. Intervals of states denoted by different letters are clearly non-intersecting except in the case of t with t'.

State type	Left limit	Right limit
$s_{i,j}$	$0.5^j6675^i67\ldots$	$0.5^j6675^{i-1}667\ldots$
$w_{i,j}$	$0.5^j675^i67\ldots$	$0.5^j675^{i-1}667\ldots$
x_i	$0.6675^i67\ldots$	$0.6675^{i-1}667\ldots$
z_i	$0.675^i67\ldots$	$0.675^{i-1}667\ldots$
q_i	$0.75^i67\ldots$	$0.75^{i-1}667\ldots$
t	$0.85^n67\ldots$	$0.8875^{n-1}667\ldots$
t'	$0.875^n67\ldots$	$0.875^{n-1}667\ldots$

Our goal is to show that $\mathcal{D}^2$ has width equal to 2. The following two lemmas, whose proofs can be found in the appendix, ensure that non-empty intersection of intervals happens only between twin states.

▶ **Lemma 26.** *Let u, v states of any automaton. If $\lambda(u) \neq \lambda(v)$, then $I^{\shortparallel}_u \cap I^{\shortparallel}_v = \varnothing$.*

▶ **Lemma 27.** *Let $\mathcal{D}^2$ be the automaton in Figure 4. Let u, v be a pair of distinct states of $\mathcal{D}^2$ such that $\lambda(u) = \lambda(v)$. Then, $I^{\shortparallel}_u \cap I^{\shortparallel}_v \neq \varnothing$ if and only if u and v are twins.*

By above lemmas it follows:

▶ **Lemma 28.** *Let $\mathcal{D}^2$ be the automaton of Figure 4. Then,* $\text{width}(\mathcal{D}^2) = 2$.

$\mathcal{D}^2$ accepts a (finite) Wheeler language, but its minimum *Wheeler* automaton has exponential size in n. The fact that $\mathcal{D}^2_w$ has size exponential in n is verified observing that strings reaching t and t' are interleaved, analogously to $\mathcal{D}^1$.

▶ **Theorem 29.** *Let $\mathcal{L} = \mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{D}_w)$, with $\mathcal{L}$ Wheeler, $\mathcal{D}$ minimum, $\mathcal{D}_w$ minimum Wheeler, and let $f(\cdot,\cdot)$ be such that $|\mathcal{D}_w| = O(f(|\mathcal{D}|, \text{width}(\mathcal{D})))$. Then, for any $k, p \in \mathbb{N}$, $f(n,p) \notin O(n^k + 2^p)$.*

5 Left and right limits: the arithmetic way

In this section we describe an alternative way to determine the left and right limit of the intervals defining the rational embedding of the automaton. Our starting point is the following lemma (proof in Appendix) establishing an arithmetic relationship between the left values ℓ_q's. An analogous result holds for the right values r_q's.

▶ **Lemma 30.** *Given $\mathcal{D} = (Q, s, \delta, F)$, DFA accepting $\mathcal{L}$ Wheeler, and $q \in Q\backslash\{s\}$, there exists a unique $q' \in Q$ such that $\delta(q') = q$ and $(\sigma+2)\cdot \ell_q = \lambda(q) + \ell_{q'}$.*

In the following we use $\mathbb{R}^Q$ to denote the set of real-valued vectors indexed by elements of Q. Given $x \in \mathbb{R}^Q$ and $q \in Q$ we write x_q to denote the entry associated to q. Similarly we use $\mathbb{Q}^Q$ to denote the set of rational-valued vectors. We write ℓ to denote the vector in $\mathbb{Q}^Q$ containing the left limits ℓ_q with $q \in Q$.

Lemma 30 suggests a way of computing left (and right) limits through constraint programming [3, 11]. Formally, for the *left* case, we consider the problem of finding the set of all real-valued vectors $x \in \mathbb{R}^Q$ that satisfy the following constraint satisfaction program, that we name $\mathcal{P}_{Left}$:

$$
\begin{aligned}
&(1) \quad x_s = 0, && \\
&(2) \quad 0 < x_q < 1, && (\forall q \in Q\backslash\{s\}) \\
&(3) \quad (\sigma+2)\cdot x_q = \lambda(q) + \min\left\{x_{q'} \mid \delta(q') = q\right\}, && (\forall q \in Q\backslash\{s\})
\end{aligned}
$$

We now prove that the vector $\ell \in \mathbb{Q}^Q$ of left limits is the only solution of the above program. As a first step, we make sure that $\mathcal{P}_{Left}$ is complete, that is, the vector ℓ satisfies constraints (1–3).

▶ **Lemma 31.** *Let $\mathcal{L}$ be a Wheeler language, and $\mathcal{D} = (Q, s, \delta, F)$ be either minimum or Wheeler accepting $\mathcal{L}$, and let $\ell \in \mathbb{Q}^Q$ be the vector of left limits. Then, ℓ is a solution of $\mathcal{P}_{Left}$.*

Proof. First of all, notice that constraints (1) and (2) of $\mathcal{P}_{Left}$ are clearly satisfied by ℓ. Consider the order $<_Q$ of the states of $\mathcal{D}$ defined by: $q <_Q q' \overset{\text{def}}{\iff} \ell_q < \ell_{q'}$. The order is well-defined and total in virtue of Corollary 21. By Lemma 30, for every state $q \neq s$ there exists a *unique* $q' \in \delta^{-1}(q)$ such that $(\sigma+2)\cdot\ell_q = \lambda(q) + \ell_{q'}$. Moreover, from the proof of Lemma 20 we know that $q' = \min_{<_Q} \delta^{-1}(q)$. By definition of $<_Q$ we have:

$$
q' = \min_{<_Q} \delta^{-1}(q) \iff \ell_{q'} = \min\left\{\ell_{q''} \mid q'' \in \delta^{-1}(q)\right\},
$$

thus satisfying constraint (3). ◀

To prove that ℓ is the only solution we need the notion of (x, q)-min-path.

▶ **Definition 32.** *Let $\mathcal{D} = (Q, s, \delta, F)$ be a DFA, $x \in \mathbb{R}^Q$, and $q \in Q$. We say that an infinite sequence of states $(q_i)_{i \geqslant 1}$ is a (x, q)*-min-path *in $\mathcal{D}$ if the following hold:*

1. $q_1 = q$,
2. $(\forall i \geqslant 1)(\delta(q_{i+1}) = q_i \vee q_i = q_{i+1} = s)$,
3. $(\forall i \geqslant 1)(x_{q_{i+1}} = \min\{x_{q'} \mid \delta(q') = q_i\} \vee q_i = q_{i+1} = s)$.

Roughly speaking, a (x, q)-min-path is a path in the automaton that follows (backward) states whose associated x-value is minimum. It does not come as a surprise that if $(q_1, q_2, \dots)$ is a (x, q)-min-path, then for every $j \geqslant 1$ we have that $(q_j, q_{j+1}, \dots)$ is a (x, q_j)-min-path as well: the proof of this simple fact follows directly from Definition 32.

Furthermore, when $x \in \mathbb{R}^Q$ is a solution of $\mathcal{P}_{Left}$, (x, q)-min-paths spell out precisely x_q's digits. Formally:

▶ **Lemma 33.** *Let $x \in \mathbb{R}^Q$ be a solution of $\mathcal{P}_{Left}$, $q \in Q$, and let $(q_i)_{i \geqslant 1}$ be any (x, q)-min-path. Then, for every $j \geqslant 1$, the j-th digit of x_q is $\lambda(q_j)$.*

Proof. See Appendix. ◀

▶ **Corollary 34.** *If $x \in \mathbb{R}^Q$ is a solution of $\mathcal{P}_{Left}$, then for every $q \in Q$ the first digit of x_q is $\lambda(q)$.*

Proof. Follows immediately from Definition 32 and Lemma 33. ◀

We can now state the main result of this section.

▶ **Theorem 35.** *Let $\mathcal{D} = (Q, s, \delta, F)$ be either minimum or Wheeler accepting $\mathcal{L}$ Wheeler, and $\ell \in \mathbb{Q}^Q$ be the vector of left limits. Then, $\mathcal{P}_{Left}$ always admits ℓ as its* unique *solution.*

Proof. By Lemma 31 we know that ℓ is a solution of $\mathcal{P}_{Left}$. Let x be a generic solution of $\mathcal{P}_{Left}$, and denote by $x_{q,j}$ the j-th digit of x_q. Suppose, for the sake of contradiction, that there exists some state $q \in Q$ such that $x_q \neq \ell_q$, let q be any for which x_q and ℓ_q have the shortest prefix of digits in common, and let j be the length of such prefix. By Corollary 34, $j \geqslant 1$. Let $(q_i)_{i \geqslant 1}$ and $(q'_i)_{i \geqslant 1}$ be, respectively, any (x, q)-min-path and (ℓ, q)-min-path. By Lemma 33 and hypothesis on q, it holds:

1. $(\forall i \leqslant j)(\lambda(q_i) = x_{q,i} = \ell_{q,i} = \lambda(q'_i))$, and
2. $\lambda(q_{j+1}) = x_{q,j+1} \neq \ell_{q,j+1} = \lambda(q'_{j+1})$.

Consider the case $\lambda(q_{j+1}) < \lambda(q'_{j+1})$ (the other case is symmetric). By minimality of the choice of q, the first j digits of both x_{q_2} and ℓ_{q_2} are the same. Similarly, the first j digits of both $\ell_{q'_2}$ and $x_{q'_2}$ are the same. Therefore, $\ell_{q_2} < \ell_{q'_2}$ contradicting the hypothesis that $(q'_i)_{i \geqslant 1}$ was a (l, q)-min-path. ◀

Program $\mathcal{P}_{Left}$ can be implemented as a Mixed Integer Program, whose solution is, in general, computationally hard to obtain [10]. The fact that a graph-oriented approach can compute the left limits in polynomial time [7], justifies the following:

▶ **Conjecture 36.** *There exists a linear programming model equivalent to $\mathcal{P}_{Left}$.*

We give a partial answer to Conjecture 36. Let $\pi : Q \to Q$ be the *parent* function, *i.e.* $\pi(q) = q'$ if and only if q' is the unique state mandated by Lemma 30 (with $\pi(s) = s$). Indeed, π is precisely what has been computed in [7, Section 4] in the form of a pruned automaton. Consider matrix $\Pi \in \{0, 1\}^{|Q| \times |Q|}$ such that $\Pi_{i,j} = 1$ if and only if $\pi(i) = j$. Combining $\mathcal{P}_{Left}$ and Theorem 35, we express the problem of computing vector ℓ of left limits given vector λ of characters as the unique solution of:

$$\begin{cases} x_s = 0, \\ (\forall q \neq s)(0 < x_q < 1), \\ (\sigma + 2) \cdot x = \lambda + \Pi^\intercal \cdot x \end{cases}$$

The third equation reminds of the condition for x to be an eigenvector of $\Pi^\intercal$, with λ acting as a *corrective* term. This intuition can be made formal. In fact, if we denote by I the $|Q| \times |Q|$ identity matrix, it is easily verified that:

$$\begin{pmatrix} \Pi^\intercal & I \\ 0 & (\sigma+2)I \end{pmatrix} \cdot \begin{pmatrix} x \\ \lambda \end{pmatrix} = (\sigma + 2) \begin{pmatrix} x \\ \lambda \end{pmatrix}$$

As already stated, we consider the above model unsatisfactory since we would need to know Π in advance. Finally, for the right limits case, we define the constraint satisfaction program $\mathcal{P}_{Right}$ by substituting max for min in (3) of $\mathcal{P}_{Left}$:

$$(3^*) \quad (\sigma + 2) \cdot x_q = \lambda(q) + \max \left\{ x_{q'} \mid \delta(q') = q \right\}, \qquad (\forall q \in Q \backslash \{s\})$$

The discussion made for $\mathcal{P}_{Left}$ computing ℓ can be translated into a discussion for $\mathcal{P}_{Right}$ computing the vector of right limits r by exchanging min-arguments into max-arguments. The key observation is that Lemma 33, rephrased in terms of *max-paths*, still holds.

6 Conclusions

One of the goals of this paper was to study the problem of building, given a Wheeler language presented by its minimum accepting automaton, a Wheeler accepting automaton of minimum size. Such minimum Wheeler DFA is proved to be, in general, exponential in size with respect to the size of the minimum input DFA. The lower bound is proved by exhibiting an example of DFA whose size explodes exponentially when we perform the "splits" necessary to guarantee the order characterizing Wheeler-ness. Moreover, and most importantly, this happens even when the *width* of the input DFA is just 2. We point out that the latter phenomenon is a sort of exception: for most classic operations, once the width is fixed we are able to put polynomial bounds on their complexity.

The above result is illustrated while introducing a simple view on DFAs and WDFAs, that starts from a mapping of strings into rational numbers. According to this *rational* embedding, automaton's states can be (over)approximated by convex sets (intervals) of rational numbers and the basic Wheeler properties turn out to be translated into ordering and non-intersecting constraint on the collection of states-intervals. Moreover, a characterisation of the number of digits necessary to identify left and right limits of states-intervals can be carried out analysing the underlying automaton's transition function and using the Wheeler order of its states.

The latter technique suggests also that the infinite alternation of strings reaching different states (the so-called *entanglement* of states) can be linked with the existence, position, and distribution of accumulation points of the collection of embedding of prefixes of strings on the $[0, 1)$ half-open interval of the real line. An interesting further direction of study is the characterisation of order-types obtainable by rationals corresponding to embedding of prefixes of general, not necessarily Wheeler, languages. The final section is devoted to propose a further angle from which the problem of determining digits of limiting rationals can be approached, namely constraint programming.

References

1 Jarno Alanko, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Regular languages meet prefix sorting. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 911–930. SIAM, 2020. doi:10.1137/1.9781611975994.55.

2 Jarno Alanko, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Wheeler languages. *Inf. Comput.*, 281:104820, 2021. doi:10.1016/j.ic.2021.104820.

3 Krzysztof Apt. *Principles of constraint programming*. Cambridge university press, 2003. doi:10.1017/CBO9780511615320.

4 Michael Burrows and David J Wheeler. A block-sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994.

5 Nicola Cotumaccio, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Co-Lexicographically Ordering Automata and Regular Languages - Part I. *J. ACM*, 70(4), August 2023. doi:10.1145/3607471.

6 Travis Gagie, Giovanni Manzini, and Jouni Sirén. Wheeler graphs: A framework for BWT-based data structures. *Theoretical Computer Science*, 698:67–78, 2017. Algorithms, Strings and Theoretical Approaches in the Big Data Era (In Honor of the 60th Birthday of Professor Raffaele Giancarlo). doi:10.1016/j.tcs.2017.06.016.

7 Sung-Hwan Kim, Francisco Olivares, and Nicola Prezza. Faster prefix-sorting algorithms for deterministic finite automata. In *Proc. CPM 23*. Schloss-Dagstuhl - Leibniz Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CPM.2023.16.

8 John Myhill. Finite automata and the representation of events. *WADD Technical Report*, 57:112–137, 1957.

9 Anil Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958.

10 Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981. doi:10.1145/322276.322287.

11 Francesca Rossi, Peter Van Beek, and Toby Walsh. Constraint programming. *Foundations of Artificial Intelligence*, 3:181–211, 2008. doi:10.1016/S1574-6526(07)03004-0.

A Proofs

Proof of Lemma 19. Since $I_q^{\shortparallel}$ is the convex closure of I_q, it is $I_q \subseteq I_q^{\shortparallel}$ and the "if" implication is immediate. If it were $\ell \notin I_q \wedge \ell_q \in I_q^{\shortparallel}$, by considering $U = I^{\shortparallel} \backslash \{\ell_q\}$ we would get a convex set $U \supseteq I_q$ strictly contained in $I_q^{\shortparallel}$ which is a contradiction. ◀

Proof of Corollary 21. We deal with left limits only since the argument for right limits is entirely similar. If both ℓ_q and $\ell_{q'}$ are not periodic, by Lemma 20 and the fact that $\mathcal{D}$ is deterministic it must be the case that $\ell_q \neq \ell_{q'}$.

Otherwise, by Lemma 18 we have that, say, $\ell_q \in I_q^{\shortparallel}$ and $\ell_{q'} \notin I_{q'}^{\shortparallel}$. By Lemma 20 this means that $\ell_{q'}$ is a periodic rational while ℓ_q is not. Since the largest digit of Σ will never label a state (see Remark 9), the two rational numbers ℓ_q and $\ell_{q'}$ cannot possibly be equal. ◀

Proof of Lemma 26. Suppose, without loss of generality, that $\lambda(u) < \lambda(v)$. It is clear that:

$$r_u = 0.\lambda(u)\cdots \; < \; 0.\lambda(v)\cdots \; = \ell_v.$$

Thus, their respective intervals do not intersect. ◀

Proof of Lemma 27. ($\Leftarrow$) The case for t and t' is clearly true (see Table 1). Consider two twin states u and u', respectively from the top and the bottom level of $\mathcal{D}^2$. By construction, there exist $\alpha, \beta \in \Sigma^*$ such that:

$$\begin{array}{ccccccc} 0.\alpha 1 & < & 0.\alpha 2 & < & 0.\beta 3 & < & 0.\beta 4 \\ \| & & \| & & \| & & \| \\ \ell_u & & \ell_{u'} & & r_u & & r_{u'} \end{array}$$

Thus, $I^{\shortparallel}_u \cap I^{\shortparallel}_{u'} \neq \varnothing$.

($\Rightarrow$) We prove the contrapositive. The case for states denoted by different letters is simple (see again Table 1). Let u and v be two non-twin states denoted by the same letter and different indexes, and suppose, without loss of generality, that $u = s_{i,j}$ and $v = s_{i',j'}$ (all other cases are proved in a similar way). We have two cases. If $j = j'$ and $i < i'$, then:

$$r_v = 0.5^j 6675^i 5^{i'-i-1} 667\cdots < 0.5^j 6675^i 67\cdots = \ell_u$$

and their respective intervals do not intersect. Otherwise, if $j < j'$, then:

$$r_v = 0.5^j 5^{j'-j} 667\cdots < 0.5^j 667\cdots = \ell_u$$

and, again, their respective intervals do not intersect. ◀

Proof of Lemma 30. By Lemma 20, we have:

$$\ell_q = 0.a_{q,1}\cdots a_{q,h}\overline{a_{q,h+1}\cdots a_{q,h+j}} = 0.\lambda(q)a_{q,2}\cdots a_{q,h}\overline{a_{q,h+1}\cdots a_{q,h+j}}.$$

Let q' be the first state visited after q by $\texttt{left\_dd}(q)$, we have:

$$\ell_{q'} = \begin{cases} 0.a_{q,2}\cdots a_{q,h}\overline{a_{q,h+1}\cdots a_{q,h+j}}, & \text{if } h > 0, \\ 0.\overline{a_{q,2}\cdots a_{q,j-1}\lambda(q)}, & \text{if } h = 0. \end{cases}$$

Since all the above values are expressed in base $\sigma + 2$ it is

$$(\sigma+2)\cdot\ell_q \;=\; \lambda(q).a_{q,2}\cdots a_{q,h}\overline{a_{q,h+1}\cdots a_{q,h+j}} \;=\; \lambda(q) + \ell_{q'}$$

as claimed. The uniqueness of q' follows from Corollary 21. ◀

Proof of Lemma 33. First of all, the unique (x, s)-min-path is $(s, s, \dots)$. Therefore, the j-th digit of x_s is $\lambda(s)$ for every j. If $q \neq s$, we prove the lemma by induction on $j \geqslant 1$. In what follows, we denote by $x_{q,j}$ the j-th digit of x_q, and we let $(q_i)_{i\geqslant 1}$ be any (x, q)-min-path.

Base. By Definition 32 and constraints of $\mathcal{P}_{Left}$ we have $\lambda(q_1) \leqslant (\sigma+2)\cdot x_q < \lambda(q_1) + 1$. Thus, $x_{q,1} = \lambda(q_1)$.

Step. Let $j > 1$, and suppose the property holds for every state and every $j' < j$. Sequence $(q_2, q_3, \dots)$ is a (x, q_2)-min-path. Thus:

$$\begin{aligned} x_{q,j} &= x_{q_2,j-1} && \text{(Constraint 3 of } \mathcal{P}_{Left} \text{ and Def. 32)} \\ &= \lambda(q_j) && \text{(Induction hypothesis on } q_2 \text{ and } j-1) \end{aligned}$$

◀