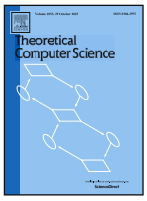
Theoretical Computer Science 1060 (2026) 115631

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs



Completing Wheeler Automata

Giuseppa Castiglione*, Antonio Restivo

Dipartimento di Matematica e Informatica, Università degli Studi di Palermo, Italy

ARTICLE INFO

Editor: Dr. Pinyan Lu

Keywords:
Wheeler Automata
Complete Automata
Boolean Operations

ABSTRACT

We consider the problem of embedding a Wheeler Deterministic Finite Automaton (WDFA, in short) into an equivalent complete WDFA, preserving the order of states and the accepted language. In some cases, such a complete WDFA does not exist. We say that a WDFA is Wheeler-complete (W-complete, in short) if it cannot be properly embedded into an equivalent WDFA. We give an algorithm that, given as input a WDFA $\mathcal{A}$, returns the smallest W-complete DFA containing $\mathcal{A}$: it is called the minimal W-completion of $\mathcal{A}$. We derive some interesting applications of this algorithm concerning the construction of a WDFA for the union and a WDFA for the complement of Wheeler languages.

1. Introduction

The problem of embedding a finite automaton into a complete one while preserving some specific properties is an old problem in automata theory (cf. [2,4,9,12,13]). It is referred to as the completion problem.

In this paper, we approach the completion problem for the class of Wheeler automata, which has recently been introduced in [10]. An automaton in this class has the property that there exists a total order on its states that is propagated along equally labeled transitions. Moreover, the order must be compatible with the underlying order of the alphabet. Wheeler automata play an important role in the emerging field of compressed data structures (cf., for example, [6,11]). The regular languages that can be accepted by a Wheeler automaton are called *Wheeler languages* whose study is deepened in [1,5] and [7].

The completion problem is of particular interest for the class of Wheeler Deterministic Finite Automata (WDFAs) since, in general, the WDFAs are not complete and there exist some Wheeler languages that cannot be accepted by any complete Wheeler automata. In more detail, we consider the problem of embedding a WDFA $\mathcal{A}$ into a complete one, denoted by $C(\mathcal{A})$, such that i) $C(\mathcal{A})$ is a complete WDFA, ii) the (total) order on the states of $C(\mathcal{A})$ is an extension of the order on the states of $\mathcal{A}$ and iii) $C(\mathcal{A})$ is equivalent to $\mathcal{A}$, i.e. they recognize the same Wheeler language. In some cases, this problem has no solution: this means that there exist some WDFAs that cannot be embedded into any equivalent complete WDFA preserving the order of the states. We say that a WDFA is Wheeler-complete (W-complete, in short) if it cannot be properly embedded into an equivalent WDFA. Hence, in any case, there always exists a W-complete DFA in which $\mathcal{A}$ can be embedded, we call it W-completion of $\mathcal{A}$. We show that, among all the W-completions of $\mathcal{A}$, the one with the minimum number of empty states is unique and we denote it by $C_W(\mathcal{A})$.

The main contribution of this paper is a completion algorithm that, having as input a WDFA $\mathcal{A}$, returns its minimal Wheeler-completion $C_W(\mathcal{A})$. In the case where $C_W(\mathcal{A})$ is a complete WDFA we say that $\mathcal{A}$ is completable.

We also consider some relevant applications of this completion algorithm. According to the fact that WDFAs are not in general complete, the family of Wheeler languages is closed under intersection, but it is neither closed under complementation nor under union (cf. [1]).

* Corresponding author.
E-mail addresses: giuseppa.castiglione@unipa.it (G. Castiglione), antonio.restivo@unipa.it (A. Restivo).

https://doi.org/10.1016/j.tcs.2025.115631
Received 21 March 2025; Received in revised form 7 September 2025; Accepted 28 October 2025
Available online 8 November 2025

In the second part of the paper, we use the completion algorithm to construct, under suitable conditions, a WDFA that recognizes the complement of a Wheeler language and a WDFA that recognizes the union of two Wheeler languages. This approach is an alternative to the one proposed in [8].

2. Preliminaries

If Σ is a finite alphabet, with Σ^* we denote the set of finite words on Σ. If $v \in \Sigma^*$ is not the empty word, with $end(v)$ we denote the last letter of v. If $L \subseteq \Sigma^*$, with $Pref(L)$ we denote the set of all prefixes of words in L, $Pref(L) = \{v \in \Sigma^* | \exists u \in \Sigma^* \text{ s.t. } vu \in L\}$.

A *deterministic finite automaton* (DFA) is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, eventually a partial function, s is the initial state and $F \subseteq Q$ is the set of final states. We denote by δ^* the generalized transition function defined on the words of Σ^*. If δ is a total function, the automaton is *complete* while, if δ is a partial function, the automaton is *incomplete*. If $\delta(p, \sigma)$ is not defined for some $p \in Q$ and $\sigma \in \Sigma$ we write $\delta(p, \sigma) = \square$ and say that $\delta(p, \sigma)$ is a *missing transition*. Note that if Σ is the input alphabet, we suppose that the automaton contains at least a transition for each letter.

We denote by $L(\mathcal{A})$ the *language accepted* by $\mathcal{A}$. It is well-known that two automata $\mathcal{A}$ and $\mathcal{B}$ are said to be equivalent if $L(\mathcal{A}) = L(\mathcal{B})$. In what follows, we consider only automata in which all states are *accessible* i.e. can be reached from s. If we denote $L_p(\mathcal{A}) = \{w \in \Sigma^* | \delta^*(p, w) \in F\}$, a state p is an *empty state* if $L_p(\mathcal{A}) = \emptyset$ and is a *coaccesible state* if $L_p(\mathcal{A}) \neq \emptyset$. For any $p \in Q$ and $a \in \Sigma$ we define $In(p) = \{a \in \Sigma | \delta(q, a) = p, \text{for some } q \in Q\}$ and $I_p(\mathcal{A}) = \{w \in \Sigma^* | \delta^*(s, w) = p\}$. An automaton is said *input-consistent* if $|In(p)| = 1$, for each $p \in Q$.

If Σ is ordered with $\prec$ we can define the *co-lexicograhic order* $\prec$ among words of Σ^*. If $u = u_1 \dots u_n$ and $v = v_1 \dots v_m$ then

$$u \prec v \Leftrightarrow (u \text{ is a suffix of } v) \text{ or } (\exists i : u_{n-i} \prec v_{m-i} \text{ and } \forall j < i \; u_{n-j} = v_{m-j})$$

Let $(X, \leq)$ be a total order on X, for any $x, y \in X$ we write $x < y$ if $x \leq y$ and $x \neq y$. We say that two elements $x, y \in X$ are *consecutive* if $x < y$ and there does not exist any $z \in X$ such that $x < z < y$. If $Y \subseteq X$ we say that Y is *a convex set* in X if $\forall x, y \in Y$ and $z \in X$, if $x < z < y$ then $z \in Y$.

Definition 1. A *Wheeler DFA* (WDFA) is an input consistent DFA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$, with Σ a total ordered alphabet, $In(s) = \emptyset$ and there exists in Q a total order $\leq$ such that

- the initial state is the minimum;
- let $q_1 = \delta(p_1, \sigma_1)$ and $q_2 = \delta(p_2, \sigma_2)$, $p_1, p_2, q_1, q_2 \in Q$ and $\sigma_1, \sigma_2 \in \Sigma$;
  if $\sigma_1 < \sigma_2$ then $q_1 < q_2$;
  if $\sigma_1 = \sigma_2$ and $p_1 \leq p_2$ then $q_1 \leq q_2$;

As a consequence, if $\sigma_1 \prec \sigma_2$ then, for all $p, q \in Q$, if $In(p) = \{\sigma_1\}$ and $In(q) = \{\sigma_2\}$ then $p < q$. Moreover, in [1], it is proved that for each $p \in Q$ the set $I_p(\mathcal{A})$ is convex in $(Pref(L(\mathcal{A})), \prec)$. Moreover, the following property holds.

Lemma 1 ([1]). *Let $\mathcal{A} = (Q, \Sigma, s, F)$ be a WDFA. Then, for all $u, v \in Pref(L(\mathcal{A}))$,*

$$u \prec v \Rightarrow \delta^*(s, u) \leq \delta^*(s, v) \text{ and } \delta^*(s, u) < \delta^*(s, v) \Rightarrow u \prec v.$$

In the original definition, all the states of a Wheeler DFA are supposed to be coaccessible. However, since we are interested in completing a WDFA, it will be quite natural to add some empty states while maintaining the Wheeler property regarding the order. For this reason, in what follows, we assume that a WDFA can also have some non-coaccessible states. Note that the automaton remains well-defined. If $\mathcal{A} = (Q, \Sigma, \delta, s, F)$, we denote by $T(\mathcal{A})$ the *trim part* of $\mathcal{A}$ i.e. the part of $\mathcal{A}$ that is both accessible and coacceccible. If $T(\mathcal{A}) = \mathcal{A}$ we say that $\mathcal{A}$ is a *trim* automaton.

We conclude the preliminary section with the definition of inclusion among automata, which is fundamental in this paper.

Let $\mathcal{A} = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $\mathcal{B} = (Q_2, \Sigma, \delta_2, s_2, F_2)$, in this paper we say that $\mathcal{A}$ *is contained in* $\mathcal{B}$ (in symbols $\mathcal{A} \subseteq \mathcal{B}$) if $T(\mathcal{A}) = T(\mathcal{B})$ and the transition graph of $\mathcal{A}$ is a subgraph of the transition graph of $\mathcal{B}$. Hence $\mathcal{A}$ and $\mathcal{B}$ are equivalent.

Since we only deal with Wheeler automata, if $\mathcal{A} \subseteq \mathcal{B}$ the order of states in $\mathcal{B}$ is an extension of the order of states in $\mathcal{A}$.

3. Wheeler-complete automata

The classical procedure of completing an automaton by adding a unique empty state (or one for each letter, for input consistency) into which all missing transitions are defined could produce a DFA that is no longer a WDFA. However, in some cases, the automaton can be completed by adding more than one empty state while managing to preserve the Wheeler property, as shown in the following example.

Example 1. Consider the automaton in Fig. 1(a), it is a WDFA (with $a \prec b$ and $s < p < q$) that accepts the language a^+b and is not complete. By adding a single empty state for each letter as in Fig. 1(b) we get the input-consistent complete equivalent automaton. Note that it is not a WDFA in fact if $r < q$ then $q < p$, a contradiction, because $In(p) = \{a\}$ and $In(q) = \{b\}$. Whereas, if $r > q$ then $s > p$, in contradiction to the fact that s is the minimum. In Fig. 1(c) an equivalent complete Wheeler automaton is depicted with states $s < p < t < r < q < w$. Note that the order of the set of states is an extension of the first one, and three empty states have been added.

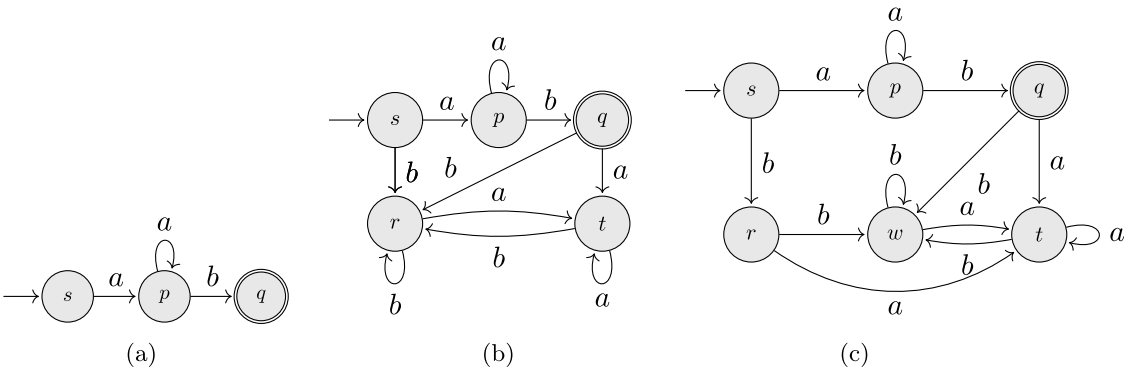


Fig. 1. (a) A WDFA (b) an equivalent complete that is not Wheeler,(c) an equivalent complete WDFA.

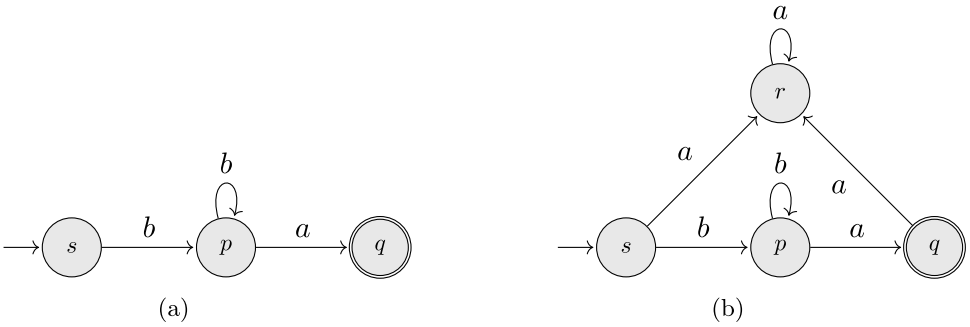


Fig. 2. (a) An incomplete Wheeler automaton accepting b^+a and (b) its (incomplete) W-completion.

In some cases, it is not possible to complete a Wheeler automaton by maintaining the Wheeler property, as the following example shows.

Example 2. The automaton in Fig. 2(a) is a Wheeler automaton (with $a \prec b$ and $s < q < p$) that accepts the language b^+a. In [1] the authors give such a language as an example of a Wheeler language for which no complete Wheeler automaton recognizes it.

In all the other examples that follow, over the alphabet $\{a, b\}$ we assume $a \prec b$.

Given a WDFA $\mathcal{A}$, we here consider the problem of finding, when there exists, an equivalent complete WDFA $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B}$. The following definition introduces a notion of completeness with respect to the Wheeler property.

Definition 2. Let $\mathcal{A}$ be a WDFA. $\mathcal{A}$ is Wheeler-complete (shortly W-complete) if for any equivalent WDFA $\mathcal{B}$ if $\mathcal{A} \subseteq \mathcal{B}$ then $\mathcal{A} = \mathcal{B}$.

The following theorem gives a characterization of W-complete automata.

Theorem 1. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a WDFA. $\mathcal{A}$ is W-complete iff for any missing transition $\delta(q, \sigma) = \square$, with $q \in Q, \sigma \in \Sigma$, there exist $p, t \in Q$ with $p < q < t$ such that $\delta(p, \sigma) = \delta(t, \sigma)$ and it is a coaccessible state.*

Proof. Let $\mathcal{A}$ be W-complete DFA and suppose, by contradiction, that there exists a missing transition $\delta(q, \sigma) = \square$ such that for all $p, t \in Q$, with $p < q < t$ either $\delta(p, \sigma) < \delta(t, \sigma)$ or $\delta(p, \sigma) = \delta(t, \sigma)$ and it is an empty state. In the first case, a new empty state r, with $\delta(p, \sigma) < r < \delta(t, \sigma)$, can be added to the set of states and the extension of the total order of the states can be considered. The missing transition can be filled with the proper transition $\delta(q, \sigma) = r$. We do not define transitions starting from r, and then r is not a coaccessible state. We obtain an equivalent WDFA $\mathcal{B}$ that contains $\mathcal{A}$. This contradicts the hypothesis that $\mathcal{A}$ is W-complete. In the second case, we infer the same contradiction by adding the transition $\delta(q, \sigma) = \delta(p, \sigma)$.

Vice versa, if, by contradiction, $\mathcal{A}$ is not W-complete, there exists an equivalent WDFA $\mathcal{B} = (Q_B, \Sigma, \delta_B, s, F)$ that properly contains $\mathcal{A}$. This means that there exists in $\mathcal{A}$ a missing transition $\delta(q, \sigma) = \square$ such that $\delta_B(q, \sigma) = r$ for some $r \in Q_B$. The state q is accessible, therefore, let $u \in I_q(\mathcal{B})$. By hypothesis, there exist $p, t \in Q$, with $p < q < t$ such that $\delta(p, \sigma) = \delta(t, \sigma) = z$, with z coaccessible state of $\mathcal{A}$ (and then also a coaccessible state of $\mathcal{B}$ because $T(\mathcal{A}) = T(\mathcal{B})$). $\mathcal{B}$ is a WDFA, then $\delta_B(p, \sigma) \leq r \leq \delta_B(t, \sigma)$ which implies $r = z$. Since z is a coaccessible state, there exists $v \in L_z(\mathcal{B})$. It follows that the word $u\sigma v \in L(\mathcal{B})$ and, by the determinism, $u\sigma v \notin L(\mathcal{A})$ contradicting the equivalence of $\mathcal{A}$ and $\mathcal{B}$. $\square$

A W-complete DFA $\mathcal{A}$ contains both coaccessible and empty states, so the following definition makes sense.

Definition 3. Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ a W-complete DFA. We define $Dom(\mathcal{A})$ as the set of words that can be read by $\mathcal{A}$ from the initial state. More formally, $Dom(\mathcal{A}) = \{w \in \Sigma^* \mid \delta^*(s, w) \neq \square\}$.

The following inclusions hold:

$$L(\mathcal{A}) \subseteq Pref(L(\mathcal{A})) \subseteq Dom(\mathcal{A})$$

Remark 1. Lemma 1 holds for W-complete automata as well, with $Dom(A)$ instead of $Pref(L(A))$.

4. The Wheeler Completion

Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a DFA, where $Q = \{q_1, q_2, \dots, q_n\}$ is a totally ordered set of states.

In the sequel, it is convenient to represent the transition function of the DFA as a *transformation* of the set Q of states, i.e., a *partial mapping* of Q into itself (cf., for instance, [3]). For each $\sigma \in \Sigma$, the transformation δ_σ is defined for $q_i \in Q$ as $\delta_\sigma(q_i) = \delta(q_i, \sigma)$. If $\delta(q_i, \sigma)$ is not defined, we write $\delta_\sigma(q_i) = \square$ and say that $\delta_\sigma(q_i)$ is a *missing σ-transition* (or a σ-hole). Hence, an arbitrary partial transformation δ_σ can be written in the form

$$\delta_\sigma = \begin{pmatrix} q_1 & q_2 & \cdots & q_{n-1} & q_n \\ p_1 & p_2 & \cdots & p_{n-1} & p_n \end{pmatrix},$$

where $p_i = \delta_\sigma(q_i)$ and $p_i \in Q \cup \{\square\}$, for $1 \le i \le n$. We denote by $R_\sigma(\mathcal{A})$ the subsequence of $(p_1, p_2, \dots, p_n)$ composed of elements different from $\square$.

With this representation, the property that the DFA, over a totally ordered alphabet Σ, is a WDFA corresponds to the following three conditions:

- q_1 is the initial state;
- for each $\sigma \in \Sigma, q_1 \notin R_\sigma$;
- for each $\sigma \in \Sigma$, R_σ is a non-decreasing sequence;
- denoted by $min(R_\sigma)$ and $max(R_\sigma)$ the first and the last element of R_σ, respectively. If $\sigma < \tau$, then $max(R_\sigma) < min(R_\tau)$.

Definition 4. Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a WDFA. We say that $\mathcal{B}$ is a *Wheeler completion* (shortly a *W-completion*) of $\mathcal{A}$ if $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{B}$ is W-complete.

In dealing with the W-completion of a given WDFA $\mathcal{A}$ we can assume, without loss of generality, that $\mathcal{A}$ is trim. Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a trim WDFA and $\mathcal{B} = (Q', \Sigma, \delta', s', F')$ is a W-completion of $\mathcal{A}$, then by definition $T(\mathcal{B}) = \mathcal{A}$, $s = s'$, $F' = F$ and $Q' = Q \cup S$, where S is the set of empty states of $\mathcal{B}$.

The following example shows that a WDFA can have more than one W-completion.

Example 3. Let us consider the WDFA $\mathcal{A}$ in Fig. 2(a) with with $s < q < p$ with the following transitions:

$$\delta_a = \begin{pmatrix} s & q & p \\ \square & \square & q \end{pmatrix}, \delta_b = \begin{pmatrix} s & q & p \\ p & \square & p \end{pmatrix}$$

By Theorem 1, it is not W-complete. Two of its W-completions are:

$$\mathcal{A}' : \delta'_a = \begin{pmatrix} s & r & q & p \\ r & r & r & q \end{pmatrix}, \delta'_b = \begin{pmatrix} s & r & q & p \\ p & \square & \square & p \end{pmatrix}$$

that is the automaton in Fig. 2(b) and

$$\mathcal{A}'' : \delta''_a = \begin{pmatrix} s & r_1 & r_2 & q & p \\ r_1 & r_1 & r_2 & r_2 & q \end{pmatrix}, \delta''_b = \begin{pmatrix} s & r_1 & r_2 & q & p \\ p & \square & \square & \square & p \end{pmatrix}$$

Indeed, by Theorem 1, they are both W-complete and contain $\mathcal{A}$. Observe that the first one has a minimal number of empty states.

Proposition 1. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ a trim WDFA. If $\mathcal{B}_1 = (Q \cup S_1, \Sigma, \delta_1, s, F)$ and $\mathcal{B}_2 = (Q \cup S_2, \Sigma, \delta_2, s, F)$ are two W-completions of $\mathcal{A}$ then $Dom(\mathcal{B}_1) = Dom(\mathcal{B}_2)$.*

Proof. We prove that $Dom(\mathcal{B}_1) \subseteq Dom(\mathcal{B}_2)$ and analogously we can prove the inverse inclusion. Each $w \in Dom(\mathcal{B}_1)$ can be factored as $w = uv$ such that $u \in \Sigma^*$ is the maximal prefix of w that belongs to $Pref(\mathcal{A})$, and $v \in \Sigma^*$ of length n.

By induction on n we prove that each word w in $Dom(\mathcal{B}_1)$ is a word of $Dom(\mathcal{B}_2)$. If $n = 0$ then $w \in Pref(\mathcal{A}) \subset Dom(\mathcal{B}_2)$. Let $|v| = n + 1$, then $w = uv'x$ with v' of length n and $x \in \Sigma$. Hence $w = uv' \in Dom(\mathcal{B}_1)$ and $w = uv' \in Dom(\mathcal{B}_2)$. By definition, $\delta_1^*(s, uv') = q_1 \in Q \cup S_1, \delta_2^*(s, uv') = q_2 \in Q \cup S_2, \delta_1(q_1, x) \neq \square$ and let us assume for the sake of contradiction that $\delta_2(q_2, x) = \square$ (i.e $uv'x \notin Dom(\mathcal{B}_2)$). Let $p \in Q$ the largest state such that $p < q_1$ and $\delta_1(p, x) \in Q$ (if there exists), and $t \in Q$ the smallest state such that $q_1 < t$ and $\delta_1(t, x) \in Q$ (if there exists). Trivially, $\delta_1(p, x) \neq \delta_1(t, x)$. By Lemma 1 and Remark 1, one has $\alpha \prec uv' \prec \beta$, for all $\alpha \in I_p(\mathcal{B}_1) \subseteq Pref(L(\mathcal{A}))$ and $\beta \in I_q(\mathcal{B}_1) \subseteq Pref(L(\mathcal{A}))$. By the inductive hypothesis $\alpha, \beta \in Dom(B_2)$ then, by Lemma 1 and Remark 1 one has $p < q_2 < t$, hence $\mathcal{B}_2$ is not complete by Theorem 1, a contradiction. If p does not exist or t does not exist (note that at least one of them exists), the proof is analogous. □

Thanks to the previous proposition, we can associate with each trim WDFA $\mathcal{A}$ a unique language that is the set $Dom(\mathcal{B})$ for any W-completion $\mathcal{B}$ of $\mathcal{A}$ and we denote it by $D_W(\mathcal{A})$ because it depends only on $\mathcal{A}$. Note that $D_W(\mathcal{A})$ is a Wheeler language. Our final aim is to find the unique minimal W-completion of $\mathcal{A}$ by minimizing the number of empty states. We define an equivalence for the words in $D_W(\mathcal{A}) \setminus Pref(L(\mathcal{A}))$.

Definition 5. Let $\mathcal{L} = D_W(\mathcal{A}) \setminus Pref(L(\mathcal{A}))$. In $\mathcal{L}$, we define the following equivalence relation. Let $u, v \in \mathcal{L}$,

$$u \equiv_{\mathcal{L}} v \Leftrightarrow \forall z \in \Sigma^*, uz \in \mathcal{L} \Leftrightarrow vz \in \mathcal{L}.$$

The input-consistent, convex refinement $\equiv^c_{\mathcal{L}}$ of $\equiv_{\mathcal{L}}$ is defined as follows:
Let $u, v \in \mathcal{L}$

$$u \equiv^c_{\mathcal{L}} v \Leftrightarrow u \equiv_{\mathcal{L}} v, end(u) = end(v), \forall z \in D_W(\mathcal{A}) : \min\{u, v\} \prec z \prec \max\{u, v\} \Rightarrow z \in \mathcal{L} \text{ and } z \equiv_{\mathcal{L}} u.$$

In fact, one can easily observe that all the words of the same class end with the same letter and each equivalence class of $\equiv^c_{\mathcal{L}}$ is a convex set of $D_W(\mathcal{A})$.

Definition 6. Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a trim WDFA and $\mathcal{B} = (Q \cup S, \Sigma, \delta', s, F)$ a W-completion of $\mathcal{A}$. In $\mathcal{L} = D_W(\mathcal{A}) \setminus Pref(L(\mathcal{A}))$ we define the following equivalence relation. Let $u, v \in \mathcal{L}$,

$$u \sim_{\mathcal{B}} v \Leftrightarrow \delta'(s, u) = \delta'(s, v).$$

Lemma 2. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a trim WDFA. For each W-completion $\mathcal{B} = (Q \cup S, \Sigma, \delta', s, F)$ of $\mathcal{A}$, $\sim_{\mathcal{B}}$ has a finite index and is a refinement of $\equiv^c_{\mathcal{L}}$.*

Proof. Since $\mathcal{B}$ is a DFA then $\sim_{\mathcal{B}}$ has a finite index equal to $|S|$. Moreover, if $u \sim_{\mathcal{B}} v$ then $end(u) = end(v)$ because $\mathcal{B}$ is input-consistent. If $z \in D_W(\mathcal{A})$ and $u \prec z \prec v$ then, by Remark 1, $\delta'(s, u) < \delta'(s, z) < \delta'(s, z)$ hence $\delta'(s, u) = \delta'(s, z) = \delta'(s, z)$. It follows that $u \equiv^c_{\mathcal{L}} v$. □

Theorem 2. *For each trim WDFA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ there exists a unique W-completion of $\mathcal{A}$ with a minimal number of states.*

Proof. First, note that $\equiv^c_{\mathcal{L}}$ is input-consistent and that each class is a convex set of $D_W(\mathcal{A})$. Furthermore, $\mathcal{L}$ is a union of classes of $\equiv_{\mathcal{L}}$, hence it is a union of classes of $\equiv^c_{\mathcal{L}}$. Moreover, $\equiv^c_{\mathcal{L}}$ has a finite index because for any other W-completion $\mathcal{B}'$, $\sim_{\mathcal{B}'}$ has a finite index and is a refinement of $\equiv^c_{\mathcal{L}}$ (by Lemma 2). Finally, $\equiv^c_{\mathcal{L}}$ is right-invariant. In fact, for any $u, v \in \mathcal{L}$ and $x \in \Sigma$ if $u \equiv^c_{\mathcal{L}} v$ then $u \equiv_{\mathcal{L}} v$ and then $ux \equiv_{D_W(\mathcal{A})} vx$. Trivially, $end(ux) = end(vx)$. If $ux \prec z \prec vx$ with $z \in D_W(\mathcal{A})$ then $end(z) = x$ i.e. $z = tx$ and $u \prec t \prec v$. By convexity, $u \equiv^c_{\mathcal{L}} t, u \equiv_{\mathcal{L}} t$ and $ux \equiv_{\mathcal{L}} tx$.

We build a WDFA $\mathcal{B}_1 = \mathcal{B}_{\equiv^c_{\mathcal{L}}} = (Q \cup S, s, \delta', F)$ as follows:

- $S = \{[u]_{\equiv^c_{\mathcal{L}}} \mid u \in \mathcal{L}\}$;
- if $p \in Q, u \in I_p$, and $\sigma \in \Sigma$ then $\delta'(p, \sigma) = \delta(p, \sigma)$ if $\delta(p, \sigma)$ is defined. Otherwise, if $u\sigma \in \mathcal{L}, \delta'(p, \sigma) = [u\sigma]$. Note that if $v \in I_p$ and $v \neq u$ then $v\sigma \equiv^c_{\mathcal{L}} u\sigma$.
- for any $\sigma \in \Sigma$ and $[u] \in S$ then $\delta'([u], \sigma) = [u\sigma]$, if $u\sigma \in \mathcal{L}$. Note that if $v \in [u]$ and $v \neq u$ then $v\sigma \equiv^c_{\mathcal{L}} u\sigma$.
- the order is an extension of the order in Q as follows. If $p \in Q, r \in S$ then $r < p$ iff $\forall u \in I_r$ and $v \in I_p, u < v$. If $s_1, s_2 \in S$ then $s_1 < s_2$ iff $\forall u \in s_1, v \in s_2, u < v$.

Hence $\mathcal{B}_1$ is a WDFA. By construction, $\mathcal{A} \subseteq \mathcal{B}_1$ and $\mathcal{B}_1$ is a W-completion of $\mathcal{A}$. Any other W-completion $\mathcal{B}_2$ of $\mathcal{A}$ induces a relation $\sim_{\mathcal{B}_2}$ on $\mathcal{L}$ that by Lemma 2 is a refinement of $\equiv^c_{\mathcal{L}}$, hence $\mathcal{B}_2$ has a number of empty states greater than $\mathcal{B}_1$ does. □

The previous theorem naturally leads to the following definition.

Definition 7. The W-completion of $\mathcal{A}$ with a minimal number of states is called *the minimal Wheeler completion of $\mathcal{A}$* and is denoted by $C_W(\mathcal{A})$.

Theorem 3. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a trim WDFA and $\mathcal{B} = (Q \cup S, \Sigma, \delta', s, F)$ a W-completion of $\mathcal{A}$. Then $\mathcal{B}$ is minimal iff for any $p, q \in S$ consecutive empty states $In(p) \neq In(q)$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a trim WDFA and $\mathcal{B} = (Q \cup S, \Sigma, \delta', s, F)$ a W-completion of $\mathcal{A}$. By Theorem 2, $\mathcal{B}$ is minimal iff for any $p, q \in S$ or (i) there exists $z \in \Sigma^*$ such that $\delta'(p, z) \in S$ and $\delta'(q, z) \notin S$ (or viceversa), or (ii) $In(p) \neq In(q)$ or (iii) there exists $t \in Q \cup S$ such that $p < t < q$. If assumption (i) were true, since q is an empty state then $\delta'(q, z) \notin Q$ hence $\delta'(q, z) = \square$. Consequently, according to Theorem 1, this would imply that $\mathcal{B}$ is not W-complete leading to a contradiction. Then if p and q are consecutive, then $In(p) \neq In(q)$. □

Corollary 1. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a trim WDFA and $\mathcal{B} = (Q \cup S, \Sigma, \delta', s, F)$ a W-completion of $\mathcal{A}$. Then $\mathcal{B}$ is minimal iff for any $p, q \in Q \cup S$ consecutive states and $\sigma \in \Sigma$, if $\delta'(p, \sigma), \delta'(q, \sigma) \notin Q$ then $\delta'(q, \sigma) = \delta'(p, \sigma)$.*

Here we give an upper bound on the number of states of the W-completion automaton.

Theorem 4. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F, \leq)$ a trim WDFA with n states and over a k letter alphabet. The W-completion $C_W(\mathcal{A})$ has at most $2n + k - 2$ states.*

Proof. Let $C_W(\mathcal{A}) = (Q \cup S, \Sigma, \delta', s, F)$. Each state of Q can be followed by a state of S. Morevover, if an empty state $p \in S$ is followed by an empty state $q \in S$, by Theorem 3, $In(p) = \sigma$ and $In(q) = \tau$, with $\sigma \neq \tau$. According to the Wheeler property, $\sigma < \tau, \sigma$ and τ are consecutive letters and $p = max(R_\sigma)$ and $q = min(R_\tau)$. Furthermore, for each $\tau, min(R_\tau) = \delta'(s, \tau)$ therefore, there exists at least a letter τ such that $min(R_\tau)$ is a coaccessible state. Therefore, an empty state p can be followed by an empty state q at most $k - 2$ times. In conclusion $|Q \cup S| = |Q| + |S| = n + n + k - 2$. □

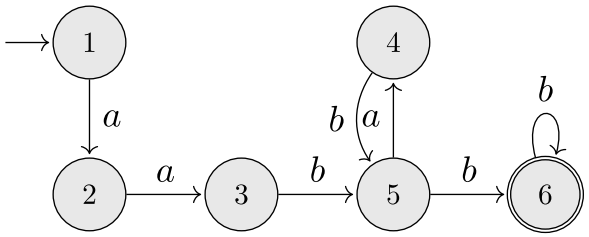


Fig. 3. The Wheeler automaton of the Example 4.

5. An algorithm for the W-completion of a Wheeler DFA

In this section we provide an algorithm that takes as input a trim WDFA $\mathcal{A}$ and computes its Wheeler completion, $C_W(\mathcal{A})$. The elements of Q are denoted by integers $\{1, 2, \dots n\}$ and the elements of S, the new empty states added, are named using rational non-integer numbers for the purpose of easily specifying their order relative to the already existing states. Let $l, r \in Q$ with $l < r$, by $H_\sigma(l, r)$ we denote the *internal interval of σ-holes* that is

$$H_\sigma(l, r) = \{q \in Q \cup S \mid l < q < r \text{ and } \delta(q, \sigma) = \square\}.$$

Remark that, in our notation, the intervals $H_\sigma(l, r)$ do not contain the endpoints l and r of the interval. Furthermore, we denote by $H_\sigma(*, r)$ and $H_\sigma(l, *)$ the *left interval of σ-holes* and *right interval of missing σ-holes*, respectively. That is

$$H_\sigma(*, r) = \{q \in Q \cup S \mid q < r \text{ and } \delta(q, \sigma) = \square\} \text{ and } H_\sigma(l, *) = \{q \in Q \cup S \mid q > l \text{ and } \delta(q, \sigma) = \square\}.$$

Example 4. Consider the WDFA in Fig. 3, the transition function is defined by the following transformations:

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & \square & \square & 4 & \square \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \square & \square & 5 & 5 & 6 & 6 \end{pmatrix}.$$

Then $R_a = (2, 3, 4)$ and $R_b = (5, 5, 6, 6)$ and the intervals of missing a-holes are $H_a(2, 5) = \{3, 4\}$ and $H_a(5, *) = \{6\}$. The interval of b-holes is $H_b(*, 3) = \{1, 2\}$.

The algorithm works by adding empty states and transitions to fill as many intervals of holes as possible while maintaining the Wheeler property.

An internal interval of holes $H_\sigma(i, j)$, with $\sigma \in \Sigma$ and $1 \le i \le j \le n$, is said to be *blocking* if $\delta_\sigma(i)$ and $\delta_\sigma(j)$ are equal integers. All other intervals of holes are *non-blocking*. By Theorem 1, an automaton is complete if and only if all the intervals of holes (if any) are blocking.

Example 5. The automaton $\mathcal{A}$ of Example 3 has a blocking interval of holes, which is $H_b(s, p)$. But it is not W-complete because it also has a non-blocking interval of holes, which is $H(*, p)$.

In the execution of the algorithm we will deal with WDFA $\mathcal{A} = (Q \cup S, \Sigma, \delta, 1, F)$ where S is the set of empty states that initially is empty and δ is the transition function. Both S and δ are updated at each iteration of the while loop. We denote by $H(\delta, Q \cup S)$ the set of non-blocking intervals of holes, with the current S and δ. Initially S is empty and δ is the transition function of $\mathcal{A}$.

The algorithm terminates when $H(\delta, Q \cup S)$ is empty (line 2). The 'for loops' at lines 3, 13 and 20, fill all the holes of the internal, left and right intervals of σ-holes extracted from $H(\delta, Q \cup S)$, respectively, and temporarily collect the new empty states, eventually created, in a set R. When all the current non-blocking intervals have been filled, from each state of R a missing transition is added (line 27) and the elements of R are added to S, finally, the new $H(\delta, Q \cup S)$ is computed.

Now we detail the filling of the intervals. Let $H_\sigma(l, r)$ be the extracted interval of σ-holes (line 4-12). If $\delta_\sigma(l) \in Q$ and $\delta_\sigma(r) \in Q$, i.e. are different and consecutive integers, all the $\square$ in between are filled with the new empty state $e = \delta_\sigma(l) + 0.5$, to guarantee the Wheeler property of R_σ, and e is added to R. Otherwise, if one of $\delta_\sigma(l)$ and $\delta_\sigma(r)$ is an empty state, then all the $\square$ in between are filled with it. This guarantees the minimality of the W-completion (cf. Corollary 1) and guarantees that the algorithm terminates.

When a left interval of σ-holes $H_\sigma(*, r)$ (lines 14-19) is considered, if $\delta_\sigma(r) \in Q$, i.e. is an integer, all the $\square$ that precede are filled with the new empty state $e = \delta_\sigma(r) - 0.3$ and e is added to R. Such a position guarantees that $max(R_\tau) < min(R_\sigma)$ when τ and σ are consecutive letters. Otherwise, if $\delta_\sigma(r)$ is an empty state, then all the preceding $\square$ are filled with it.

Finally, if $H_\sigma(l, *)$ is extracted (lines 21-26), if $\delta_\sigma(l) \in Q$, i.e. is an integer, all the $\square$ that follow are filled with the new empty state $e = \delta_\sigma(r) + 0.5$, and e is added to R. Such a position guarantees that $max(R_\sigma) < min(R_\tau)$ if σ and τ are consecutive letters. Otherwise, if $\delta_\sigma(l)$ is an empty state, then all the subsequent $\square$ are filled with it.

The alphabet Σ, the set Q of coaccessible states and the set of final states F remain unchanged.

Theorem 5. *Let A be a trim WDFA. Algorithm 1 computes the minimal W-completion of $\mathcal{A}$.*

Proof. The algorithm produces a WDFA that is W-complete because it does not have any blocking interval of holes. It contains $\mathcal{A}$ because it joins to $\mathcal{A}$ some empty states and transitions involving only empty states; hence it preserves the trim part of $\mathcal{A}$. Finally, it is minimal because it verifies Corollary 1. □

Algorithm 1 Compute $C_W(\mathcal{A})$.

$S \leftarrow \emptyset$
while $(H(\delta, Q \cup S) \neq \emptyset)$ **do**
  for all $(H_\sigma(l,r) \in H(\delta, Q \cup S))$ **do**
    if $(\delta_\sigma(l) \in Q$ and $\delta_\sigma(r) \in Q)$ **then**
      $e = \delta_\sigma(l) + 0.5$
      for all $(p \in H_\sigma(l,r))$ **do** $\delta_\sigma(p) = e$
      $R \leftarrow R \cup \{e\}$
    else
      if $(\delta_\sigma(l) \in S)$ **then**
        for all $(p \in H_\sigma(l,r))$ **do** $\delta_\sigma(p) = \delta_\sigma(l)$
      else
        for all $(p \in H_\sigma(l,r))$ **do** $\delta_\sigma(p) = \delta_\sigma(r)$
  for all $(H_\sigma(*,r) \in H(\delta, Q \cup S))$ **do**
    if $\delta_\sigma(r) \in Q$ **then**
      $e = \delta_\sigma(r) - 0.3$
      for all $(p \in H_\sigma(*,r))$ **do** $\delta_\sigma(p) = e$
      $R \leftarrow R \cup \{e\}$
    else
      for all $(p \in H_\sigma(*,r))$ **do** $\delta_\sigma(p) = \delta_\sigma(r)$
  for all $(H_\sigma(l,*) \in H(\delta, Q \cup S))$ **do**
    if $\delta_\sigma(l) \in Q$ **then**
      $e = \delta_\sigma(l) + 0.5$
      for all $(p \in H_\sigma(l,*))$ **do** $\delta_\sigma(p) = e$
      $R \leftarrow R \cup \{e\}$
    else
      for all $(p \in H_\sigma(l,*))$ **do** $\delta_\sigma(p) = \delta_\sigma(l)$
  for all $(r \in R$ and $\gamma \in \Sigma)$ **do** $\delta_\gamma(r) = \square$
  $S \leftarrow S \cup R$
  $R \leftarrow \emptyset$
  COMPUTE$H(\delta, Q \cup S)$

Running time. Given a fixed alphabet of size k, which is considered constant, we analyze the running time of the algorithm with respect to the number n of states. By Theorem 4, the algorithm adds at most $n + k - 2$ states. This implies that, during execution, the number of possible non-blocking intervals that lead to the creation of a new empty state is upper bounded by $n + k - 2$. Therefore, there are $\mathcal{O}(n)$ iterations in which $H(\delta, Q \cup S)$ is nonempty, i.e., the main **while** iteration is executed $\mathcal{O}(n)$ times. The **for** loops at lines 3, 13, and 20 are executed, in total, as many times as the number of intervals of missing transitions at each iteration. This number is trivially upper bounded by a linear function of n. The **for** loops at lines 6, 10, 12, 16, 19, 23 and 26 are executed, in total, as many times as the number of holes, which is also upper bounded by a linear function of n. Finally, COMPUTE$H(\delta, Q \cup S)$ has linear time complexity. In conclusion, the algorithm presented here runs in time $\mathcal{O}(n^3)$.

Example 6. Let $\mathcal{A} = (Q, \Sigma, \delta, 1, F, \leq)$ with $Q = \{1, 2, \ldots, 6\}$ depicted in Fig. 3 and transition functions:

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & \square & \square & 4 & \square \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \square & \square & 5 & 5 & 6 & 6 \end{pmatrix}.$$

By filling $H_a(2,5)$, $H_a(5,*)$ and $H_b(*,3)$ three empty states (3.5, 4.5 and 4.7) are added and the transition function is updated as follows:

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 4.7 & 5 & 6 \\ 2 & 3 & 3.5 & \square & 3.5 & \square & \square & 4 & 4.5 \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 4.7 & 5 & 6 \\ 4.7 & 4.7 & 5 & \square & 5 & \square & \square & 6 & 6 \end{pmatrix}.$$

By filling $H_b(4,5)$ one more empty state is added (the empty state 5.5) and by filling $H_a(3,4)$ and $H_a(4,5)$ no empty state is created. The transition function is updated as follows:

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 4.7 & 5 & 5.5 & 6 \\ 2 & 3 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 4 & \square & 4.5 \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 4.7 & 5 & 5.5 & 6 \\ 4.7 & 4.7 & 5 & \square & 5 & 5.5 & 5.5 & 6 & \square & 6 \end{pmatrix}.$$

By filling the last non-blocking interval, we get the following W-complete automaton depicted in Fig. 4. The empty states and new transitions are dashed.

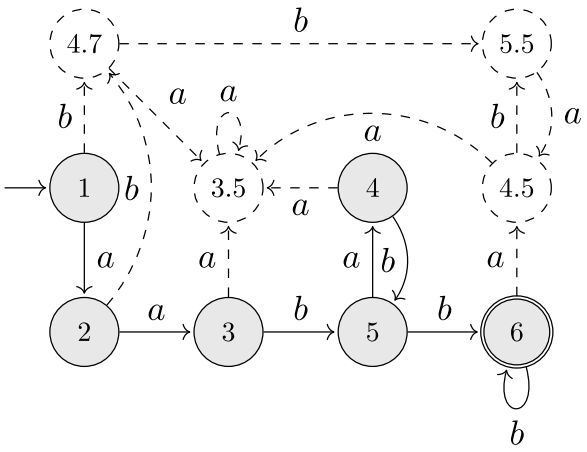


Fig. 4. The W-completion of the automaton in Fig. 3.

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 4.7 & 5 & 5.5 & 6 \\ 2 & 3 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 4 & 4.5 & 4.5 \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 4.7 & 5 & 5.5 & 6 \\ 4.7 & 4.7 & 5 & \square & 5 & 5.5 & 5.5 & 6 & \square & 6 \end{pmatrix}.$$

A W-complete automaton is, in general, not complete. See, for example, the WDFA in 2(b) is the W-completion of the automaton in Fig. 2(a), but it is not complete. We say that a Wheeler automaton $\mathcal{A}$ is *completable* if $C_W(\mathcal{A})$ is complete.

Note that, as shown in the following example, for the same Wheeler language, there can exist a WDFA that is completable and another equivalent WDFA that is not.

Example 7. Let $\mathcal{A}$ a WDFA with final state 4 and transition function as follows.

$$\mathcal{A} : \delta_a = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & \square & \square \end{pmatrix}, \delta_b = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & \square & 4 & \square \end{pmatrix}$$

and $\mathcal{A}'$ a WDFA with final states 4 and 5 and transition function as follows.

$$\mathcal{A}' : \delta'_a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & \square & \square & \square \end{pmatrix}, \delta'_b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & \square & 5 & \square & \square \end{pmatrix}$$

Both $\mathcal{A}$ and $\mathcal{B}$ recognize the finite language $L = \{b, aab\}$ but $\mathcal{A}$ is not completable while $\mathcal{A}'$ is. Indeed, the respective minimal W-completions are:

$$C_W(\mathcal{A}) : \delta_a = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 \\ 2 & 3 & 3.5 & 3.5 & 3.5 & 3.5 \end{pmatrix}, \delta_b = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 \\ 4 & \square & 4 & 4.5 & 4.5 & 4.5 \end{pmatrix}$$

$$C_W(\mathcal{A}') : \delta'_a = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 5 & 5.5 \\ 2 & 3 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 \end{pmatrix}, \delta'_b = \begin{pmatrix} 1 & 2 & 3 & 3.5 & 4 & 4.5 & 5 & 5.5 \\ 4 & 4.5 & 5 & 5.5 & 5.5 & 5.5 & 5.5 & 5.5 \end{pmatrix}$$

Remark that $C_W(\mathcal{A})$ is complete iff $D_W(\mathcal{A}) = \Sigma^*$, this fact is important for defining the operations on Wheeler automata described in the next section.

6. Operations on Wheeler automata

We start this section by recalling some basic constructions in theory of automata.

Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a DFA and let $L(\mathcal{A})$ be the language recognized by $\mathcal{A}$. Let $\mathcal{A}_c = (Q, \Sigma, \delta, s, Q \setminus F)$. If $\mathcal{A}$ is a complete DFA then $\mathcal{A}_c$ recognizes the complement of $L(\mathcal{A})$, i.e. $L(\mathcal{A}_c) = \Sigma^* \setminus L(\mathcal{A})$.

Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be two DFAs over the same alphabet Σ, recognizing, respectively, the languages $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$. The *Cartesian product* of $\mathcal{A}_1$ and $\mathcal{A}_2$ is the DFA $\mathcal{A}_1 \times \mathcal{A}_2 = (Q, \Sigma, \delta, s, F)$, where:

- $Q = Q_1 \times Q_2$,
- $s = (s_1, s_2)$,
- $\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$, with $(q_1, q_2) \in Q$ and $\sigma \in \Sigma$.

If $F = F_1 \times F_2$ then $\mathcal{A}$ recognizes the intersection of $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$, i.e $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. Whereas, if $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$ and $\mathcal{A}_1$ and $\mathcal{A}_2$ are complete DFAs, then $\mathcal{A}$ recognizes the union of $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$, i.e $L(\mathcal{A}) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$.

Remark 2. Note that the completeness hypothesis for $\mathcal{A}_1$ and $\mathcal{A}_2$ guarantees that the constructions give an automaton for the complement and union, respectively.

In the case of Wheeler languages, we are dealing with automata that are not, in general, complete; therefore, the above constructions could fail for WDFAs. Indeed, the class of Wheeler languages is closed under intersection, but it is not closed under union and complementation.

In the following, we give a procedure for complementation and a procedure for the union of Wheeler languages. The basic idea in both constructions is the following: First, apply to the input WDFA the completion algorithm given in the previous section; then apply to the output of the completion algorithm the classical constructions for the complement and the union.

If the W-completion is a complete WDFA, we are able to construct WDFAs both for the complement and for the union. If not, some special cases are considered.

6.1. The Complement construction

Let $\mathcal{A} = (Q, \Sigma, \delta, 1, F)$ a WDFA. We compute the W-completion $C_W(\mathcal{A}) = (Q \cup S, \delta', s, F)$. We then construct the automaton $\mathcal{A}_c = (Q \cup S, \Sigma, \delta', s, Q \cup S \setminus F)$. The language $L(\mathcal{A}_c)$ is a Wheeler language and is such that

$$L(\mathcal{A}_c) = D_W(\mathcal{A}) \setminus L(\mathcal{A}).$$

If $C_W(\mathcal{A})$ is complete, then $L(\mathcal{A}_c) = L(\mathcal{A})^c$; therefore, we can state the following proposition.

Proposition 2. *Let $\mathcal{A}$ be a trim WDFA. If $\mathcal{A}$ is completable, then $L(\mathcal{A})^c$ is a Wheeler language.*

Example 8. Let us consider the transition function of the Wheeler automaton $\mathcal{A}$ in Fig. 1(a) recognizing the language a^+b

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & \square \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 \\ \square & 3 & \square \end{pmatrix}.$$

The minimal W-completion $C_W(\mathcal{A})$ is the following:

$$\delta'_a = \begin{pmatrix} 1 & 2 & 2.5 & 2.7 & 3 & 3.5 \\ 2 & 2 & 2.5 & 2.5 & 2.5 & 2.5 \end{pmatrix} \quad \delta'_b = \begin{pmatrix} 1 & 2 & 2.5 & 2.7 & 3 & 3.5 \\ 2.7 & 3 & 3.5 & 3.5 & 3.5 & 3.5 \end{pmatrix}.$$

It is the complete automaton ($D_W(\mathcal{A}) = \Sigma^*$) in Fig. 1(c) with, $s = 1, p = 2, t = 2.5, r = 2.7, q = 3$ and $w = 3.7$. Hence, the complement of the Wheeler language a^+b is a Wheeler language.

If $C_W(\mathcal{A})$ is not complete (i.e. $\mathcal{A}$ is not completable), we have $L(\mathcal{A}_c) = D_W(\mathcal{A}) \setminus L(\mathcal{A})$ and then the complement of $L(\mathcal{A})$, with respect to $D_W(\mathcal{A})$, is a Wheeler language. Remark that this result extends the one stated in Lemma 5.1, point 5, of [1], where the Wheelerness of $Pref(L(\mathcal{A})) \setminus L(\mathcal{A})$ is considered.

Example 9. Let us consider the transition function of the Wheeler automaton in Fig. 2(a) recognizing the language b^+a

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 \\ \square & \square & 2 \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 \\ 3 & \square & 3 \end{pmatrix}.$$

It cannot be completed (cf. Example 2) but it has the following minimal W-completion.

$$\delta'_a = \begin{pmatrix} 1 & 1.7 & 2 & 3 \\ 1.7 & 1.7 & 1.7 & 2 \end{pmatrix} \quad \delta'_b = \begin{pmatrix} 1 & 1.7 & 2 & 3 \\ 3 & \square & \square & 3 \end{pmatrix}.$$

It is the automaton in Fig. 2(b) with $s = 1, r = 1.7, q = 2$ and $p = 3$.

It is known that the Wheeler language b^+a is not recognized by any complete WDFA, hence any WDFA that recognizes it is not completable. On the other hand, it can occur that an automaton $\mathcal{A}$ is not completable but recognizes a Wheeler language whose complement is a Wheeler language, as shown in the following example.

Example 10. Consider the language $aab + b$. It is a Wheeler language because it is finite and its complement is a Wheeler language because it is cofinite. The following is the transition function of a Wheeler automaton that recognizes it and is not completable

$$\delta_a = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & \square & \square \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & \square & 4 & \square \end{pmatrix}.$$

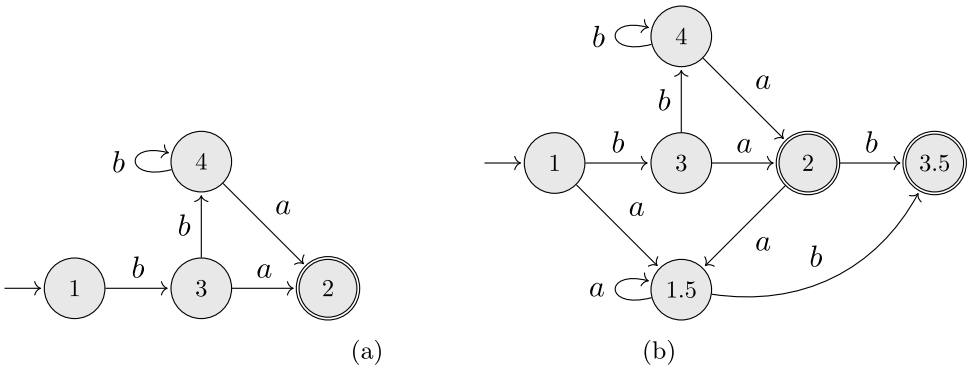


Fig. 5. (a) An incomplete Wheeler automaton accepting b^+a and (b) its W-completion.

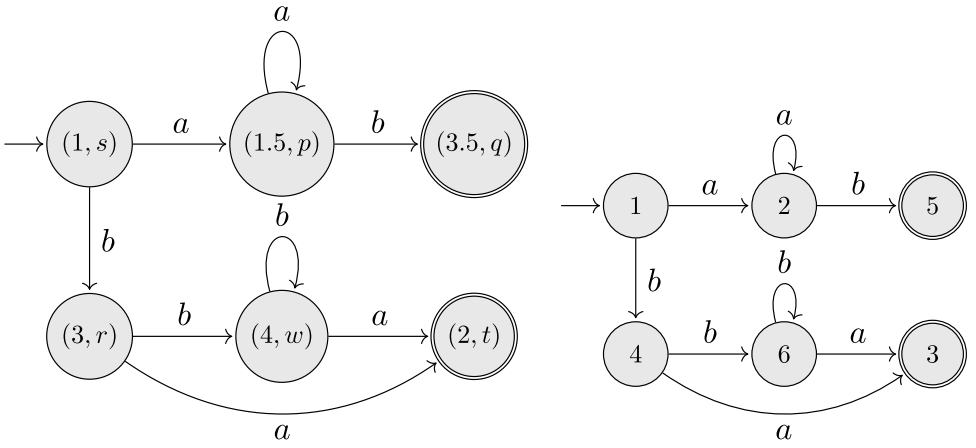


Fig. 6. A Wheeler automaton for $a^+b \cup b^+a$.

6.2. The Union construction

Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be two WDFAs over the same alphabet Σ, recognizing respectively the languages $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$.

We first construct the minimal W-completion $C_W(\mathcal{A}_1)$ of the automaton $\mathcal{A}_1$ and the minimal W-completion $C_W(\mathcal{A}_2)$ of the automaton $\mathcal{A}_2$.

Let $Q_1' = Q_1 \cup S_1$ the set of states of $C_W(\mathcal{A}_1)$ and $Q_2' = Q_2 \cup S_2$ the set of states of $C_W(\mathcal{A}_2)$.

We construct the automaton $C_W(\mathcal{A}_1) \times C_W(\mathcal{A}_2)$, as in the classical way. More precisely, we consider only the accessible and coaccessible part of $C_W(A_1) \times C_W(A_2)$ i.e. we consider accessible pairs (p, q) such that at least one of the two states is coaccessible. Moreover, we choose as the set of final states the set of accessible states of $(F_1 \times Q_2') \cup (Q_1' \times F_2)$. The construction uses the computation of the minimal W-completion, and a Cartesian product thus runs in $\mathcal{O}(n^3)$.

If $C_W(\mathcal{A}_1) \times C_W(\mathcal{A}_2)$ does not contain any pair with a component $\square$, it is a WDFA since, given two states (q_1, q_2) and (p_1, p_2) of $C_W(\mathcal{A}_1) \times C_W(\mathcal{A}_2)$ we have $q_1 \leq p_1 \iff q_2 \leq p_2$, since the co-lexicographic order over the words corresponds to the total order between the states.

We can give the following proposition.

Proposition 3. *Let $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be two WDFAs over the same alphabet Σ. If $L(\mathcal{A}_1) \subseteq D_W(A_2)$ and $L(\mathcal{A}_2) \subseteq D_W(A_1)$ then $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$ is a Wheeler language.*

Proof. Since $L(\mathcal{A}_1) \subseteq D_W(A_2)$ it follows that $Pref(L(\mathcal{A}_1)) \subseteq D_W(A_2)$ and consider the construction by the Cartesian product defined before. Consider an accessible pair (p, q) in the automaton $C_W(A_1) \times C_W(A_2)$. By definition, either p or q is coaccessible. If p is coaccessible then $q \neq \square$ because $Pref(L(\mathcal{A}_1)) \subseteq D_W(A_2)$, then the set of states of the Cartesian product does not contain any pair of type $(p, \square)$. Analogously, from $L(\mathcal{A}_2) \subseteq D_W(A_1)$ it follows that the set of states of the Cartesian product does not contain any pair of type $(\square, q)$. Hence, the set of pairs can be totally ordered, i.e. the Cartesian product is a WDFA that recognizes $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$. □

Example 11. Let $L = a^+b + b^+a$. The automaton $\mathcal{A}_1$ in Fig. 1(a) recognizes a^+b and the automaton in Fig. 1(c) is its W-completion $C_W(\mathcal{A}_1)$. The automaton $\mathcal{A}$ in Fig. 2 recognizes the language b^+a but $a^+b \not\subset D_W(\mathcal{A})$. Consider the automaton $\mathcal{A}_2$ in Fig. 5(a) and its

W-completion in Fig. 5(b). We have $a^+b \subseteq D_W(\mathcal{A}_2)$. As Fig. 6 shows, the union procedure in this case gives an automaton that contains only comparable pairs, hence it is a Wheeler automaton that recognizes $a^+b \cup b^+a$.

Propositions 2 and 3 provide only sufficient conditions. An interesting open problem is to find necessary and sufficient conditions for the Wheelerness of complement and union of Wheeler languages.

CRediT authorship contribution statement

Giuseppa Castiglione: Writing – original draft; **Antonio Restivo:** Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Alanko, G. D'agostino, A. Policriti, N. Prezza, 281, Wheeler languages, 2021.
[2] J.J. Ashley, B.H. Marcus, D. Perrin, S. Tuncel, Surjective extensions of sliding-block codes, SIAM J. Discret. Math 6 (4) (1993) 582–611.
[3] J.A. Brzozowski, B. Li, D. Liu, Syntactic complexities of six classes of star-free languages, J. Autom. Lang. Comb 17 (2-4) (2012) 83–105.
[4] M.P. Béal, S. Lombardy, D. Perrin, Embeddings of local automata, Illinois Journal of Mathematics 54 (1) (2010) 155–174.
[5] N. Cotumaccio, G. D'agostino, A. Policriti, N. Prezza, Co-lexicographically ordering automata and regular languages - part I, J. ACM 70 (4) (2023) 73.
[6] N. Cotumaccio, N. Prezza, On indexing and compressing finite automata, in: D. Marx (Ed.), Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, SIAM, 2021, pp. 2585–2599.
[7] D. Agostino, G. Martincigh, D. Policriti, A, Ordering regular languages and automata, Complexity. Theor. Comput. Sci 949 (2023) 113709.
[8] L. Egidi, F.A. Louza, G. Manzini, Space efficient merging of de bruijn graphs and wheeler graphs, Algorithmica 84 (3) (2022) 639–669.
[9] A. Ehrenfeucht, G. Rozenberg, Each regular code is included in A maximal regular code, RAIRO Theor. Informatics Appl 20 (1) (1986) 89–96.
[10] T. Gagie, G. Manzini, J. Sirén, Wheeler graphs: A framework for bwt-based data structures, Theor. Comput. Sci 698 (2017) 67–78.
[11] D. Gibney, S.V. Thankachan, On the complexity of recognizing wheeler graphs, Algorithmica 84 (3) (2022) 784–814.
[12] R. Montalbano, Proceedings. Lecture Notes in Computer Science, Local automata and completion, 665 of *Würzburg, Germany*, Springer, 1993.
[13] M.P. Schützenberger, A remark on incompletely specified automata, Inf. Control 8 (4) (1965) 373–376.