Completing Wheeler Automata*

Giuseppa Castiglione[1], Antonio Restivo[1]

[1]Dipartimento di Matematica e Informatica, Università degli studi di Palermo, Palermo, Italy.

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 WDFA containing $\mathcal{A}$: it is called the 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.

Keywords: Wheeler Automata, Complete Automata, Boolean Operations.

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. [1], [2], [3], [4], [5]). It is referred as the completion problem.

In this paper we approach the completion problem for the class of Wheeler automata, that has been recently introduced in [6]. 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, [7], [8]). The regular languages that can be accepted by a Wheeler automaton are called *Wheeler languages* whose study is deepened in [9], [10] and [11]. The completion problem is of particular interest for the class of Wheeler Deterministic Finite Automata (WDFA) 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 WDFA, ii) the (total) order on the states of $C(\mathcal{A})$ is an extension of the order on the state 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 an any equivalent complete WDFA preserving the order of the states. We show that, in any case, there exists a "maximal" WDFA in which $\mathcal{A}$ can be embedded. It is maximal in the sense that does not exist another WDFA containing it. We prove that, among all the maximal WDFA containing $\mathcal{A}$,

ICTCS 2024: Italian Conference on Theoretical Computer Science, September 11-13, 2024, Torino, Italy

* Partially supported by INdAM-GNCS Project 2024- CUP E53C23001670001 and FFR fund University of Palermo.

giuseppa.castiglione@unipa.it (G. Castiglione); antonio.restivo@unipa.it (A. Restivo)

the one with the minimal number of states is unique. We call it the Wheeler completion of $\mathcal{A}$ 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 Wheeler completion $C_W(\mathcal{A})$. In the case $C_W(\mathcal{A})$ is a complete WDFA we say that $\mathcal{A}$ is completable.

We further consider some relevant applications of this completion algorithm. In fact, there are some important constructions in automata theory that require the automata to be complete. This is the case of boolean operations. According to the fact that WDFAs are not in general complete, the family of Wheeler languages is closed under intersection, but it is not closed neither under complementation nor under union (cf. [9]).

In the second part of the paper we use the completion algorithm to construct, under suitable condistions, a WDFA that recognizes the complement of a Wheeler language and a WDFA that recognizes the union of two Wheeler languages. This approach is alternative to the one proposed in [12].

2. Preliminaries and notations

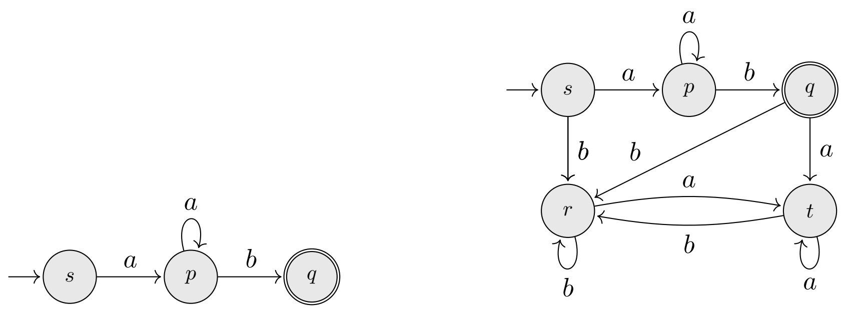
If Σ is a finite alphabet, with Σ^* we denote the set of finite words on Σ. If $L \subseteq \Sigma^*$, with $Pref(L)$ we denote the set of all prefixes of words in L, $Pref(L) = \{v \in \Sigma^* |\ \exists u \in \Sigma^* \text{ s.t. } vu \in L\}$. A *deterministic finite automaton* (DFA) is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ 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* whereas 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 we say that $\delta(p, \sigma)$ is a *missing transition*.

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

Let $(X, \leq)$ a total order on X, for any $x, y \in X$ we write $x < y$ if $x \leq y$ and $x \neq y$. Two elements $x, y \in X$ are *consecutive* if $x < y$ and does not exist any $z \in X$ such that $x < z < y$. If $X \subseteq Y$ we say that we extend the order $\leq$ on Y if we define a total order $\leq'$ on Y such that $\leq$ is the restriction on X of $\leq'$. In what follows we will say that $(X, \leq)$ is a restriction of $(Y, \leq')$ and $(Y, \leq')$ is an extension of $(X, \leq)$.

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 $v_1 = \delta(u_1, \sigma_1)$ and $v_2 = \delta(u_2, \sigma_2)$, $u_1, u_2, v_1, v_2 \in Q$ and $\sigma_1, \sigma_2 \in \Sigma$;
  if $\sigma_1 < \sigma_2$ then $v_1 < v_2$;
  if $\sigma_1 = \sigma_2$ and $u_1 \leq u_2$ then $v_1 \leq v_2$;



(a) (b)

Figure 1: (a) A Wheeler automaton and (b) an equivalent complete automaton that is not Wheeler.

As a consequence, if $\sigma_1 < \sigma_2$ then, for all $p, q \in Q$, if $In(p) = \{\sigma_1\}$ and $In(q) = \{\sigma_2\}$ then $p < q$. We say that a regular language is a *Wheeler language* if it is recognizable by a WDFA. It is well known that Wheeler languages are star-free languages (cf. [9]). 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 is quite natural to add some sink states while maintaining the Wheeler property regarding the order. For this reason, in what follows, we assume that a WDFA can also have non-coaccessible states. Note that, by allowing for non-coaccessible states the automaton remains well-defined.

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

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

3. Completion of Wheeler Automata.

Here we face the problem of completing a Wheeler automaton $\mathcal{A}$ by an extension of the original order and preserving the language recognized by the automaton. Given a WDFA $\mathcal{A}$, the goal is to find, when it exists, an equivalent complete Wheeler automaton $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B}$ and the order of states of $\mathcal{B}$ is an extension of the one in $\mathcal{A}$. First, recall that any incomplete DFA can be completed by adding an empty state to which all the missing transitions converge. Such an operation does not ensure that we obtain a Wheeler automaton, as showed in the following example.

Example 1. *Let us consider the automaton in Figure 1(a), it is a WDFA (with $a < b$ and $s < p < q$) that accepts the language a^+b and it is not complete. By adding a single sink state for each letter as in Figure 1(b) we get the minimal input-consistent complete equivalent automaton. Note that it is not a WDFA indeed if $r < q$ then $q < p$, a contradiction, because $In(p) = \{a\}$ and $In(q) = \{b\}$. Whereas, if $r > q$ then $s > p$, a contradiction because s is the minimum. In Figure 2 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 sink states have been added.*

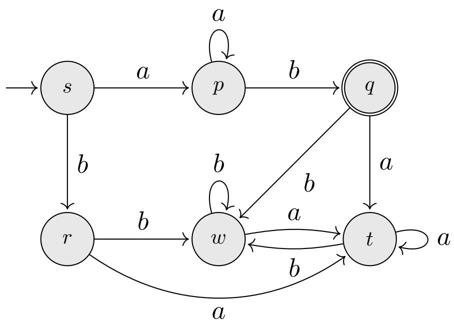


Figure 2: A complete Wheeler automaton, with $s < p < t < r < q < w$, accepting a^+b.

The previous example emphasizes two issues. On one hand, the classical procedure of completing an automaton by adding only one sink state (or one for each letter, for the input-consistency) could produce a DFA that is no more a WDFA. On the other hand, the automaton can, in some cases, be completed by adding more than one empty state in order to preserve the Wheeler property. 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 Figure 3(a) is a Wheeler automaton (with $a < b$ and $s < q < p$) that accepts the language b^+a. In [9] the authors give such a language as an example of Wheeler language for which there is not any complete Wheeler automaton that recognizes it. To prove this fact they use some properties of the co-lexicographic order on $Pref(L(\mathcal{A}))$.*

Given a WDFA $\mathcal{A}$, although there does not always exist a complete WDFA that contains $\mathcal{A}$, nevertheless among the WDFAs that contain $\mathcal{A}$ there always exists a maximal one. This leads to the following definition.

Definition 1. *Let $\mathcal{A}$ be a Wheeler automaton. $\mathcal{A}$ is Wheeler-complete (shortly W-complete) if for any equivalent Wheeler automaton $\mathcal{B}$ if $\mathcal{A} \subseteq \mathcal{B}$ then $\mathcal{A} = \mathcal{B}$.*

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

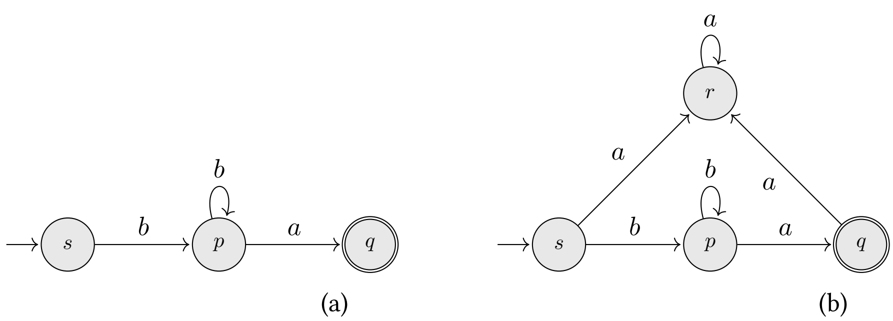


Figure 3: (a) An incomplete Wheeler automaton accepting b^+a and (b) its (incomplete) W-completion.

Theorem 1. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ be a WDFA. $\mathcal{A}$ is W-complete iff for any missing transition $\delta(q, \sigma)$, 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.*

Let $\mathcal{A}$ be a WDFA. Denote by $Q_{\mathcal{A}}$ the set of states of $\mathcal{A}$ and suppose that all the elements of $Q_{\mathcal{A}}$ are coaccessible. Let $\mathcal{B} = (Q, \Sigma, \delta, s, F)$ be a W-complete WDFA such that $\mathcal{A} \subseteq \mathcal{B}$. One has that $Q = Q_{\mathcal{A}} \cup S$, where S is the set of sink states of $\mathcal{B}$. We say that two states p, q are *consecutive in* $Q_{\mathcal{A}}$ (resp. in $Q_{\mathcal{A}} \cup S$) if $p < q$ and does not exist any $t \in Q_{\mathcal{A}}$ (resp. $t \in Q_{\mathcal{A}} \cup S$) such that $p < t < q$.

We say that $\mathcal{B}$ *has a minimal number of sink states*, if for any other W-complete WDFA $\mathcal{B}'$ containing $\mathcal{A}$, $\mathcal{B}'$ has a number of states greater than or equal to that of $\mathcal{B}$.

Lemma 1. *Let $\mathcal{A}$ be a WDFA with $Q_{\mathcal{A}} = \{q_1, q_2, \ldots, q_n\}$ all coaccesible states. Let $\mathcal{B} = (Q_{\mathcal{A}} \cup S, \Sigma, \delta, s, F)$ be a W-complete DFA such that $\mathcal{A} \subseteq \mathcal{B}$ with the minimal number of sink states. Two sink states $p, q \in S$ are consecutive in $Q_{\mathcal{A}} \cup S$ iff there exist two consecutive letters $\sigma < \tau$ such that $\delta(q_n, \sigma) = p$ and $\delta(q_1, \tau) = q$.*

From the Wheeler property and previous Lemma we can infer as a corollary the following lemma.

Lemma 2. *Let p and q be two consecutive states of $Q_{\mathcal{A}} \cup S$. If $\delta(p, \sigma)$ and $\delta(q, \sigma)$ are sink states, for some $\sigma \in \Sigma$, then $\delta(p, \sigma) = \delta(q, \sigma)$.*

By using Lemma 1 and Lemma 2 we can prove the following Theorem.

Theorem 2. *For each WDFA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ with all states coaccessible there exists a unique W-complete DFA $\mathcal{B}$ with minimal number of states such that $\mathcal{A} \subseteq \mathcal{B}$. We call it the* Wheeler completion (shortly W-completion) *of $\mathcal{A}$ and we denote it by $C_W(\mathcal{A})$.*

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

Theorem 3. *Let $\mathcal{A} = (Q, \Sigma, \delta, s, F, \leq)$ a WDFA with n states and over a k letter alphabet. The W-completion $C_W(\mathcal{A})$ has at most $2n + k - 2$ states.*

Remark that a W-complete automaton is not in general complete. See, for instance, the WDFA in 3(b) is the W-completion of the automaton in Figure 3(a), but it is not complete. Moreover, we say that a Wheeler automaton $\mathcal{A}$ is *completable* if $C_W(\mathcal{A})$ is complete.

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

Let $\mathcal{A} = (Q, \Sigma, \delta, 1, F)$ be a solo DFA, where $Q = \{q_1, q_2, \ldots, q_n\}$ is a totally ordered set of states.

In the sequel it is convenient to represent the transition function of the DFA as *transformations* of the set Q of states, i.e. a *partial mapping* of Q into itself (cf. for instance [13]). For each $\sigma \in \Sigma$, the transformation δ_σ is defined for $i \in Q$ as $\delta_\sigma(i) = \delta(i, \sigma)$. If $\delta(i, \sigma)$ is not defined

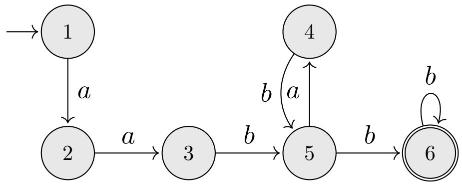


Figure 4: The Wheeler automaton of the Example 3.

we write $\delta_\sigma(i) = \square$ and we say that $\delta_\sigma(i)$ is a *missing σ-transition* (or a σ-hole). Hence, an arbitrary partial transformation δ_σ can be written in the form

$$\delta_\sigma = \begin{pmatrix} 1 & 2 & \cdots & n-1 & n \\ p_1 & p_2 & \cdots & p_{n-1} & p_n \end{pmatrix},$$

where $p_i = \delta_\sigma(i)$ and $p_i \in Q \cup \{\square\}$, for $1 \leq i \leq n$. We denote by R_σ the subsequence of $(p_1, p_2, \ldots, p_n)$ composed by the elements different from $\square$.

For each word $w \in \Sigma^*$, the transition function defines a transformation δ_w of Q: for all $i \in Q$, $\delta_w(i) = \delta^*(i, w)$.

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

- For each $\sigma \in \Sigma$, $1 \notin R_\sigma$;
- 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 $max(R_\sigma) < min(R_\tau)$.

The notion of *interval of missing transitions* (or *interval of holes*) plays an important role in our construction. For $i, j \in Q$ with $j - i \geq 2$, by $I_{i,j}$ we denote the *internal interval* $I_{i,j} = \{q \in Q |\ i < q < j\}$. Remark that, in our notation, the intervals $I_{i,j}$ do not contain the endpoints i and j. Further, we denote by $I_{0,j}$ and $I_{i,n+1}$ the *left* and *right intevals*: $I_{0,j} = \{q \in Q |\ q < j\}$ and $I_{i,n+1} = \{q \in Q |\ q > i\}$. We say that $I_{i,j}$, is an *interval of missing σ-transitions* if for each $k \in I_{i,j}$, $\delta_\sigma(k) = \square$ and $\delta_\sigma(i), \delta_\sigma(j) \neq \square$. We denote it by $H_\sigma(i, j)$. In a similar way, we define the *left interval of σ-holes* $H_\sigma(0, j)$ and the *right interval of σ-holes* $H_\sigma(i, n + 1)$.

Example 3. *Consider the WDFA in Figure 4 over the alphabet $\{a, b\}$. The transition function is defined by the following transformations:*

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

Then $R_a = (2, 3, 4)$ and $R_b = (5, 5, 6, 6)$ and the interval of missing a-transitions (a-holes) are $H_a(2, 5) = \{3, 4\}$ and $H_a(5, 7) = \{6\}$. The (unique) interval of missing b-transition (b-holes) is $H_b(0, 3) = \{1, 2\}$.

Now we give an algorithm that, receiving as input a WDFA $\mathcal{A}$ with all coaccessible states computes $C_W(\mathcal{A})$. It works by adding some sink states and transitions as much as possible to fill as much holes as possible by maintaining the Wheeler property. In the execution of the algorithm we will deal with WDFA $\mathcal{A} = (Q, \Sigma, \delta, 1, F)$ in which the set Q of states is the union $Q = Q_c \cup S$, where Q_c is the set of coaccesible states and S is the set of the sink states. The elements of Q_c are denoted by the integers $\{1, 2, \ldots n\}$ and the elements of S by rational non-integer numbers r with $1 < r < n + 1$. The total order in the elements of $Q = Q_c \cup S$ is the order of the rational numbers. In the input WDFA $\mathcal{A}$, we have $S = \emptyset$ and $Q = Q_c$.

At every step, in the run of the algorithm, we update only the set S and the transition function δ. The alphabet Σ, the set Q_c of coaccessible states and the set of final states F remain unchanged. The goal is to replace the missing transitions with proper transitions (maintaining the Wheeler property), hence we add a new sink state, when needed, and replace all the missing transitions in the interval with proper transitions converging to such a sink state. The original transitions remain unchanged.

If we refer to the missing transitions has 'holes', the above replacements are called *filling the holes*. More in detail, we distinguish two kinds of interval of holes. For each $\sigma \in \Sigma$, the interval of holes $H_\sigma(i, j)$ is said to be *of integer type* if both $\delta_\sigma(i)$ and $\delta_\sigma(j)$ are integers, i.e. elements of Q_c. Similarly, $H_\sigma(0, j)$ ($H_\sigma(i, n + 1)$) is said to be of *of integer type* if $\delta_\sigma(j)$ (resp. $\delta_\sigma(i)$) is an integer. The intervals of holes that are not of integer type are said to be of *non-integer type*.

Finally, an interval of holes $H_\sigma(i, j)$ of integer type is said to be *blocking* if $\delta_\sigma(i) = \delta_\sigma(j)$. All other intervals of holes are *non-blocking*.

The basic operations we consider consist of filling the holes and are described below in detail.

Fill internal and right intervals of σ-holes of integer type.

Let $H_\sigma(i, j)$, or $H_\sigma(i, n + 1)$, be an interval of σ-holes of integer type,

- Update S by creating a new state $t^i = \delta_\sigma(i) + 0.5$:
  $S := S \cup \{t^i\}$;
- Update δ:
  For all $i < k < j,\ \delta_\sigma(k) = t^i$;
  For all $\tau \in \Sigma,\ \delta_\tau(t^i) = \square$.

Fill left intervals of σ-holes of integer type.

Let $H_\sigma(0, j)$ be an interval of σ-holes of integer type,

- Update S by creating a new state $t^j = \delta_\sigma(j) - 0.3$:
  $S := S \cup \{t^j\}$;
- Update δ:
  For all $k < j,\ \delta_\sigma(k) = t^j$;
  For all $\tau \in \Sigma,\ \delta_\tau(t^j) = \square$.

Fill intervals of σ-holes of non-integer type.

In this case we do not distinguish between internal, right and left intervals of σ-holes. We denote by H_σ this interval. By hypothesis, one of the endpoints (say i) of H_σ is such that $\delta_\sigma(i)$ is not an integer, i.e. it is an element of S (a sink state). In this case, we do not add states to S, but we update only the transition function as follows:

- Update δ:
  For all $k \in H_\sigma,\ \delta_\sigma(k) = \delta_\sigma(i)$.

One can easily verify that applying any of the basic operations mentioned above to a WDFA results in another WDFA. Moreover, notice that in order to get a total ordered set of states, only holes in non-blocking intervals can be filled.

At each step, all the holes of all the non-blocking intervals are filled. Hence, some new states are created, from which the transitions have not yet been defined. And therefore some new holes are created, which, in turn, are filled by iterating the procedure.
The procedure stops when either all holes disappear (in such a case we obtain a complete automaton) or only blocking intervals of holes remains (in this second case we obtain a W-complete automaton which is not complete) (cf. Theorem 1).

Example 4. *Let $\mathcal{A} = (Q, \Sigma, \delta, 1, F, \leq)$ with $Q = \{1, 2, \ldots, 6\}$ depicted in Figure 4 and transition functions:*

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

By filling $H_a(2,5)$, $H_a(5,7)$ and $H_b(0,3)$ three sink states (3.5, 4.5 and 4.7) are added and the transition function is updated as follows:

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

By filling $H_b(4,5)$ one more state is added (the sink state 5.5) and by filling all the intervals of non-integer type the transition function is updated as follows:

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

By filling the last interval of non-integer type, we get the following W-complete automaton also depicted in Figure 5.

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

By the description of previous operations and by Theorem 1 one can infer the following theorem.

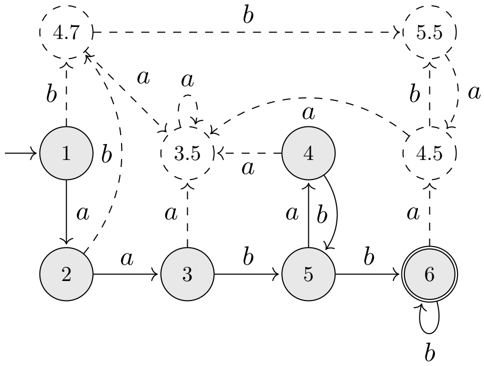


Figure 5: The W-completion of the automaton in Figure 4.

.

Theorem 4. *Let $\mathcal{A} = (Q, \Sigma, \delta, 1, F, \leq)$ be a WDFA with all coaccesible states. The Algorithm constructs $C_W(\mathcal{A})$.*

Figure 5 shows the W-completion where the new sink states and new transitions are dashed.

The W-completion $C_W(\mathcal{A}) = (Q_c \cup S, \delta', 1, F)$ contains both coaccessible and sink states, hence the following definition makes sense. We define $Dom(\mathcal{A})$ as the set of words that can be read by $C_W(\mathcal{A})$ from the initial state. More formally, $Dom(\mathcal{A}) = \{w \in \Sigma^* | \ \delta'^*(1, w) \neq \square\}$. For instance, if $\mathcal{A}$ is the WDFA of Example 2, $Dom(\mathcal{A}) = Pref(b^+a^+a + a^+)$. The following inclusions hold:

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

Moreover, we have that $C_W(\mathcal{A})$ is complete iff $Dom(\mathcal{A}) = \Sigma^*$. Such a concept is crucial for defining the operations on Wheeler automata described in the next section.

5. 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 1. *The construction of the DFA for the intersection does not require the DFA $\mathcal{A}_1$ and $\mathcal{A}_2$ to be complete. On the contrary, with the completeness hypothesis the constructions relative to the complement and the union always works.*

In the case of Wheeler languages, we are dealing with automata that are not, in general, complete then 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 the 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.

5.1. The Complement construction

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

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

If $C_W(\mathcal{A})$ is complete then $L(\mathcal{A}_c) = L(\mathcal{A})^c$.

Example 5. *Let us consider the transition function of the Wheeler automaton $\mathcal{A}$ in Figure 1(a) recognizing the language a^+b*

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

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

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

It is the complete automaton ($Dom(\mathcal{A}) = \Sigma^*$) in Figure 2 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) $L(\mathcal{A}_c)$ depends on $\mathcal{A}$, $L(\mathcal{A}_c) = Dom(\mathcal{A}) \setminus L(\mathcal{A})$ is a subset of $L(A)^c$. Remark that this result extends the one stated in Lemma 5.1, point 5, of [9], where the Wheelereness of $Pref(L(\mathcal{A})) \setminus L(\mathcal{A})$ is considered.

Example 6. *Let us consider the transition function of the Wheeler automaton in Figure 3(a) recognizing the language b^+a*

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

It cannot be completed (cf. Example 2) but it has a W-completion as follows.

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

It is the automaton in Figure 3(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 recognizing 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 7. *Let us 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*

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

5.2. 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 W-completion $C_W(\mathcal{A}_1)$ of the automaton $\mathcal{A}_1$ and the 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 set of final states the set of accessible states of $(F_1 \times Q'_2) \cup (Q'_1 \times F_2)$.

If $C_W(\mathcal{A}_1)$ and $C_W(\mathcal{A}_2)$ are complete, the automaton $C_W(\mathcal{A}_1) \times C_W(\mathcal{A}_2)$ is a WDFA indeed, 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 that $q_1 \leq p_1 \iff q_2 \leq p_2$, since the co-lexicographic order over the words corresponds to the total order between the states. By this remark, one can define a total order on the states of $C_W(A_1) \times C_W(A_2)$ satisfying the Wheeler conditions.

If one of the automata $C_W(\mathcal{A}_i)$, with $i \in \{1, 2\}$, is not complete, it happens that, for some state $p_i \in Q'_i$ and some letter $\sigma \in \Sigma$, the transition $\delta'_i(p_i, \sigma) = \square$ is a missing transition. We can consider $\square$ as new special sink state, that cannot be placed in order relation with other states.

In this case, the cartesian product $C_W(\mathcal{A}_1) \times C_W(\mathcal{A}_2)$ contains some states of type (p, q), but also some states of type $(p, \square)$ and $(\square, q)$ and $(\square, \square)$. Finally, let us define $\delta((p, \square), \sigma) = (\delta'_1(p, \sigma), \square)$, for any $p \in Q_1$ and $\sigma \in \Sigma$ and $\delta((\square, q), \sigma) = (\square, \delta'_2(q, \sigma))$, for any $q \in Q_2$ and $\sigma \in \Sigma$.

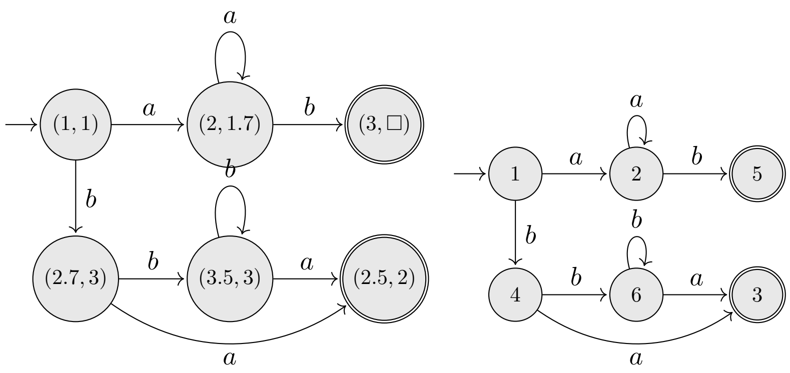


Figure 6: A Wheeler automaton for $a^+b \cup b^+a$.

If only states of the form (p, q) and $(p', \square)$ appear (in such accessible part), we are able to order the pairs with different first component: $(p, \square) < (p', \square) \iff p < p'$ and $(p, q) < (p', \square) \iff p < p'$, as showed in the following example. In a similar way we are able to order states of the form (p, q) and $(\square, q')$.

Example 8. *Let $\mathcal{A}_1$ be the WDFA (with $a < b$) recognizing the language a^+b and $C_W(\mathcal{A}_1)$ its W-completion, as in the Example 5, and let $\mathcal{A}_2$ be the WDFA recognizing the language b^+a and $C_W(\mathcal{A}_2)$ its W-completion, as in the Example 6. As Figure 6 shows, the union procedure in this case gives an automaton that contains only comparable pairs hence it is a Wheeler automaton that recognizes $a^+b \cup b^+a$.*

Theorem 5. *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 WDFA over the same alphabet Σ. If $L(\mathcal{A}_1) \subseteq Dom(A_2)$ and $L(\mathcal{A}_2) \subseteq Dom(A_1)$ then $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$ is a Wheeler language.*

Remark that Theorem 5 gives only a sufficient condition for the union. Example 8 shows that the construction relative to the union works also under more general conditions. An interesting open problem is to find necessary and sufficient conditions for the constructions relative to the union and the complement of Wheeler languages.

References

[1] M.-P. Béal, S. Lombardy, D. Perrin, Embeddings of local automata, Illinois Journal of Mathematics 54 (2010) 155 − 174.

[2] J. J. Ashley, B. H. Marcus, D. Perrin, S. Tuncel, Surjective extensions of sliding-block codes, SIAM J. Discret. Math. 6 (1993) 582–611.

[3] A. Ehrenfeucht, G. Rozenberg, Each regular code is included in A maximal regular code, RAIRO Theor. Informatics Appl. 20 (1986) 89–96.

[4] R. Montalbano, Local automata and completion, in: P. Enjalbert, A. Finkel, K. W. Wagner (Eds.), STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science,

Würzburg, Germany, February 25-27, 1993, Proceedings, volume 665 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 333–342.

[5] M. P. Schützenberger, A remark on incompletely specified automata, Inf. Control. 8 (1965) 373–376.

[6] T. Gagie, G. Manzini, J. Sirén, Wheeler graphs: A framework for bwt-based data structures, Theor. Comput. Sci. 698 (2017) 67–78.

[7] D. Gibney, S. V. Thankachan, On the complexity of recognizing wheeler graphs, Algorithmica 84 (2022) 784–814.

[8] 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, January 10 - 13, 2021, SIAM, 2021, pp. 2585–2599.

[9] J. Alanko, G. D'Agostino, A. Policriti, N. Prezza, Wheeler languages, Information and Computation 281 (2021) 104820.

[10] N. Cotumaccio, G. D'Agostino, A. Policriti, N. Prezza, Co-lexicographically ordering automata and regular languages - part I, J. ACM 70 (2023) 27:1–27:73.

[11] G. D'Agostino, D. Martincigh, A. Policriti, Ordering regular languages and automata: Complexity, Theor. Comput. Sci. 949 (2023) 113709.

[12] L. Egidi, F. A. Louza, G. Manzini, Space efficient merging of de bruijn graphs and wheeler graphs, Algorithmica 84 (2022) 639–669.

[13] J. A. Brzozowski, B. Li, D. Liu, Syntactic complexities of six classes of star-free languages, J. Autom. Lang. Comb. 17 (2012) 83–105.