A linear-size cascade decomposition for Wheeler automata

Giovanna D'Agostino[0000−0002−8920−483X], Luca Geatti[0000−0002−7125−787X],
Davide Martincigh[0000−0001−9988−6443], and Alberto
Policriti[0000−0001−8502−5896]

University of Udine, Italy
`{name.surname}@uniud.it`

Abstract. The Krohn-Rhodes Decomposition Theorem (KRDT) is a
central result in automata and semigroup theories: it states that any (de-
terministic) finite-state automaton can be disassembled into a collection
of automata of two simple types, that can be arranged into a combina-
tion – *cascade* – that simulates the original automaton. The elementary
building blocks of the decomposition are either *resets* or *permutations*.
The full-fledged theorem features two orthogonal dimensions of complex-
ity: the type and the number of building blocks appearing in the cascade,
and a deep step in the proof is the characterization of the permutations
appearing in the decomposition. This characterization implies, in the
case of *counter-free* automata, that the resulting cascade contains no
permutations.

In this paper we start analysing KRDT for two compression-oriented
classes of automata: (i) *path-coherent*: state-ordered automata mapping
state-intervals to state-intervals; (ii) *Wheeler*: a subclass of path-coherent
automata whose order is the one induced by the co-lexicographic order
of words. These classes were recently defined and studied and they turn
out to be efficiently encodable and indexable.

We prove that each automata in these classes can be decomposed as a
cascade with a number of components which is *linear* in the number of
states of the original automaton and, for the Wheeler class, we prove
that only two-state resets are needed. Our line of reasoning avoids the
necessity of using full KRDT and proves our results directly by a simple
inductive argument.

Keywords: Wheeler automata · Cascade decomposition · Krohn-Rhodes The-
orem.

1   Introduction

One of the most fruitful, pervasive, and basic idea in mathematics is *primal* decomposition, where a complex mathematical structure is decomposed in a more or less structured collection of simpler components. Examples are the factorization of a number into prime numbers, the many matrix decomposition used in linear algebra, the Fourier decomposition of a periodic function, or the Jordan-Holder Decomposition Theorem for finite groups.

The Krohn-Rhodes Decomposition Theorem (KRDT), in general terms, falls into this category: it is a method for decomposing a deterministic finite automaton into "simple" finite automata [6]. These components correspond to two-state *reset* automata (one bit memory automata acting as a two states resets or, equivalently, as on-off switches) and *permutation* automata, and are combined in a feedback-free composition called a "cascade".

Finite deterministic automata (DFA) correspond, via their transition semigroups, to finite semigroups and a cascade of DFAs correspond to a *wreath product* of semigroups. As a consequence KRDT is also an unexpected major result in the theory of semigroups, an analogue of the Jordan–Holder Theorem for finite groups (see [2]). In fact, even though semigroups/DFAs are characterised by very weak properties, KRDT reveals a surprising inner structure that was exploited in a wealth of connections within a variety of different areas (see [9]).

The permutations appearing in the KRDT cascade of an automaton are (homomorphic images of) subgroups of the transition semigroup of the original automaton. This is in fact the part of the KRDT that is most difficult to prove, requiring a careful analysis for the permutations' components. This is exploited for proving that the cascade decomposition of counter-free automata (that is automata without non-primitive cycles) is permutation-free, i.e. it contains only two-state resets. However, to the best of our knowledge, no proof of the decomposition in the counter-free case guarantees the linearity of the number of the two-state reset blocks [1].

In this paper we consider the KRDT for two classes of automata: *path-coherent* and *Wheeler* automata. The first class comprises automata that are state-ordered in such a way that the image of an interval of states, by any transition, is still an interval of states. The Wheeler class (see [1]) is defined

---

[1] In [8] it is proved that there is a family of counter-free automata admitting only exponential-sized cascades, but in this case the size refers to the total number of transitions in the cascade and not to the number of blocks.

using some *local axioms* constraining the state order to agree with the the co-lexicographic (co-lex) order of the words reaching states (with respect to an *a priori* fixed order of the alphabet's letters). Both classes where recently studied ([1, 3, 4]) and considered also for their amenability to being efficiently encodable and indexable.

We prove that for both path-coherent and Wheeler classes it is always possible to find a decomposition into a cascade made of a linear number of components. Moreover, the cascade whose existence we prove in the case of Wheeler automata is permutation-free and, contrary to the counter-free case, the proof is a simple induction, not requiring particular care for handling permutations.

The paper is organized as follows. Section 2 deals with notation and basic definitions. Section 3 introduces the path-coherent and Wheeler classes and prove some basic dependencies/independencies of the properties involved in their definitions. In section 4 we prove our main results on the existence of a cascade with a linear number of component for the aforementioned classes. Finally, section 5 is dedicated to conclusions and further works. The proofs that are not present in the body of the paper can be found in the Appendix.

2   Notation and First Definitions

The KRDT is a theorem on deterministic finite automata (DFAs), but initial and final states play no role in its statement. Hence we start defining semiautomata, which are DFAs where initial and final states are not specified. Moreover, the general KRDT theorem is stated for complete deterministic automata, where transitions from any states are always defined for every letter. The path-coherent and Wheeler automata, to be formally introduced below, are inherently partial, since their classes are not closed under complementation. Hence, we introduce the class of partial semiautomata and in Section 4 we state KRDT for this class.

A partial semiautomaton is a tuple $A = (S^A, \Sigma^A, \delta^A)$ where: (i) S^A is a finite set of states; (ii) Σ^A is the alphabet; (iii) $\delta^A : S^A \times \Sigma^A \to S^A$ is a partial function. In this paper, by a semiautomaton we always mean a partial semiautomaton. For any $a \in \Sigma^A$, we denote by $\delta^A(-, a)$ the (possibly partial) function from S^A to S^A that maps q in $\delta^A(q, a)$, for each $q \in S^A$. Moreover, for any subset of states $S \subseteq S^A$ and any symbol $a \in \Sigma^A$, we denote by $\delta^A(S, a)$ the set $\{q' \in S^A \mid q' = \delta^A(q, a) \wedge q \in S\}$. Since we will also introduce semiautomata $A = (S^A, \Sigma^A, \delta^A)$ and $B = (S^B, \Sigma^B, \delta^B)$ where $\Sigma^A = \Sigma^B$ and states in S^B are subsets of S^A – that is, $S^B \subseteq Pow(S^A)$ –, we avoid ambiguity by always

explicitly indicating the automaton, as in $\delta^A(S, a)$ and $\delta^B(S, a)$, so that it is clear whether S is a state of B or a subset of A-states.

Finally, if (q, a) does *not* belong to the domain of δ^A we write $\delta^A(q, a) = \bot$, where $\bot \notin S^A$. More in general, for a partial function $f\colon X \to Y$ and an element $x \in X$ we write $f(x) = \bot$ to denote that $x \notin \text{dom}(f)$. Clearly, a function can be defined only if all of its arguments are defined, thus we consider every function that has $\bot$ as one of its arguments as undefined – e.g. $f(\bot) = \bot$ and $(\bot, x) = (x, \bot) = \bot$ for all x. Moreover, given two functions $f, g\colon X \to Y$, when we write $\forall x \in X \; f(x) = g(x)$ we always imply that $f(x) = \bot$ if and only if $g(x) = \bot$, that is, f and g have the same domain.

Definition 1. *A transition* $a \in \Sigma$ *in a semiautomaton* $A = (S^A, \Sigma^A, \delta^A)$ *is: 1) a* reset *if either a) there exists a state* $r \in S^A$ *such that* $\delta^A(q, a) \neq \bot$ *implies* $\delta^A(q, a) = r$, *for all* $q \in S^A$; *or b)* $\delta^A(q, a) \neq \bot$ *implies* $\delta^A(q, a) = q$, *for all* $q \in S^A$ — *that is,* $\delta^A(-, a)$ *is the identity over its domain. 2) A* permutation *if it is not a reset and the function* $\delta^A(-, a)$ *is a permutation over its domain i.e. it is injective and maps its domain onto itself. A semiautomaton* A *is a permutation-reset semiautomaton if all its transitions are permutations or resets. We say that* A *is a reset semiautomaton if all its transitions are resets. We say that* A *is a permutation automaton if all its transitions are permutations.*
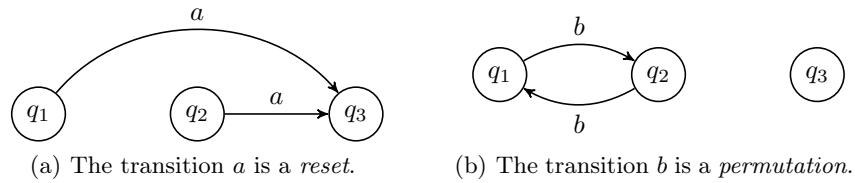

(a) The transition a is a *reset*.   (b) The transition b is a *permutation*.

Fig. 1. An example of *reset* and *permutation* transitions.

Definition 2 (Homomorphic image). *Let* A, B *be two semiautomata over the same alphabet* Σ. *We say that* A *is an* homomorphic image *of* B *if there is a surjective (total) function* $\phi\colon S^B \to S^A$ *such that, for all* $q \in S^B$ *and* $a \in \Sigma$,

$$\phi(\delta^B(q, a)) = \delta^A(\phi(q), a).$$

If ϕ *is a bijection,* A *and* B *are said to be* isomorphic.

A semiautomaton A can be transformed into a language acceptor by fixing an initial state s and a set of final states F and setting, as usual, $\mathcal{L}(A) = \{\alpha \in (\Sigma^A)^* \mid \delta^A(s, \alpha) \in F\}$. We say that a semiautomaton B is as least as expressive as the semiautomaton A if any language accepted by A is also accepted by B. The next lemma proves that if A is a homomorphic image of B then B is at least as expressive as A.

Lemma 1. *If the semiautomaton A is a homomorphic image of the semiautomaton B and the language L is accepted by A, then L is also accepted by B.*

We now introduce the operation we shall use in our decomposition results.

Definition 3 (Cascade product). *Let $B = (S^B, \Sigma, \delta^B)$ be a semiautomaton and $C = (S^C, S^B \times \Sigma, \delta^C)$ be a semiautomaton whose alphabet is the cartesian product of S^B and Σ. The cascade product $B \circ C = (S^{B \circ C}, \Sigma, \delta^{B \circ C})$ is the semiautomaton with set of states $S^{B \circ C} := S^B \times S^C$ and transition function defined by*

$$\delta^{B \circ C}((q, r), a) = (\delta^B(q, a), \delta^C(r, (q, a))).$$

Note that both δ^B and δ^C might be partial functions, thus we are implicitly requiring that $\delta^{B \circ C}((q, r), a) \neq \bot$ if and only if $\delta^B(q, a) \neq \bot \wedge \delta^C(r, (q, a)) \neq \bot$.

Figure 2 shows an example of cascade product between two automata.

(a) The automata B and C. The two links from Σ to C and from B to C indicate that, at each step, the automaton C reads both the input letter from Σ and the current state of the automaton B.

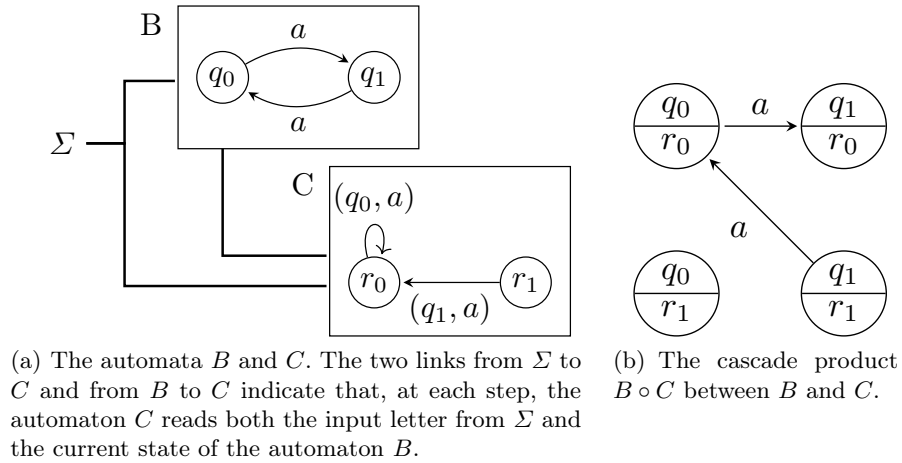(b) The cascade product $B \circ C$ between B and C.

Fig. 2. An example of cascade product.

Next, we state some basic results about cascades and homomorphic images. Since these results are normally stated and proved for complete semiautomata, we add the proofs for the more general partial case in the Appendix.

Proposition 1. *1) The cascade product is associative (see [5]). 2) If A, B, C, D are semiautomata such that A is an homomorphic image of $B \circ C$ and C is an homomorphic image of D, then A is an homomorphic image of $B \circ D$.*

3   Path-Coherence and Wheeler Axioms

In this section we introduce the two classes we shall analyze in terms of KRDT. We first consider a related property.

Definition 4 (Input Consistency).
A semiautomaton $A = (S^A, \Sigma^A, \delta^A)$ is input consistent *iff $\delta^A(v, a) = \delta^A(w, b) = u$ implies $a = b$, whenever $u, v, w \in S^A$ and $a \in \Sigma^A$.*

Path-coherence is a condition on graphs equipped with a total order over their nodes, that requires intervals of states to be mapped by transitions into intervals of states. It was originally introduced in the context of Wheeler graphs [4] and later exploited in the study of Wheeler languages [1]. We introduce below the adaptation of the definition to the case of semiautomata. As usual, if $(X, <)$ is a totally ordered set and $x \leq y \in X$, we denote by $[x, y]$ the interval $\{z \in X : x \leq z \leq y\}$.

Definition 5 (Path Coherence). *Let $<$ be a total order over the set of states S^A of a semiautomaton $A = (S^A, \Sigma^A, \delta^A)$. Then $(A, <)$ is* path-coherent *if, for all $a \in \Sigma^A$, the function $\delta(-, a)$ maps intervals in intervals. Equivalently, $(A, <)$ is* path-coherent *if, for all $q < q' \in S^A$, $a \in \Sigma^A$, and $x, y, z \in S^A$ it holds that:*

$$x < z < y \ \wedge \ x, y \in \delta^A([q, q'], a) \ \rightarrow \ z \in \delta^A([q, q'], a).$$

Figure 3 shows an example of a path-coherent automaton on the alphabet $\Sigma = \{a\}$.

Wheeler automata where first introduced in [4]. To define a Wheeler automaton we need to fix both an order $<$ on its states and an order $\prec$ on the alphabet Σ^A.

Definition 6 (Wheeler Axioms).
Let $A = (S^A, \Sigma^A, \delta^A)$ be a semiautomaton and let $\prec, <$ be a total order over the

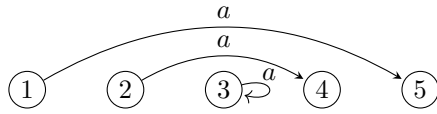**Fig. 3.** Example of a path-coherent semiautomaton.
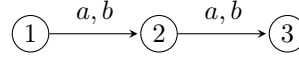
Fig. 4. A path coherent semiautomaton not satisfying W1.

alphabet Σ^A and a total order over the set of states S^A, respectively. Wheeler Axioms 1 (W1) and 2 (W2)[2] are stated as follows (see [4]). If $\delta^A(u_1, a_1) = v_1$ and $\delta^A(u_2, a_2) = v_2$ then:

(W1) $a_1 \prec a_2 \rightarrow v_1 < v_2$;
(W2) $(a_1 = a_2 \wedge u_1 < u_2) \rightarrow v_1 \leq v_2$.

A semiautomaton satisfying W1 and W2 is called a Wheeler semiautomaton. *A* Wheeler DFA *(WDFA) is a DFA based on a Wheeler semiautomaton. Moreover, it is required that the initial state has no ingoing transitions and it is the minimum in the order $<$.*
A Wheeler language *is a language recognized by a WDFA [3].*

If not specified otherwise we always assume that the set of states S^A of a semiautomaton A satisfying W1, W2, or path-coherence is $S^A = \{1, ..., n\}$, ordered as $1 < \cdots < n$.

In general, a path-coherent semiautomaton may not satisfy W1, for any order of the alphabet. Consider for example the semiautomaton A in Figure 4. Moreover, path-coherence and W1 do not imply W2. Consider for example the semiautomaton A in Figure 5.
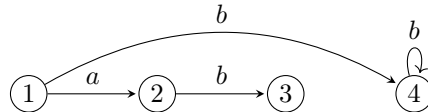

Fig. 5. A path-coherent semiautomaton satisfying W1 but not W2.

---

[2] The W2 axiom was introduced in [10] in the context of complete DFA to define the class of ordered DFAs.

[3] In [3] a more general definition of Wheeler Automata is given, not implying the input consistency of the automaton. However, the two definitions are equivalent with respect to the class of languages they recognize, i.e. the Wheeler Languages.

However, we do have the following dependencies.

Proposition 2. *1) If a semiautomaton A satisfies both path-coherence and input consistency then there is an order $\prec$ over the alphabet Σ for which A satisfies W1. In particular, a path-coherent, input consistent semiautomaton satisfying W2 is a Wheeler semiautomaton. 2) Any semiautomaton A satisfying both W1 and W2 and such that any state has at least an incoming transition is path-coherent.*

4   Krohn-Rhodes Decomposition for Path Coherent Graphs and Wheeler Graphs

The general KRDT states that each semiautomaton A is a homomorphic image of a sub-semiautomaton of a cascade product of two-state resets and permutation automata. Moreover, the cascade can be choosen in such a way that, for each permutation automaton in the cascade, the semigroup generated by the transitions of the automaton is a simple group which is an homomorphic image of subgroups of the semigroup generated by the A-transitions. This implies that any counter-free automaton is a homomorphic image of a (sub-)semiautomaton[4] of a cascade product of two-state resets. In this section we give a simpler proof of the decomposition for the path-coherent and Wheeler class. In the Wheeler case we also prove that the numbers of the blocks –two-state resets– is linear in the number of states of the original automaton.

We start by defining the first element of the cascade using the notion of *admissible decomposition* (see [5, 11]).

Definition 7. *A set $\mathcal{D}$ of subsets of the set of states of a semiautomaton $A = (S^A, \Sigma^A, \delta^A)$ is said to be an* admissible decomposition *if:*

1. *$\cup \mathcal{D} = S^A$;*
2. *for any $a \in \Sigma^A$ the image of an element of $\mathcal{D}$ under $\delta^A(-, a)$ is contained in at least one element of $\mathcal{D}$:*

$$\forall a \in \Sigma^A, D \in \mathcal{D} \; \exists D' \in \mathcal{D} \; \delta^A(D, a) \subseteq D'$$

---

[4] The general KRDT does not prove that the original automaton is a homomorphic image of the cascade: one has to go through a sub-semiautomaton of the cascade. For path consistent and Wheeler class we can avoid the use of sub-semiautomata.

Given an admissible decomposition $\mathcal{D}$ of a semiautomata A we can build a *factor* semiautomaton $A/\mathcal{D}$ over the same alphabet with $S^{A/\mathcal{D}} = \mathcal{D}$ and $\delta^{A/\mathcal{D}}(D, a) = D'$ where $D' \in \mathcal{D}$ is such that $\delta^A(D, a) \subseteq D'$ (if there is more than one, then choose one arbitrarily).[5]

Given a semiautomaton A, the first element of a cascade decomposition for A can be choosen to be a factor of A (see Zimmermann [11], Ginzburg [5]). However for the full KRDT, which proves that there are cascades where the permutation-blocks are homomorphic images of subgroups of the transition monoid of the original automaton, one has to choose a particularly well behaved decomposition of A.

In the following lemma and theorem we prove that, in case of path-coherent semiautomata, a particular choice for the decomposition of A allows to use a simple induction for obtaining the full decomposition. Moreover, when applied to the Wheeler case, we shall see that this decomposition avoid altogether the use of permutations in the cascade.

Lemma 2. *Let $A = (S^A, \Sigma, \delta^A, <)$ be a path-coherent semiautomaton with $n = |S^A| > 2$. Then, there are a permutation-reset semiautomaton B with $|S^B| = 2$ and a path-coherent semiautomaton C with $|S^C| = n - 1$, such that A is a homomorphic image of $B \circ C$.*

Proof. Consider the decomposition $\mathcal{D} = \{D_0, D_1\}$ of S^A where

$$D_0 = \{1, \ldots, n-1\}, D_1 = \{2, \ldots, n\}.$$

We have either $\delta^A(D_i, a) \subseteq D_0$ or $\delta^A(D_i, a) \subseteq D_1$ (possibly both) for $i = 0, 1$: it cannot be the case that both 1 and n are in $\delta^A(D_i, a)$ or, by path-coherence, we would have $\delta^A(D_i, a) = \{1, \ldots, n\}$ which is not possible, being A *deterministic*. In other words, the decomposition $\mathcal{D}$ is admissible. Let $B = A/\mathcal{D} = (S^B, \Sigma, \delta^B)$ be the $\mathcal{D}$-*factor* of A, defined as follows. The set of states is $S^B = \{D_0, D_1\}$ and we define $\delta^B(D_j, a) = \bot$ if and only if $\delta^A(D_j, a) = \emptyset$. If instead $\delta^A(D_j, a) \neq \emptyset$, then we define $\delta^B(D_j, a) = D_0$ if $\delta^A(D_j, a) \subseteq D_0$ (or, equivalently, if $n \notin \delta^A(D_j, a)$), whereas we define $\delta^B(D_j, a) = D_1$ if $\delta^A(D_j, a) \not\subseteq D_0$ (or, equivalently, $n \in \delta^A(D_j, a)$) and hence $\delta^A(D_j, a) \subseteq D_1$. More succinctly,

---

[5] For a complete semiautomaton A, where the transition functions are total, a factor of A is always a homomorphic image of A. This is not always the case for partial automata.

when $\delta^B(D_j, a) \neq \bot$ we have $\delta^B(D_j, a) = D_{\gamma(j,a)}$, where $\gamma(j, a)$ is defined as

$$\gamma(j, a) = \begin{cases} 0 & \text{if } n \notin \delta^A(D_j, a) \\ 1 & \text{if } n \in \delta^A(D_j, a) \end{cases}$$

for all $j \in \{0, 1\}$ and $a \in \Sigma$. Note that B has only two states. Hence, a transition in B can only be a (eventually partial) reset, the identity, or the permutation $\delta^B(D_0, a) = D_1$, $\delta^B(D_1, a) = D_0$. Therefore, B is a permutation-reset automaton.

Let $C = (S^C, S^B \times \Sigma, \delta^C)$ be the semiautomaton with set of states

$$S^C = \{C_1, \ldots, C_{n-1}\} \quad \text{with} \quad C_i = \{(i, D_0), (i+1, D_1)\} \quad \text{for } 1 \leq i \leq n-1$$

and transition function defined as follows, for $i = 1, \ldots, n - 1$ and $j = 0, 1$:

$$\delta^C\Big(C_i, (D_j, a)\Big) = C_{\delta^A(i+j,a)-\gamma(j,a)}. \tag{1}$$

Note that $\delta^C\Big(C_i, (D_j, a)\Big)$ is well defined because it always hold

$$0 < \delta^A(i + j, a) - \gamma(j, a) < n.$$

Assuming, for contradiction, that $\delta^A(i + j, a) - \gamma(j, a) = n$, we must have both $\delta^A(i + j, a) = n$ and $\gamma(j, a) = 0$. Note that $i + j \in D_j$, for all $i = 1, \ldots, n - 1$ and $j = 0, 1$, thus, from $\delta^A(i+j, a) = n$ it follows that $n \in \delta^A(D_j, a)$. By definition of γ, this means that $\gamma(j, a) = 1$: a contradiction. Therefore $\delta^A(i+j, a) - \gamma(j, a) \neq n$. The fact that $\delta^A(i + j, a) - \gamma(j, a) \neq 0$ can be proved in a similar manner.

We prove that A is a homomorphic image of the cascade $B \circ C$. Consider the function

$$\phi \colon S^{B \circ C} \to S^A \quad \text{defined by} \quad \phi(D_j, C_i) = i + j.$$

The codomain of ϕ is S^A because if $i = 1, \ldots, n - 1$ and $j = 0, 1$ then $i + j \in \{1, \ldots, n\} = S^A$. Moreover, ϕ is surjective. Notice that, as opposed to the Krohn-Rhodes general case, ϕ is a total function.

We prove that ϕ is a homomorphism from $B \circ C$ to A, that is, for all $j = 0, 1$ and $i = 1, \ldots, n - 1$:

$$\phi(\delta^{B \circ C}((D_j, C_i), a)) = \delta^A(\phi(D_j, C_i), a).$$

In fact, we have

$$\phi(\delta^{B \circ C}((D_j, C_i), a)) = \phi(\delta^B(D_j, a), \delta^C(C_i, (D_j, a))) =$$

$$\phi(D_{\gamma(j,a)}, C_{\delta^A(i+j,a)-\gamma(j,a)}) = \delta^A(i+j,a) - \gamma(j,a) + \gamma(j,a) = \delta^A(\phi(D_j, C_i), a).$$

To conclude the proof of the lemma we prove that the order $C_1 < \cdots < C_{n-1}$ makes C path-coherent. Let $[C_k, C_l]$ be any interval, with $1 \leq k \leq l \leq n-1$. From equation (1) it follows that, for all $j \in \{0, 1\}$ and for all $a \in \Sigma$,

$$\delta^C\Big([C_k, C_l], (D_j, a)\Big) = \{C_h \ : \ h + \gamma(j, a) \in \delta^A([k+j, l+j], a)\}.$$

By the hypothesis that A is path-coherent it follows that $\delta^A([k+j, l+j], a)$ is an interval, hence also the set $\delta^C\Big([C_k, C_l], (D_j, a)\Big)$ is an interval. $\square$

Theorem 1. *Each path-coherent semiautomaton A with $n \geq 2$ states is an homomorphic image of a cascade product of $n-1$ permutation-reset semiautomata, each one having exactly 2 states.*

Proof. By induction using Lemma 2. $\square$

Figure 6 shows the cascade decomposition of the path-coherent automaton depicted in Figure 3.

Note that, although the automaton in Figure 3 is counter-free and, as so, KRDT decomposes it using resets only, the decomposition obtained from Theorem 1 (see Figure 6) contains blocks with permutations. In this sense, this decomposition is not optimal as the one obtained in the original KRDT. However, as we shall see shortly, in case of path-coherence plus W2 (hence in the Wheeler case as well), in which automata are always counter-free, the decomposition that we propose in Theorem 2 allows us to produce a permutation-free cascade.

Lemma 3. *Let $A = (S^A, \Sigma, \delta^A, <)$ be a path-coherent semiautomaton satisfying Wheeler axiom 2 (W2). Then the construction of Lemma 2 provides a reset semiautomaton B with $|S^B| = 2$ and a path-coherent semiautomaton C with $|S^C| = n-1$ still satisfying W2 such that A is an homomorphic image of $B \circ C$.*

Theorem 2. *Each path-coherent semiautomaton A satisfying W2 with $n \geq 2$ states is the homomorphic image of a cascade product of $n-1$ reset semiautomata, each one having exactly 2 states.*
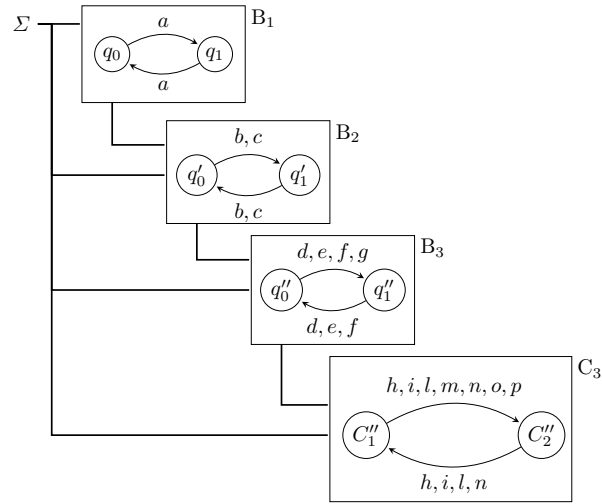
Fig. 6. Cascade decomposition of the automaton in Figure 3. The letters $b, c, d, e, f, g, h, i, l, m, n, o, p$ are shortcuts for $(q_0, a), (q_1, a), (q'_0, b), (q'_1, b), (q'_0, c), (q'_1, c), (q''_0, d), (q''_1, d), (q''_0, e), (q''_1, e), (q''_0, f), (q''_1, f), (q''_0, g), (q''_1, g)$, respectively.

Proof. The proof is the same as the one of Theorem 1, but we apply Lemma 3 instead of Lemma 2 everywhere. □

Since Wheeler automata are path-coherent and satisfy W2, we get

Corollary 1. *Wheeler semiautomaton A with $n \geq 2$ states is the homomorphic image of a cascade product of $n-1$ reset semiautomata, each one having exactly 2 states.*

5   Conclusions

In this paper, we proved that path-coherent automata admit a linear-sized cascade decomposition into permutation-reset automata. Moreover we showed that, for the case of Wheeler automata, the cascade is permutation-free.

Some further natural questions arise. For example:

– Can we find a linear-sized permutation-free decomposition for counter-free, path-coherent automata?
– Is there a condition on the blocks of a cascade of resets in order for it to be a Wheeler automaton?

- The *Krohn-Rhodes complexity* (KR-complexity) of a finite semigroup was introduced in [7] as a way of measuring the complexity of a cascade based on the least number of simple groups in it. Can the KR-complexity of path-coherent automata be effectively computed? If yes, what is the computational complexity of establishing their KR-complexity?
- Are there efficient algorithms implementing Theorems 1 and 2?

In addition to the above, more theoretical, issues one might wonder, on a more practical side, if the cascade decomposition can be used in the index production/optimization for a Wheeler automaton.

Bibliography

[1] J. Alanko, G. D'Agostino, A. Policriti, and N. Prezza. Wheeler languages. *Information and Computation*, 281:104820, 2021.

[2] M. Aschbacher. *Finite group theory*. Cambridge studies in advanced mathematics. Cambridge University Press, 1986. ISBN 9780521303415. URL `https://books.google.it/books?id=SGQrnwEACAAJ`.

[3] N. Cotumaccio, G. D'Agostino, A. Policriti, and N. Prezza. Co-lexicographically ordering automata and regular languages. Part I. *J. ACM*, jul 2023. ISSN 0004-5411. `https://doi.org/10.1145/3607471`.

[4] T. Gagie, G. Manzini, and J. Sirén. Wheeler graphs: a framework for BWT-based data structures. *Theoretical computer science*, 698:67–78, 2017.

[5] A. Ginzburg. *Algebraic theory of automata*. New York, 1968.

[6] K. Krohn and J. Rhodes. Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. *Transactions of the American Mathematical Society*, 116:450–464, 1965.

[7] K. Krohn and J. Rhodes. Complexity of finite semigroups. *Annals of Mathematics*, 88(1):128–160, 1968. ISSN 0003486X. URL `http://www.jstor.org/stable/1970558`.

[8] O. Maler and A. Pnueli. Tight bounds on the complexity of cascaded decomposition of automata. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, pages 672–682. IEEE Computer Society, 1990. `https://doi.org/10.1109/FSCS.1990.89589`.

[9] J. Rhodes. *Applications of automata theory and algebra: via the mathematical theory of complexity to biology, physics, psychology, philosophy, and games*. World Scientific Publishing, Singapore, 2010. ISBN 9812836977.

[10] H.-J. Shyr and G. Thierrin. Ordered automata and associated languages. *Tamkang J. Math*, 5:9–20, 1974.

[11] K. Zimmermann. On krohn-rhodes theory for semiautomata. *CoRR*, abs/2010.16235, 2020. URL `https://arxiv.org/abs/2010.16235`.