Information and Computation ••• (••••) ••••••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Regular ω-languages with an informative right congruence

Dana Angluin [a], Dana Fisman [b,*]

[a] Yale University, United States of America
[b] Ben-Gurion University, Israel

ARTICLE INFO

Article history:
Available online xxxx

ABSTRACT

A regular language is almost fully characterized by its right congruence relation. The same does not hold for regular ω-languages. The right congruence of a regular ω-language may not be informative enough; many regular ω-languages have a trivial right congruence, and in general it is not always possible to define an ω-automaton recognizing a given language that is isomorphic to its right congruence.

The *weak regular ω-languages* do have fully informative right congruences. That is, any weak regular ω-language can always be recognized by a deterministic Büchi automaton that is isomorphic to its right congruence. Weak regular ω-languages reside in the lower levels of the expressiveness hierarchy of regular ω-languages. Are there more expressive sub-classes of regular ω-languages that have fully informative right congruences? Can we characterize the class of languages that have trivial right congruences? In this paper we try to place some additional pieces of this big puzzle.

1. Introduction

Regular ω-languages play a key role in reasoning about reactive systems. Algorithms for verification and synthesis of reactive system typically build on the theory of ω-automata. The theory of ω-automata enjoys many properties that the theory of automata on finite words enjoys. These make it amenable for providing the basis for analysis algorithms (e.g., model checking algorithms rely on the fact that emptiness can be checked in nondeterministic logarithmic space). However, in general, the theory of ω-automata is much more involved than that of automata on finite words, and many fundamental questions, such as minimization, are still open.

One of the fundamental theorems of regular languages on finite words is the Myhill-Nerode theorem stating a one-to-one correspondence between the states of the minimal deterministic finite automaton (DFA) for a language L and the equivalence classes of the right congruence of L.[1] When moving to ω-words, there is no similar theorem, and there are many regular ω-languages where any minimal automaton requires more states than the number of equivalence classes in the right congruence of the language. For instance, consider the ω-language $L = (a + b)^* a^\omega$. Its right congruence has only one equivalence class. That is, for any finite words x and y and any ω-word w we have that $xw \in L$ iff $yw \in L$ as membership in L is determined only by the suffix. However, the smallest ω-automaton for this language requires at least two states. We thus say that the right congruence for L is *not fully informative*. If L has an ω-automaton that is isomorphic to the automaton derived from the right congruence for L, we say that the right congruence for L is *fully informative*.

* Corresponding author.
E-mail addresses: angluin@cs.yale.edu (D. Angluin), dana@cs.bgu.ac.il (D. Fisman).

[1] Formal definitions are deferred to Section 2.

https://doi.org/10.1016/j.ic.2020.104598
0890-5401/

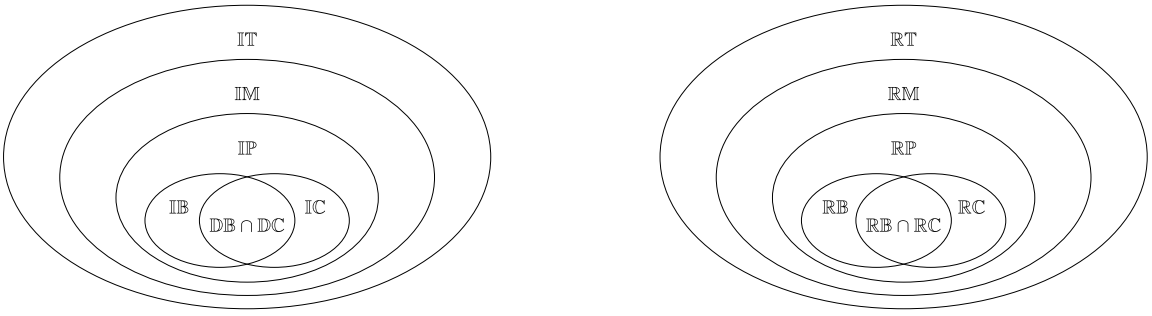

Fig. 1. On the left, a summary of inclusion of the classes of languages accepted by ω-automata that are isomorphic to their rightcon automaton. On the right a summary of inclusion of the classes of languages accepted by ω-automata that are isomorphic to their rightcon automaton and also respective of their right congruence. For each $\mathbb{X} \in \{\mathbb{B}, \mathbb{C}, \mathbb{P}, \mathbb{M}, \mathbb{T}\}$, the class $\mathbb{RX}$ is properly included in the class $\mathbb{IX}$.

The tight relationship between the equivalence classes of the right congruence and the states of a minimal DFA is at the heart of minimization and learning algorithms for regular languages of finite words, and seems to be a severe obstacle in obtaining efficient minimization and learning algorithms for regular ω-languages. For this reason, we set ourselves to study classes of regular ω-languages that do have a right congruence that is fully informative.

Several acceptance criteria are in use for ω-automata, in particular, Büchi, co-Büchi, parity, and Muller. There are differences in the expressiveness of the corresponding deterministic automata. We use $\mathbb{DB}$, $\mathbb{DC}$, $\mathbb{DP}$ and $\mathbb{DM}$ to denote the classes of languages accepted by deterministic Büchi, co-Büchi, parity, and Muller automata, respectively. We also consider a version of Muller automata where acceptance is defined using sets of transitions, refer to this acceptance criterion as Transition-Table, and use $\mathbb{DT}$ for the corresponding class of languages. The classes $\mathbb{DT}$, $\mathbb{DM}$ and $\mathbb{DP}$ are as expressive as the full class of regular ω-languages whereas $\mathbb{DB}$ and $\mathbb{DC}$ are strictly less expressive and are dual to each other. The intersection of $\mathbb{DB}$ and $\mathbb{DC}$ is the class of *weak* regular ω-languages. This class does have the property that any language in $\mathbb{DB} \cap \mathbb{DC}$ has a fully informative right congruence. The regular ω-languages can be arranged in an infinite hierarchy of expressive power defined by Wagner [32] and the class $\mathbb{DB} \cap \mathbb{DC}$ corresponds to one of the lowest levels of the hierarchy.

We define the classes $\mathbb{IB}$, $\mathbb{IC}$, $\mathbb{IP}$, $\mathbb{IM}$, $\mathbb{IT}$ to be the classes of regular ω-languages that can be accepted by Büchi, co-Büchi, parity and Muller and Transition-Table automata, respectively, which are isomorphic to the automaton derived from the right congruence of the language. We show that these form a strictly inclusive hierarchy of expressiveness as shown in Fig. 1 (p. 2) on the left, and moreover in every class of the infinite Wagner hierarchy of regular ω-languages there are languages whose right congruence is fully informative.

Noting another difficulty in inferring a regular ω-language from examples of ω-words in the language, we consider a further restriction on languages, that if $u(x)^\omega$ is accepted by a minimal automaton for the language, then that automaton has a loop of size at most $|x|$ in which $u(x)^\omega$ loops. We term this property being *respective* of the right congruence. This property is reminiscent of the property of being non-counting [13], and we show that a language that is non-counting is respective of its right congruence but the reverse implication does not hold in general. We define the classes $\mathbb{RB}$, $\mathbb{RC}$, $\mathbb{RP}$, $\mathbb{RM}$, $\mathbb{RT}$ of languages in $\mathbb{IB}$, $\mathbb{IC}$, $\mathbb{IP}$, $\mathbb{IM}$, $\mathbb{IT}$ that are respective of their right congruence. We show that these classes constitute a further restriction in terms of expressive power as shown in Fig. 1 (p. 2) on the right. Yet in this case as well, in every class of the infinite Wagner hierarchy of regular ω-languages there are languages which are respective of their fully informative right congruence.

The rest of the paper is organized as follows. In Section 2 we provide the necessary definitions of the ω-automata that we consider and of right congruence, state the well known results about their expressiveness, and briefly summarize the importance of the relation between the syntactic right congruence of a language and its minimal acceptor in learning algorithms. In Section 3 we prove basic results about the relations between an ω-acceptor and the syntactic right congruence of the language it recognizes. In Section 4 we provide a new characterization of regular ω-languages for which the right congruence is trivial. In Section 5 we establish expressiveness results related to the classes of languages with fully informative right congruences. In Section 6 we establish expressiveness results related to the classes of languages that not only have fully informative right congruences, but are also respective of their right congruences. In Section 7 we consider closure properties of these classes. In Section 8, we present the results of a small empirical experiment on the frequency of fully informative right congruences among randomly generated Muller automata, and in Section 9 we conclude.

2. Preliminaries

An *alphabet* Σ is a finite set of symbols. The set of finite words over Σ is denoted by Σ^*, and the set of infinite words, termed ω-words, over Σ is denoted by Σ^ω. We use ϵ for the empty word and Σ^+ for $\Sigma^* \setminus \{\epsilon\}$. A *language* is a set of finite words, that is a subset of Σ^*, while an ω-language is a set of ω-words, that is a subset of Σ^ω. For a language L, its *complement* L^c is the language $(\Sigma^* \setminus L)$. For an ω-language L, its *complement* L^c is the ω-language $(\Sigma^\omega \setminus L)$.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

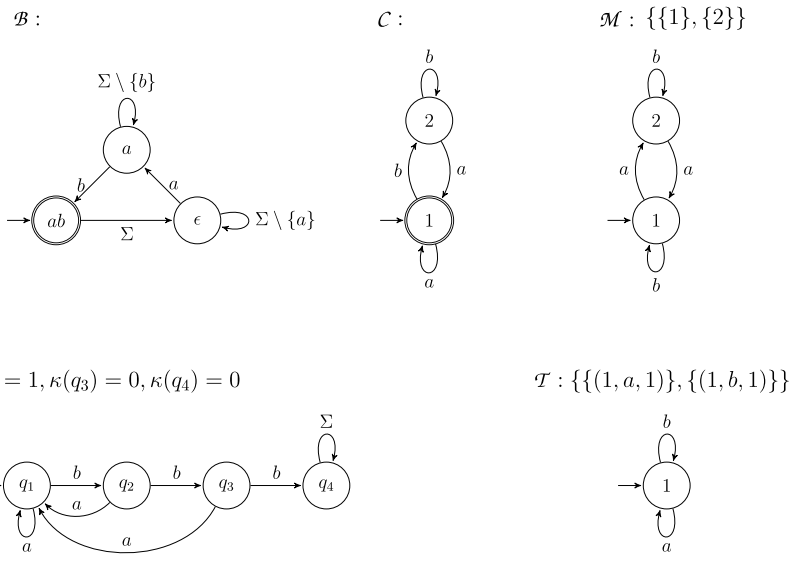

Fig. 2. A DBA $\mathcal{B}$ accepting $(\Sigma^* a \Sigma^* b)^\omega$ where $\Sigma = \{a, b, c\}$, a DCA C accepting $(a+b)^* b^\omega$, a DMA $\mathcal{M}$ accepting $(a+b)^* b^\omega$, a DPA $\mathcal{P}$ accepting $(a + ba + bba)^*(a^* ba)^\omega$, and a DTA $\mathcal{T}$ accepting $(a+b)^*(a^\omega + b^\omega)$. For DBAs and DCAs states in F are marked by double-lines.

The set of natural numbers is denoted $\mathbb{N}$. For natural numbers i and j and a word w, we use $[i..j]$ for the set $\{i, i+1, \ldots, j\}$, $w[i]$ for the i-th letter of w (where indexing starts from 1), and $w[i..j]$ for the subword of w starting at the i-th letter and ending at the j-th letter, inclusive.

Automata and acceptors An *automaton* is a tuple $\mathcal{A} = \langle \Sigma, Q, \lambda, \delta \rangle$ consisting of an alphabet Σ, a finite set Q of states, an initial state $\lambda \in Q$, and a transition function $\delta : Q \times \Sigma \to 2^Q$. A run of an automaton on a finite word $v = a_1 a_2 \ldots a_n$ is a sequence of states $q_0, q_1, \ldots, q_n$ such that $q_0 = \lambda$, and for each $i \geq 0$, $q_{i+1} \in \delta(q_i, a_{i+1})$. A run on an infinite word is defined similarly and results in an infinite sequence of states. We say that $\mathcal{A}$ is *deterministic* if $|\delta(q,a)| \leq 1$ and *complete* if $|\delta(q,a)| \geq 1$, for every $q \in Q$ and $a \in \Sigma$.[2] When $\mathcal{A}$ is deterministic we regard δ as a function $\delta : Q \times \Sigma \to Q$. Let $\mathcal{A} = \langle \Sigma, Q, \lambda, \delta \rangle$ be a deterministic automaton. The transition function of $\mathcal{A}$ is naturally extended to a function from $Q \times \Sigma^*$, by defining $\delta(q, \epsilon) = q$ and $\delta(q, av) = \delta(\delta(q,a), v)$ for $q \in Q$, $a \in \Sigma$ and $v \in \Sigma^*$. We often use $\mathcal{A}(v)$ as a shorthand for $\delta(\lambda, v)$ and $|\mathcal{A}|$ for the number of states in Q. We use $\mathcal{A}_q$ to denote the automaton $\langle \Sigma, Q, q, \delta \rangle$ obtained from $\mathcal{A}$ by replacing the initial state with q.

By augmenting an automaton with an acceptance condition α, obtaining a tuple $\langle \Sigma, Q, \lambda, \delta, \alpha \rangle$, we get an *acceptor*, a machine that accepts some words and rejects others. An acceptor accepts a word if at least one of the runs on that word is accepting. For finite words the acceptance condition is a set $F \subseteq Q$ and a run $q_0, q_1, \ldots, q_n$ on a word v is accepting if it ends in an accepting state, i.e., if $q_n \in F$. We use DFA (resp. NFA) for deterterministic (resp. non-deterministic) finite state acceptors on finite words. For infinite words, there are various acceptance conditions in the literature; we consider five: Büchi, co-Büchi, parity, Muller and Transition-Table [28]. The Büchi and co-Büchi acceptance conditions are also a set $F \subseteq Q$. A run of a Büchi automaton is accepting if it visits F infinitely often. A run of a co-Büchi is accepting if it visits F only finitely many times. A parity acceptance condition is a map $\kappa : Q \to \mathbb{N}$ assigning each state a natural number termed a color (or priority). A run is accepting if the minimal color visited infinitely often is odd. A Muller acceptance condition is a set of sets of states $\alpha = \{F_1, F_2, \ldots, F_k\}$ for some $k \in \mathbb{N}$ and $F_i \subseteq Q$ for $i \in [1..k]$. A run of a Muller automaton is accepting if the set S of states visited infinitely often in the run is in α. A Transition-Table acceptance condition is a set $\alpha = \{T_1, T_2, \ldots, T_k\}$ of sets of transitions, where a transition is a tuple in $Q \times \Sigma \times Q$. A run of a Transition-Table automaton is accepting if the set T of transitions visited infinitely often in the run is in α. We use $[\![\mathcal{A}]\!]$ to denote the set of words accepted by a given acceptor $\mathcal{A}$.

We use three letter acronyms to describe acceptors, where the first letter is in {D, N} and denotes whether the automaton is *deterministic* or *nondeterministic*. The second letter is one of {B,C,P,M,T} for the first letter of the acceptance condition: Büchi, co-Büchi, parity, Muller or Transition-Table. The third letter is always A for acceptor. Fig. 2 (p. 3) gives examples for a DBA, DCA, DPA, DMA, and DTA, and specifies the accepted languages. A language is said to be *regular* if it is accepted by a DFA. An ω-language is said to be *regular* if it is accepted by a DMA.

Complexity and expressiveness of sub-classes of regular ω-languages We use $\mathbb{DB}$, $\mathbb{DC}$, $\mathbb{DP}$, $\mathbb{DM}$ and $\mathbb{DT}$ to denote the classes of languages accepted by DBA, DCA, DPA, DMA and DTA, respectively, and $\mathbb{NB}$, $\mathbb{NC}$, $\mathbb{NP}$, $\mathbb{NM}$ and $\mathbb{NT}$ for the class

[2] Throughout the paper we use mainly complete deterministic automata, though in figures we often omit the sink state.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

of languages accepted by NBA, NCA, NPA, NMA and NTA, respectively. The classes $\mathbb{NB}$, $\mathbb{NP}$, $\mathbb{NM}$, $\mathbb{NT}$, $\mathbb{DP}$, $\mathbb{DM}$ and $\mathbb{DT}$ are equi-expressive and contain all ω-regular languages. The classes $\mathbb{DB}$ and $\mathbb{DC}$ are strictly less expressive and are dual to each other in the sense that $L \in \mathbb{DB}$ iff $L^c \in \mathbb{DC}$, where L^c is the complement of L. The classes $\mathbb{NC}$ and $\mathbb{DC}$ are equi-expressive.

An ω-language is said to be *weak* if it is accepted by a DBA as well as by a DCA. In the notation introduced in [20], a weak version of an automaton $\mathbb{DX}$, denoted $\mathbb{DWX}$, is obtained by replacing in its acceptance definition the requirement to visit infinitely often with the requirement to visit at least once. The class $\mathbb{DB} \cap \mathbb{DC}$ is equivalent to the class $\mathbb{DWP}$ of languages accepted by deterministic weak parity automata. Since the class $\mathbb{DP}$ accepts the entire class of regular ω-languages, the term *weak ω-regular languages* has come to be used for the class $\mathbb{DWP}$.

A nonempty subset S of the automaton states, where for every $s_1, s_2 \in S$ there exists a string $x \in \Sigma^+$ such that $\delta(s_1, x) = s_2$ and $\delta(s_1, x') \in S$ for every prefix x' of x is termed an SCC (abbreviating strongly connected component). Note that there is no requirement for S to be maximal in this sense. A Muller automaton $\mathcal{M}$ can be seen as classifying its SCCs into *accepting* and *rejecting*. An important measure of the complexity of a Muller automaton is the number of alternations between accepting and rejecting SCCs along an inclusion chain (we expand on this in the sequel, on p. 9). For instance, the Muller automaton $\mathcal{M}$ in Fig. 2 (p. 3) has an inclusion chain of SCCs with one alternation: $\{1\}$ (accepting) $\subseteq \{1, 2\}$ (rejecting); and the Muller automaton $\mathcal{M}$ in Fig. 3 (p. 7) whose only accepting SCC is $\{\lambda, b\}$ has an inclusion chain of SCCs with two alternations: $\{b\}$ (rejecting) $\subseteq \{\lambda, b\}$ (accepting) $\subseteq \{\lambda, a, b\}$ (rejecting). Wagner [32] shows that this complexity measure is a property of the language and is invariant over all Muller automata accepting the same language. Under this view, $\mathbb{DB}$ is the class of languages where no superset of an accepting SCC can be rejecting, $\mathbb{DC}$ is the class of languages where no subset of an accepting SCC can be rejecting, and $\mathbb{DB} \cap \mathbb{DC}$ is the class of languages where no alternation between accepting and rejecting SCCs is allowed along any inclusion chain. Thus, the language $[\![\mathcal{M}]\!]$ for the $\mathcal{M}$ in Fig. 2 (p. 3) is not in $\mathbb{DB}$ and the language $[\![\mathcal{M}]\!]$ for the $\mathcal{M}$ in Fig. 3 (p. 7) is not in $\mathbb{DB} \cup \mathbb{DC}$.

Right congruence and the rightcon automaton An equivalence relation $\sim$ on Σ^* is a *right congruence* if $x \sim y$ implies $xv \sim yv$ for every $x, y, v \in \Sigma^*$. The *index* of $\sim$, denoted $|\sim|$ is the number of equivalence classes of $\sim$. Given a language L its *canonical right congruence* $\sim_L$ is defined as follows: $x \sim_L y$ iff $\forall v \in \Sigma^*$ we have $xv \in L \iff yv \in L$. When it is clear from the context, for a word $v \in \Sigma^*$ the notation $[v]$ is used for the class of $\sim$ in which v resides.

With a right congruence $\sim$ of finite index one can naturally associate an automaton $\mathcal{M}_\sim = \langle \Sigma, Q, \lambda, \delta \rangle$ as follows: the set of states Q consists of the equivalence classes of $\sim$. The initial state λ is the equivalence class $[\epsilon]$. The transition function δ is defined by $\delta([u], a) = [ua]$. In the sequel, we use $\mathcal{R}_L$ for the automaton $M_{\sim_L}$ associated with the right congruence of a given language L, and call it the *rightcon automaton of L*.

Similarly, given an automaton $\mathcal{M} = \langle \Sigma, Q, \lambda, \delta \rangle$ we can naturally associate with it a right congruence as follows: $x \sim_{\mathcal{M}} y$ iff $\mathcal{M}(x) = \mathcal{M}(y)$. The Myhill-Nerode Theorem states that a language L is regular iff $\sim_L$ is of finite index. Moreover, if L is accepted by a DFA $\mathcal{A}$ then $\sim_{\mathcal{A}}$ refines $\sim_L$. Finally, the index of $\sim_L$ gives the size of the minimal DFA for L.

For ω-languages, the right congruence $\sim_L$ is defined similarly, by quantifying over ω-words. That is, $x \sim_L y$ iff $\forall w \in \Sigma^\omega$ we have $xw \in L \iff yw \in L$. Given a deterministic automaton $\mathcal{M}$ we can define $\sim_{\mathcal{M}}$ exactly as for finite words. However, for ω-regular languages, right congruence alone does not suffice to obtain a "Myhill-Nerode" characterization. As an example consider the language $L = (a + b)^*(aba)^\omega$. We have that $\sim_L$ consists of just one equivalence class, since for any $x \in \Sigma^*$ and $w \in \Sigma^\omega$ we have that $xw \in L$ iff w has $(aba)^\omega$ as a suffix. But an ω-acceptor recognizing L obviously needs more than a single state. Note that the other side of the story entails that there are ω-automata that are minimal, although two different states recognize the same language. For instance, the DBA $\mathcal{B}$ in Fig. 2 (p. 3) is minimal for $[\![\mathcal{B}]\!]$ but $[\![\mathcal{B}_\epsilon]\!] = [\![\mathcal{B}_a]\!] = [\![\mathcal{B}_{ab}]\!]$, and the DMA $\mathcal{M}$ of Fig. 2 (p. 3) is minimal for $[\![\mathcal{M}]\!]$ but $[\![\mathcal{M}_1]\!] = [\![\mathcal{M}_2]\!]$. In general the problem of minimizing DBAs and DPAs is known to be NP-complete [29].

Grammatical inference *Grammatical inference* or *automata learning* refers to the problem of designing algorithms for inferring an unknown language from good and bad examples, i.e. from words labeled by their membership in the language. The learning algorithm is required to return some concise representation of the language, typically an automaton. The task of a learning algorithm can thus be thought of as trying to distinguish the different necessary states of an automaton recognizing the language and establishing the transitions between them. For a regular language, by the Myhill-Nerode theorem, $\sim_L$ can be used to distinguish states. Indeed, if the algorithm learns that $u_1 v \in L$ and $u_2 v \notin L$ for some $u_1, u_2, v \in \Sigma^*$ then $u_1 \not\sim_L u_2$ and the words u_1 and u_2 must reach two different states of the minimal DFA for L. Once all equivalence classes of $\sim_L$ are discovered, the automaton $\mathcal{M}_{\sim_L}$ can be extracted, and by setting the state corresponding to the empty word to be the initial state, and states corresponding to positive examples as accepting, the minimal DFA is obtained. Many learning algorithms, e.g. Biermann and Feldman's algorithm [9] for learning a regular language from a finite sample, and Angluin's L^* [2] algorithm for learning a regular language using membership and equivalence queries build on this idea.

The idea of trying to distinguish states using right congruence relations is in the essence of many learning algorithms for formalisms richer than regular languages (cf. [1,10,11,14,21,25]). For instance, learning of deterministic weighted automata [7,8] is founded on Fliess's theorem [17] which is a generalization of the Myhill-Nerode theorem to the weighted automata setting.

For ω-regular languages, learning algorithms encounter the problem that the right congruence may not be informative enough. Maler and Pnueli [22] give a polynomial time algorithm for learning the class $\mathbb{DB} \cap \mathbb{DC}$ using membership and

equivalence queries. Their algorithm also works by trying to distinguish all equivalence classes of $\sim_L$ for the unknown language L, and relies on the fact that $\sim_L$ is fully informative for the class $\mathbb{DB} \cap \mathbb{DC}$. The problem of learning the full class of regular ω-languages via membership and equivalence queries was considered open for many years [19]. It was then suggested by Farzan et al. [15] to use the reduction to finite words [12], building on the fact that two ω-languages are equivalent iff they agree on the set of ultimately periodic words. We followed a different route [5,6], and devised an algorithm for learning the full class of ω-regular languages using families of DFAs as acceptors [3,4]. Families of DFAs build on the notion of families of right congruences [23], a set of right congruence relations for a given regular ω-language L which are enough to fully characterize L. Both solutions, however, may encounter big automata in intermediate stages. The solution in [15] may involve DFAs of size $2^n + 2^{2n^2+n}$ where n is the number of states in a non-deterministic Büchi automaton for the language [12], and the solution in [5] may involve families of DFAs of size $m2^m$ where m is the number of states in a minimal deterministic parity automaton for L [16]. We do not go into further details here since they are not used in the current work, which focuses on languages for which the right congruence is fully informative (in hopes of providing a basis for more efficient learning algorithms for these restricted classes). The reader interested in these details is referred to [16].

3. Automata, acceptors and right congruences

This section establishes some basic facts about the automata that we consider. The acceptance conditions for DBAs, DCAs, DPAs, DMAs and DTAs form a partial order of increasing power described in the following lemma.

Lemma 1. *Let L be an ω-language. If L is recognized by a* DBA *or* DCA *then a* DPA *recognizing L can be obtained by changing just the acceptance condition. If L is recognized by a* DPA *then a* DMA *recognizing L can be obtained by changing just the acceptance condition. If L is recognized by a* DMA *then a* DTA *recognizing L can be obtained by changing just the acceptance condition.*

Proof. If L is recognized by the DBA $\mathcal{B} = \langle \Sigma, Q, \lambda, \delta, F \rangle$ then L is also recognized by the DPA $\mathcal{P} = \langle \Sigma, Q, \lambda, \delta, \kappa \rangle$, where $\kappa(q) = 1$ if $q \in F$ and $\kappa(q) = 2$ otherwise. If L is recognized by the DCA $C = \langle \Sigma, Q, \lambda, \delta, F \rangle$, then L is also recognized by the DPA $\mathcal{P} = \langle \Sigma, Q, \lambda, \delta, \kappa \rangle$, where $\kappa(q) = 0$ if $q \in F$, and $\kappa(q) = 1$ otherwise.

If L is recognized by the DPA $\mathcal{P} = \langle \Sigma, Q, \lambda, \delta, \kappa \rangle$ then L is also recognized by the DMA $\mathcal{M} = \langle \Sigma, Q, \lambda, \delta, \alpha, \rangle$, where α contains all subsets of Q whose minimum color is odd. If L is recognized by the DMA $\mathcal{M} = \langle \Sigma, Q, \lambda, \delta, \alpha \rangle$ then L is also recognized by the DTA $\mathcal{T} = \langle \Sigma, Q, \lambda, \delta, \alpha' \rangle$, where α' contains every set of transitions that visits precisely the states of some $F \in \alpha$. □

Turning to language complementation, it is straightforward to see that an ω-language and its complement have the same right congruence, and therefore have the same rightcon automaton.

Lemma 2. *Let L be an ω-language, and let L^c be its complement. Then $\sim_L$ is equal to $\sim_{L^c}$.*

Proof. For any $u, v \in \Sigma^*$, $u \sim_L v$ if and only if $\forall w \in \Sigma^\omega$, $uw \in L \iff vw \in L$, which is true if and only if $\forall w \in \Sigma^\omega$, $uw \notin L \iff vw \notin L$. The latter is the condition for $u \sim_{L^c} v$. Thus $\sim_L$ and $\sim_{L^c}$ are equal. □

The following lemma addresses the question of deriving an acceptor for L^c from an acceptor for L.

Lemma 3. *Let L be an ω-language. If L is recognized by a* DPA *(resp.,* DMA*,* DTA*) $\mathcal{A}$, then a* DPA *(resp.,* DMA*,* DTA*) recognizing L^c can be obtained by changing just the acceptance condition of $\mathcal{A}$. If L is recognized by a* DBA *(resp.,* DCA*) $\mathcal{A}$, then a* DCA *(resp.,* DBA*) recognizing L^c can be obtained by changing just the acceptance condition of $\mathcal{A}$.*

Proof. If L is recognized by a DPA $\mathcal{P} = \langle \Sigma, Q, \lambda, \delta, \kappa \rangle$, then the DPA $P^c = \langle \Sigma, Q, \lambda, \delta, \kappa' \rangle$ recognizes L^c, where $\kappa'(q) = 1 + \kappa(q)$ for all $q \in Q$. This flips the parity of all states and preserves the ordering of the colors. If L is recognized by a DMA $\mathcal{M} = \langle \Sigma, Q, \lambda, \delta, \{F_1, F_2, \ldots, F_k\} \rangle$, then L^c is recognized by the DMA $\mathcal{M}^c = \langle \Sigma, Q, \lambda, \delta, (2^Q \setminus \{F_1, F_2, \ldots, F_k\}) \rangle$. This exchanges the accepting and non-accepting sets of states. If L is recognized by a DTA $\mathcal{T} = \langle \Sigma, Q, \lambda, \delta, \{T_1, T_2, \ldots, T_k\} \rangle$, then L^c is recognized by the DTA $\mathcal{T}^c = \langle \Sigma, Q, \lambda, \delta, (2^{Q \times \Sigma \times Q} \setminus \{T_1, T_2, \ldots, T_k\}) \rangle$. This exchanges the accepting and non-accepting sets of transitions.

If L is recognized by a DBA $\mathcal{B} = \langle \Sigma, Q, \lambda, \delta, F \rangle$ then L^c is recognized by the DCA $C = \langle \Sigma, Q, \lambda, \delta, F \rangle$. Note that the set F giving the acceptance condition is the same, but its interpretation is changed: a word is accepted if it visits at least one element of F infinitely often versus avoids visiting any elements of F infinitely often. A DCA for L is transformed to a DBA for L^c the same way. □

We show that as is the case in finitary regular languages, every deterministic ω-automaton $\mathcal{D}$ refines the rightcon automaton of the respective language $[\![\mathcal{D}]\!]$, and the automaton for the powerset construction of a non-deterministic ω-automaton $\mathcal{N}$ refines the right congruence of the respective language $[\![\mathcal{N}]\!]$.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

Proposition 4. *Let $\mathcal{D}$ be a deterministic ω-automaton. Then $\sim_{\mathcal{D}}$ refines $\sim_{[\![\mathcal{D}]\!]}$.*

Proof. Assume $u \sim_{\mathcal{D}} v$ for some $u, v \in \Sigma^*$. Thus $\mathcal{D}(u) = \mathcal{D}(v)$. That is, there exists a state q of $\mathcal{D}$ such that reading u or v from the initial state of $\mathcal{D}$ reaches q. Therefore, for every $x \in \Sigma^*$ we have $\mathcal{D}(ux) = \mathcal{D}(vx)$. In particular, for every $w \in \Sigma^\omega$ and every prefix $y \in \Sigma^*$ of w, $\mathcal{D}(uy) = \mathcal{D}(vy)$. Therefore exactly the same set of states are visited infinitely often by $\mathcal{D}$ when reading uw or vw. Thus $uw \in [\![\mathcal{D}]\!]$ iff $vw \in [\![\mathcal{D}]\!]$. (Note that this is true for all the acceptance conditions considered by the different ω-automata.) Thus $u \sim_{[\![\mathcal{D}]\!]} v$. □

Let $\mathcal{N}$ be a non-deterministic automaton. We use $\mathcal{P}_{\mathcal{N}}$ to denote the deterministic automaton obtained from $\mathcal{N}$ by applying the powerset construction to $\mathcal{N}$. That is, if $\mathcal{N} = \langle \Sigma, Q, \lambda, \delta \rangle$ then $\mathcal{P}_{\mathcal{N}} = \langle \Sigma, 2^Q, \{\lambda\}, \delta' \rangle$ where $\delta'(S, a) = \cup_{q \in S} \delta(q, a)$ for any $S \subseteq Q$ and $a \in \Sigma$.

Proposition 5. *Let $\mathcal{N}$ be a non-deterministic ω-automaton. Then $\sim_{\mathcal{P}_{\mathcal{N}}}$ refines $\sim_{[\![\mathcal{N}]\!]}$.*

Proof. Suppose $u \sim_{\mathcal{P}_{\mathcal{N}}} v$. Then $\mathcal{P}_{\mathcal{N}}$ reaches the same set S of states of $\mathcal{N}$ on u and on v. If $w \in \Sigma^\omega$ is such that uw is in the language of $\mathcal{N}$, then there exists a state $q \in S$ that $\mathcal{N}$ reaches after reading u such that there is an accepting computation of $\mathcal{N}$ from q on w. Then $\mathcal{N}$ also reaches q after reading v and vw is also in the language of $\mathcal{N}$. Thus $u \sim_{[\![\mathcal{N}]\!]} v$. □

4. Regular ω-languages with a trivial right congruence

If L is an arbitrary ω-language such that $|\sim_L| = 1$, we say that the right congruence and the rightcon automaton of L are *trivial*. Staiger [30] gave a characterization of arbitrary ω-languages that have a trivial right congruence, which enabled him to prove that there are $2^{2^{\aleph_0}}$ such languages. In this section we give a semantic characterization of the class of regular ω-languages that have a trivial right congruence. We start with some basic observations.

The property of having a trivial right congruence can be characterized using the notion of prefix-independence, where an ω-language L is *prefix-independent* if and only if for all $v \in \Sigma^*$ and $w \in \Sigma^\omega$, $w \in L \iff vw \in L$. Note that this notion is symmetric in the sense that a prefix-independent language L is closed under taking (infinite) suffixes and adding (finite) prefixes: $L = \{uw \mid u \in \Sigma^* \wedge w \in L\} = \{w \mid \exists u \in \Sigma^* (uw \in L)\}$.

Proposition 6. *Let L be an ω-language. Then L has a trivial right congruence if and only if L is prefix-independent.*

Proof. Suppose L has a trivial right congruence. Then for any $v \in \Sigma^*$ and $w \in \Sigma^\omega$, $\varepsilon \sim_L v$, so $w \in L \iff vw \in L$, and L is prefix-independent. Conversely, suppose L is prefix-independent. Let $u, v \in \Sigma^*$ and $w \in \Sigma^\omega$. Then $w \in L \iff uw \in L$ and $w \in L \iff vw \in L$, so $uw \in L \iff vw \in L$, so $u \sim_L v$ and L has a trivial right congruence. □

The class of ω-languages that have trivial right congruences is closed under Boolean operations.

Proposition 7. *Let L_1 and L_2 be ω-languages that have trivial right congruences. Then L_1^c, $L_1 \cap L_2$ and $L_1 \cup L_2$ also have trivial right congruences.*

Proof. By Lemma 2, L_1 and its complement L_1^c have the same right congruence, so the right congruence of L_1^c must be trivial. Let $v \in \Sigma^*$ and $w \in \Sigma^\omega$. Then $w \in L_1 \cap L_2$ iff $w \in L_1$ and $w \in L_2$ iff $vw \in L_1$ and $vw \in L_2$ iff $vw \in L_1 \cap L_2$, so $L_1 \cap L_2$ is prefix-independent and therefore has a trivial right congruence. Since $L_1 \cup L_2 = (L_1^c \cap L_2^c)^c$, $L_1 \cup L_2$ must also have a trivial right congruence. □

Next we consider the relation between the class of languages with a trivial right congruence and the class of liveness properties. An ω-language L is a *liveness property* if and only if for every $u \in \Sigma^*$ there exists a $w \in \Sigma^\omega$ such that $uw \in L$. Clearly, a liveness property must be nonempty.

Proposition 8. *Let L be a nonempty ω-language. If L has a trivial right congruence then L is a liveness property, but the converse implication does not hold.*

Proof. Because L is nonempty, there exists some $w \in L$. Assume that L has a trivial right congruence, and is therefore prefix-independent. Then for any $u \in \Sigma^*$, $uw \in L$, which implies that L is a liveness property. For the converse, consider the language L given by $a\Sigma^* a^\omega + b\Sigma^* b^\omega$. This is a liveness property, because any finite prefix u may be followed by either a^ω or b^ω (dependent on the first letter of u) to yield an accepted ω-word. But $a \not\sim_L b$, thus L does not have a trivial right congruence. □

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

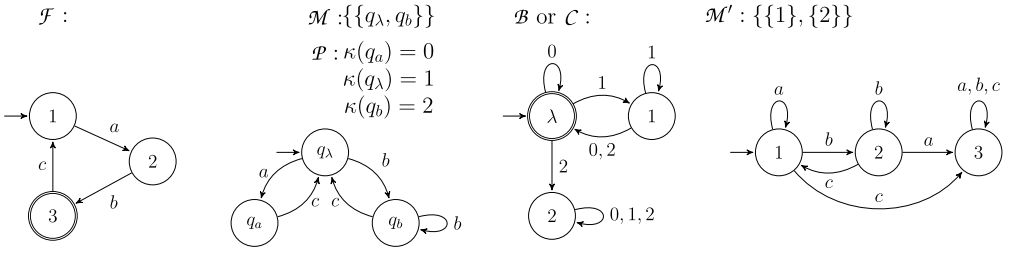

Fig. 3. First, a DBA $\mathcal{F}$ for $L = (abc)^\omega$. Second, a DMA $\mathcal{M}$ and an equivalent DPA $\mathcal{P}$ for a language L_{PM} such that $L_{PM} \in \mathbb{IP} \setminus (\mathbb{IB} \cup \mathbb{IC})$. Third, when regarded as a DBA $\mathcal{B}$ we have $[\![\mathcal{B}]\!] \in \mathbb{IB} \setminus \mathbb{DC}$. When regarded as a DCA $\mathcal{C}$ we have $[\![\mathcal{C}]\!] \in \mathbb{IC} \setminus \mathbb{DB}$. Fourth, a DMA $\mathcal{M}'$ such that $[\![\mathcal{M}']\!] \in \mathbb{IM} \setminus \mathbb{IP}$.

Now we turn to the question of when a regular ω-language has a trivial right congruence. If $L = \Sigma^*(v)^\omega$ for some $v \in \Sigma^*$ then L is prefix-independent and its right congruence is trivial. The following example shows that this is not the case if we replace $(v)^\omega$ with an arbitrary regular ω-language. Take $L = \Sigma^* b a^\omega$. Then $a \not\sim_L b$ since $a \cdot a^\omega \notin L$ whereas $b \cdot a^\omega \in L$. Thus $\sim_L$ has more than one equivalence class. Thus, if $L = \Sigma^* L'$ for some ω-regular language L' then L does not necessarily have a trivial right congruence.

If $L = \Sigma^*(R)^\omega$ for some regular language R, we can show that then $|\sim_L| = 1$. Is this a necessary condition for having a trivial right congruence? As the following example shows, the answer is negative: The language $L = (a + b)^*(a^\omega + b^\omega)$ has $|\sim_L| = 1$, but it is not of the form $(a + b)^* R^\omega$ for any regular set R. However, if $L = \Sigma^* L'$ and L' is a union of languages $L_1 \cup L_2 \cup \cdots L_k$ where each L_i is of the form $(R_i)^\omega$ for some regular language R_i then we obtain a full characterization:

Proposition 9. *A regular ω-language has a trivial right congruence iff $L = \Sigma^*(R_1^\omega + R_2^\omega + \ldots + R_k^\omega)$ for some regular languages $R_1, \ldots, R_k$.*

Proof. Let L be a regular ω-language with a trivial right congruence. Let $\mathcal{M} = \langle \Sigma, Q, \lambda, \delta, \alpha \rangle$ be a DMA for L with no unreachable states. Let $\alpha = \{F_1, F_2, \ldots, F_k\}$. We can assume wlog that all F_i's are strongly connected (because a set F_i that is not an SCC can be omitted). For $i \in [1..k]$ let q_i be some state in F_i and let R_i be the regular set of finite words that traverse $\mathcal{M}$ starting at q_i, ending in q_i, and visiting all states in F_i and no other states.

Let $L' = \Sigma^*(R_1^\omega + R_2^\omega + \ldots + R_k^\omega)$. We claim that $L = L'$. Let $w \in L$. Then $w = uw'$, where for some i, $u \in \Sigma^*$ reaches $q_i \in F_i$ and $w' \in R_i^\omega$, so $w \in L'$. Conversely, if $w \in L'$ then $w = uw'$, where $u \in \Sigma^*$ and for some i, $w' \in R_i^\omega$. Because M contains no unreachable states, there exists $v \in \Sigma^*$ such that v reaches state $q_i \in F_i$. Then $vw' \in L$, so by Proposition 6, $uw' \in L$, that is, $w \in L$.

For the converse, suppose that $R_1, \ldots, R_k$ are regular languages and $L = \Sigma^*(R_1^\omega + \ldots + R_k^\omega)$. Let $u, v \in \Sigma^*$ and $w \in \Sigma^\omega$ and assume $uw \in L$. Because $uw \in L$, there exists an i such that $uw \in \Sigma^* R_i^\omega$. Thus for some $u' \in \Sigma^*$ and elements $w_1, w_2, \ldots$ of R_i, $uw = u' w_1 w_2 \cdots$. Hence there exists $u'' \in \Sigma^*$ and $w'' \in R_i^\omega$ such that $uw = uu''w''$. But then $vw = vu''w''$, and $vu''w'' \in \Sigma^* R_i^\omega$, which implies that $vw \in L$. Conversely, $vw \in L$ implies $uw \in L$, and L has a trivial right congruence. □

5. Regular ω-languages with a fully informative right congruence

We next consider the cases where the right congruence of a language is as informative as it can be for a particular class of automata. If a language L is recognized by a DBA (resp., DCA, DPA, DMA, DTA) whose automaton is isomorphic to the rightcon automaton for L, then we say that L is *fully informative* for DBAs (resp., DCAS, DPAS, DMAS, DTAS). We use $\mathbb{IB}$ (resp., $\mathbb{IC}$, $\mathbb{IP}$, $\mathbb{IM}$, $\mathbb{IT}$) to denote the class of languages that are fully informative for DBAS (resp. DCAS, DPAS, DMAS, DTAS). For instance, the language $(abc)^\omega$ is fully informative for DBAS. Indeed, it is recognized by the DBA $\mathcal{F}$ of Fig. 3 (p. 7) with 4 states (sink state is not shown), where state 1 corresponds to $[abc]$, state 2 to $[a]$, state 3 to $[ab]$ and the sink state to $[aa]$. In this section we consider the expressiveness of these classes.

By Proposition 4, the state congruence of a deterministic ω-acceptor $\mathcal{M}$ refines the right congruence of the language L it recognizes. Thus, to show that $\mathcal{M}$ is isomorphic to the rightcon automaton of L, it suffices to find ω-words that distinguish all the states of $\mathcal{M}$. Another consequence is that if $\mathcal{M}$ and $\mathcal{M}'$ are two deterministic ω-acceptors for the same language, and $\mathcal{M}$ has fewer states than $\mathcal{M}'$, then $\mathcal{M}'$ cannot be isomorphic to the rightcon automaton of the language.

Proposition 10. $(\mathbb{IB} \cup \mathbb{IC}) \subseteq \mathbb{IP} \subseteq \mathbb{IM} \subseteq \mathbb{IT}$.

Proof. This follows directly from Lemma 1, because an ω-acceptor for L that is isomorphic to the rightcon automaton of L can be converted to an ω-acceptor for L with the same automaton and a more powerful acceptance condition. □

When is a language with a trivial right congruence fully informative for a class of ω-automata?

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

Proposition 11. *Let L be a regular ω-language with a trivial right congruence. L is fully informative for* DBAs, DCAs, DPAs *or* DMAs *if and only if L is either the empty language or Σ^ω. L is fully informative with respect to* DTAs *if and only if it is of the form*

$$\Sigma^*(\Gamma_1^\omega + \Gamma_2^\omega + \ldots + \Gamma_k^\omega),$$

where each Γ_i, for $i \in \{1, \ldots, k\}$, is of the form $(\sigma_{i,1} \cdot \Sigma_i^* \cdot \sigma_{i,2} \cdot \Sigma_i^* \cdots \sigma_{i,\ell_i} \cdot \Sigma_i^*)$ and $\Sigma_i = \{\sigma_{i,1}, \sigma_{i,2}, \ldots, \sigma_{i,\ell_i}\}$ is a subset of Σ.

Proof. The rightcon automaton of L has just one state, and the transition on each symbol from Σ maps that state to itself. Because the acceptance conditions for DBAs, DCAs, DPAs and DMAs are specified in terms of the states visited infinitely often, such a structure can only recognize the empty language or Σ^ω.

Consider a one-state DTA $\mathcal{T}$ with acceptance condition $\{T_1, T_2, \ldots, T_k\}$. Let $\Sigma_i = \{\sigma_{i,1}, \ldots, \sigma_{i,\ell_i}\}$ denote the symbols from Σ that occur in the transitions in T_i, and let $\Gamma_i = (\sigma_{i,1} \cdot \Sigma_i^* \cdot \sigma_{i,2} \cdot \Sigma_i^* \cdots \sigma_{i,\ell_i} \cdot \Sigma_i^*)$. Then the set of transitions visited infinitely often by an ω-word w will be T_i if and only if w is in $\Sigma^*(\Gamma_i)^\omega$. □

In [30, Thm. 24 and Ex. 1] Staiger showed that all languages in $\mathbb{DB} \cap \mathbb{DC}$ are in $\mathbb{IM}$ and yet there exists a language in $\mathbb{IM}$ that is not in $\mathbb{DB} \cap \mathbb{DC}$. We first compare $\mathbb{DB} \cap \mathbb{DC}$ with the classes $\mathbb{IB}$ and $\mathbb{IC}$.

Proposition 12. $\mathbb{DB} \cap \mathbb{DC} \subsetneq \mathbb{IB}$ *and* $\mathbb{DB} \cap \mathbb{DC} \subsetneq \mathbb{IC}$.

Proof. Assume $L \in \mathbb{DB} \cap \mathbb{DC}$. By Staiger's theorem we have $L \in \mathbb{IM}$. Suppose $\mathcal{M}$ is a minimal DMA for L. It follows, as explained in the preliminaries, that in $\mathcal{M}$ there is no alternation between accepting and rejecting SCCs along any inclusion chain. Therefore, defining a DBA $\mathcal{B}_\mathcal{M}$ from $\mathcal{M}$ by changing the acceptance condition to $\{q \mid q \in S$ for some accepting SCC $S\}$ gives $[\![\mathcal{B}_\mathcal{M}]\!] = [\![\mathcal{M}]\!]$ and thus $L \in \mathbb{IB}$. Similarly, defining a DCA $C_\mathcal{M}$ from $\mathcal{M}$ by changing the acceptance condition to $\{q \mid q \in S$ for some rejecting SCC $S\}$ gives $[\![C_\mathcal{M}]\!] = [\![\mathcal{M}]\!]$ and thus $L \in \mathbb{IC}$. This completes the inclusion part.

To see that the inclusion is strict for $\mathbb{IB}$ consider the DBA $\mathcal{B}$ from Fig. 3 (p. 7). Because in $\mathcal{B}$ the rejecting SCC $\{1\}$ is a subset of the accepting SCC $\{\lambda, 1\}$, it follows that $[\![\mathcal{B}]\!] \in \mathbb{DB} \setminus \mathbb{DC}$. Thus $[\![\mathcal{B}]\!] \notin \mathbb{DB} \cap \mathbb{DC}$.

Next we show that $\sim_{[\![\mathcal{B}]\!]}$ has at least three equivalence classes. The ω-word 0^ω is accepted only from states λ and 1, and the ω-word $2(0)^\omega$ is accepted only from state 1, so these two experiments distinguish all three states. Since by Proposition 4, $\mathcal{B}$ refines $\sim_{[\![\mathcal{B}]\!]}$, it follows that the automaton for $\sim_{[\![\mathcal{B}]\!]}$ is isomorphic to $\mathcal{B}$, thus $[\![\mathcal{B}]\!] \in \mathbb{IB}$.

The proof for strictness for $\mathbb{IC}$ is dual, using the DCA C in Fig. 3 (p. 7). By Lemma 3, $[\![C]\!]$ is the complement of $[\![\mathcal{B}]\!]$, and therefore has the same right congruence by Lemma 2. Thus $[\![C]\!] \in \mathbb{IC}$. Because in C the accepting SCC $\{1\}$ is a subset of the rejecting SCC $\{\lambda, 1\}$, it follows that $[\![C]\!]$ is not in $\mathbb{DB}$, and thus $[\![C]\!] \in \mathbb{IC} \setminus \mathbb{DB} \cap \mathbb{DC}$. □

Since $\mathbb{IB} \subseteq \mathbb{DB}$ and $\mathbb{IC} \subseteq \mathbb{DC}$, from $\mathbb{DB} \cap \mathbb{DC} \subseteq \mathbb{IB} \cap \mathbb{IC}$ we obtain the following corollary.

Corollary 13. $\mathbb{DB} \cap \mathbb{DC} = \mathbb{IB} \cap \mathbb{IC}$.

We show that the first inclusion in Proposition 10 is proper.

Proposition 14. $\mathbb{IB} \cup \mathbb{IC} \subsetneq \mathbb{IP}$.

Proof. The language L_{PM} recognized by the DPA $\mathcal{P}$ in Fig. 3 (p. 7) is in $\mathbb{IP}$ but not in $\mathbb{IB} \cup \mathbb{IC}$. In this language, $(bc)^\omega$ and $c(bc)^\omega$ are sufficient to distinguish the equivalence classes of $[\epsilon]$, $[a]$, $[b]$, and $[c]$. (The class $[c]$ is the sink state, not shown in the diagram.) The rightcon automaton of L_{PM} is isomorphic to $\mathcal{P}$. Thus, $L_{PM} \in \mathbb{IP}$.

The acceptor $\mathcal{P}$ has an alternation between accepting and rejecting SCCs along the inclusion chain $\{q_b\} \subseteq \{q_\lambda, q_b\} \subseteq \{q_\lambda, q_a, q_b\}$. L_{PM} cannot be accepted by a DBA because the accepting SCC $\{q_\lambda, q_b\}$ is a subset of the rejecting SCC $\{q_\lambda, q_a, q_b\}$, so $L_{PM} \notin \mathbb{IB}$. L_{PM} cannot be accepted by a DCA because the rejecting SCC $\{q_b\}$ is a subset of the accepting SCC $\{q_\lambda, q_b\}$, so $L_{PM} \notin \mathbb{IC}$. □

The second inclusion in Proposition 10 is also proper.

Proposition 15. $\mathbb{IP} \subsetneq \mathbb{IM}$.

Proof. We show that the language L recognized by the DMA $\mathcal{M}'$ in Fig. 3 (p. 7) is in $\mathbb{IM} \setminus \mathbb{IP}$. To see that it is in $\mathbb{IM}$, note that ca^ω distinguishes state 2 from the other states, and a^ω distinguishes states 1 and 3. Thus the rightcon automaton has 3 states, and since by Proposition 4, $\mathcal{M}'$ refines $\mathcal{R}_{[\![\mathcal{M}']\!]}$ they are isomorphic. Assume towards contradiction that $L \in \mathbb{IP}$. Then there must be a DPA isomorphic to the rightcon automaton of L, that is, isomorphic to M'. In that case, state 1 must have an odd color, so that when the set of states visited infinitely often is $\{1\}$ it will accept. For the same reason 2 must have an odd color. But then, when the set of states visited infinitely often is $\{1, 2\}$, the automaton will accept as well, contradicting the definition of L. □

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

The last inclusion in Proposition 10 is also proper.

Proposition 16 *([31]).* $\mathbb{IM} \subsetneq \mathbb{IT}$.

Proof. It is shown in [31] that any DMA can be defined on a DTA with the same structure, and that the converse is not true. As an example, the language $L = (a+b)^*(a^\omega + b^\omega)$ has a trivial right congruence and is recognized by the one-state DTA $\mathcal{T}$ in Fig. 2 (p. 3), so $L \in \mathbb{IT}$. By Proposition 11, no DMA with one state recognizes L, thus L is not in $\mathbb{IM}$. □

The following corollary summarizes these results, which are also depicted in Fig. 1 (p. 2) on the left.

Corollary 17. *The following results regarding inclusion of the* $\mathbb{IX}$ *classes hold.*

$$(\mathbb{DB} \cap \mathbb{DC}) = (\mathbb{IB} \cap \mathbb{IC}),$$
$$\mathbb{IB} \cap \mathbb{IC} \subsetneq \mathbb{IB},$$
$$\mathbb{IB} \cap \mathbb{IC} \subsetneq \mathbb{IC},$$
$$(\mathbb{IB} \cup \mathbb{IC}) \subsetneq \mathbb{IP} \subsetneq \mathbb{IM} \subsetneq \mathbb{IT}.$$

Relation to the Wagner Hierarchy

The last question we address in this section is how complex a language in $\mathbb{IP}$ can be, according to the complexity measure of the Wagner Hierarchy (recalled next). We show that such languages can be arbitrarily complex. That is, the complexity expressed by the hierarchy and the property of having a fully informative right congruence are independent, so that being "complex" does not imply losing the fully informative property.[3] Formally, we show that for every class $\mathbb{DM}^p_{n,m}$ of the Wagner Hierarchy, there is a language $L \in \mathbb{IP}$ that is in $\mathbb{DM}^p_{n,m}$ and not in any proper subclass of the Wagner Hierarchy. First we recall the definition of the Wagner classes.

Let $\mathcal{M} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$ be a DMA. A sequence of SCCs $S_1, S_2, S_3, \cdots, S_k$ is an *inclusion chain* if $S_i \subset S_{i+1}$ for all $1 \leq i < k$ and $S_i \in \alpha$ iff $S_{i+1} \notin \alpha$, i.e. if the sequence alternates the acceptance nature of the SCCs. We define the *positive inclusion measure* of $\mathcal{M}$, denoted $|\mathcal{M}|^+_\subseteq$ as the length of a maximal inclusion chain of $\mathcal{M}$ where S_1 is accepting. Likewise, the *negative inclusion measure* of $\mathcal{M}$, denoted $|\mathcal{M}|^-_\subseteq$ is the length of a maximal inclusion chain of $\mathcal{M}$ where S_1 is rejecting. Note that for any $\mathcal{M}$ the difference between $|\mathcal{M}|^+_\subseteq$ and $|\mathcal{M}|^-_\subseteq$ may be at most one. A sequence of maximal SCCs $M_1, M_2, \ldots, M_m$ is referred to as a *reachability chain* if M_{i+1} is reachable from M_i for every $1 < i \leq m$ and $M_i \in \alpha$ iff $M_{i+1} \notin \alpha$. We use $|\mathcal{M}|_\rightsquigarrow$ to denote the length of a maximal reachability chain in $\mathcal{M}$.

Let n, m be two natural numbers and $p \in \{+, -, \pm\}$. The classes $\mathbb{DM}^p_{n,m}$ of the Wagner hierarchy are defined as follows:

$$\mathbb{DM}^+_{k,m} = \{L \mid \exists \text{ DMA } \mathcal{M} \text{ s.t. } [\![\mathcal{M}]\!] = L, |\mathcal{M}|^+_\subseteq \leq k, |\mathcal{M}|^-_\subseteq < k \text{ and } |\mathcal{M}|_\rightsquigarrow = m\}$$
$$\mathbb{DM}^-_{k,m} = \{L \mid \exists \text{ DMA } \mathcal{M} \text{ s.t. } [\![\mathcal{M}]\!] = L, |\mathcal{M}|^+_\subseteq < k, |\mathcal{M}|^-_\subseteq \leq k \text{ and } |\mathcal{M}|_\rightsquigarrow = m\}$$
$$\mathbb{DM}^\pm_{k,m} = \{L \mid \exists \text{ DMA } \mathcal{M} \text{ s.t. } [\![\mathcal{M}]\!] = L, |\mathcal{M}|^+_\subseteq \leq k, |\mathcal{M}|^-_\subseteq \leq k \text{ and } |\mathcal{M}|_\rightsquigarrow = m\}$$

Proposition 18. *Let* n, m *be two natural numbers and* $p \in \{+, -, \pm\}$. *There exists a language* $L \subseteq \Sigma^\omega$ *for* $\Sigma = \{a, b\}$ *such that* $L \in \mathbb{IT} \cap \mathbb{IM} \cap \mathbb{IP} \cap \mathbb{DM}^p_{n,m}$ *and* $L \notin \mathbb{DM}^p_{n-1,m}$ *and* $L \notin \mathbb{DM}^p_{n,m-1}$.

Proof. Consider the DMA $\mathcal{D}^+_{n,m}$ in Fig. 4 (p. 10) with acceptance condition $\mathcal{F} = \{\{0^i, 1^i, \ldots, j^i\} \mid j \text{ is odd iff } i \text{ is odd}\}$. For instance, $\{0^0\} \in \mathcal{F}$, $\{0^0, 1^0, 2^0\} \in \mathcal{F}$, and $\{0^3, 1^3\} \in \mathcal{F}$ but $\{1^0\} \notin \mathcal{F}$ and $\{0^0, 1^0\} \notin \mathcal{F}$. It strictly belongs to the Wagner hierarchy class $\mathbb{DM}^+_{n,m}$ since $|\mathcal{M}|^+_\subseteq = n$ and $|\mathcal{M}|_\rightsquigarrow = m$. To see that it is in $\mathbb{IM}$, we show for each state, a word that distinguishes it from other states. We fix an order between the states: a state k^ℓ is smaller than $k'^{\ell'}$ if either $\ell < \ell'$ or $\ell = \ell'$ and $k < k'$. Thus the last state n^m is the biggest in this order. The word a^ω distinguishes the last state n^m from all states on odd rows, and the word $(ab)^\omega$ distinguishes m^n from all states on even rows. For $k \in [0..n]$, the word $b^{n-k}a^\omega$ distinguishes state k^m from all smaller states on odd rows, and the word $b^{n-k}(ab)^\omega$ distinguishes it from all smaller states on even rows. Finally, for $k \in [0..n]$ and $\ell \in [0..m]$ the word $(b^{n+1})^{m-\ell}b^{n-k}a^\omega$ distinguishes state k^ℓ from all smaller states on odd rows, and the word $(b^{n+1})^{m-\ell}b^{n-k}(ab)^\omega$ distinguishes it from all smaller states on even rows. This shows that the rightcon automaton has $(n+1)(m+1)$ states, and thus $[\![\mathcal{D}^+_{n,m}]\!] \in \mathbb{IM}$. It is also in $\mathbb{IP}$, since we can define a DPA on the same structure, by assigning state k^ℓ the color k if ℓ is odd, and $k+1$ if ℓ is even. It is in $\mathbb{IT}$ since $\mathbb{IM} \subset \mathbb{IT}$.

The proof for $\mathbb{DM}^-_{n,m}$ is symmetric, and the and the result for $\mathbb{DM}^\pm_{n,m}$ immediately follows from those for $\mathbb{DM}^+_{n,m}$ and $\mathbb{DM}^-_{n,m}$. □

[3] A similar result is mentioned without a proof in a footnote in the paper of Maler and Staiger [23], with credit to "N. Gutleben (personal communication)".

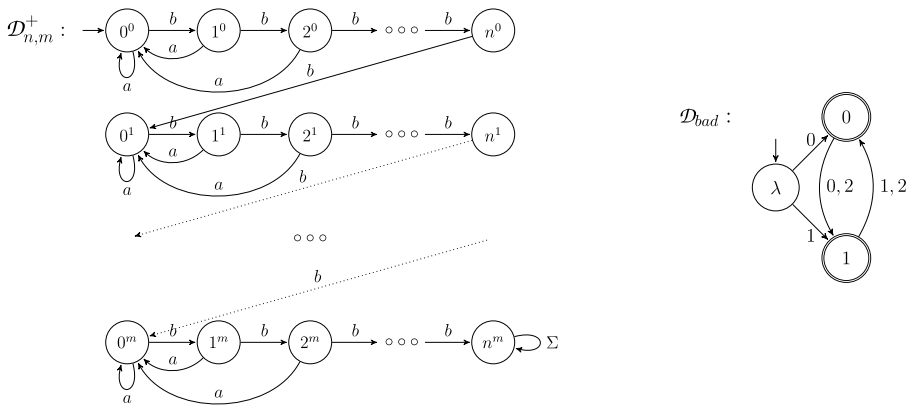

Fig. 4. On the left, a representative example for the Wagner class $\mathbb{DM}^+_{n,m}$. The acceptance condition is $\mathcal{F} = \{\{0^i, 1^i, \ldots, j^i\} \mid j \text{ is odd iff } i \text{ is odd}\}$. On the right a DBA $\mathcal{D}_{bad}$ in $\mathbb{IB} \cap \mathbb{IC}$ that is not respective of its right congruence.

6. Respective of the right congruence

As mentioned above, one of the motivations for studying classes of languages that are isomorphic to the right congruence is in the context of learning an unknown language. In this context, positive and negative examples (ω-words labeled by their membership in the language) should help a learning algorithm to infer an automaton for the language. Consider the positive example $(ab)^\omega$ for an unknown language L. Intuitively, we expect that a minimal automaton for L would have a loop of size 2 in which the word ab cycles. This is not necessarily the case, as shown by the language $L = (aba + bab)^\omega$, whose minimal DBA $\mathcal{B}_{\bowtie}$ is given in Fig. 5 (p. 14). In the case of regular languages of finite words, if we regard the automaton $\mathcal{B}_{\bowtie}$ as a DFA, we note that ab, $abab$ are negative examples, while $ababab$ is a positive example. From this a learning algorithm can clearly infer the smallest loop on which ab cycles is of length 6, and not 2. But in the case of ω-languages there are no negative ω-words that can provide such information. We thus define a class of languages in which if $u(v)^\omega$ is a positive example for L, then a minimal automaton for L has a cycle of length at most $|v|$ in which $u(v)^\omega$ loops.

Definition 19 *(Respective of $\sim_L$).* A language L is said to be *respective* of its right congruence if $\exists n_0 \in \mathbb{N}.\ \forall n > n_0.\ \forall x, u \in \Sigma^*.$ $x(u)^\omega \in L$ implies $x(u)^n \sim_L x(u)^{n+1}$.

Intuitively, a language that is respective of its right congruence, can "delay" entering a loop as much as needed, but once it loops on a periodic part, it loops on the smallest period possible. To see why this holds, take an ultimately periodic word $x(y)^\omega$ and assume its smallest periodic part is v, so that $x(y)^\omega = u(v)^\omega$ for some word u. If the automaton eventually, say after n_0 letters of $u(v)^\omega$ has been read, loops in a cycle of length of $|v|$, then for any $n > n_0$ we have that $u(v)^n \sim_L u(v)^{n+1}$. That is, the automaton ends up in the same state after reading another substring of size $|v|$. Note that this property does not depend on any ω-automaton accepting the language.

Being respective of the right congruence does not entail having an ω-automaton that is isomorphic to the right congruence. Any language L with $|\sim_L| = 1$ is (trivially) respective of its right congruence. By Proposition 9, $L = (a + b)^*(aba)^\omega$ has $|\sim_L| = 1$, but L is not in $\mathbb{IT}$, $\mathbb{IM}$, $\mathbb{IP}$, $\mathbb{IB}$ or $\mathbb{IC}$, because every ω-automaton accepting L requires more than one state. We thus concentrate on languages which are both isomorphic to the rightcon automaton and respective of their right congruence. We use $\mathbb{R}$ for the class of languages that are respective of their right congruence, and $\mathbb{RX} = \mathbb{R} \cap \mathbb{IX}$ for $\mathbb{X} \in \{\mathbb{B}, \mathbb{C}, \mathbb{P}, \mathbb{M}, \mathbb{T}\}$. That is, $\mathbb{RB}$, $\mathbb{RC}$, $\mathbb{RP}$, $\mathbb{RM}$ and $\mathbb{RT}$ stand for the classes of languages that are respective of their right congruence and reside in $\mathbb{IB}$, $\mathbb{IC}$, $\mathbb{IP}$, $\mathbb{IM}$ and $\mathbb{IT}$, respectively. By definition, thus, $\mathbb{RX} \subseteq \mathbb{IX}$ for $\mathbb{X} \in \{\mathbb{B}, \mathbb{C}, \mathbb{P}, \mathbb{M}, \mathbb{T}\}$. We show that these inclusions are strict as a corollary of the following.

Proposition 20. *There exists a language in $\mathbb{IB} \cap \mathbb{IC}$ that is not respective of its right congruence.*

Proof. The DBA $\mathcal{D}_{bad}$ in Fig. 4 (p. 10) recognizes a language L in $\mathbb{IB} \cap \mathbb{IC}$ since it has no alternations between accepting and rejecting SCCs along an inclusion chain. To see that $\mathcal{D}_{bad}$ is isomorphic to the rightcon automaton of L, note that the ω-words $(01)^\omega$, $(10)^\omega$ and $0(01)^\omega$ distinguish the equivalence classes $[\epsilon]$, $[0]$, $[1]$ and $[2]$. To see that L is not respective of its right congruence, consider the pair $(0, 2)$. The ω-word $0(2)^\omega$ is in L, but $0(2)^n$ and $0(2)^{n+1}$ reach different states of $\mathcal{D}_{bad}$ for every $n \in \mathbb{N}$. Since the automaton has been shown to be isomorphic to the right congruence, the state the automaton reaches after reading a certain word dictates the equivalence class the word belongs to. Therefore the fact that the two words $0(2)^n$ and $0(2)^{n+1}$ reach different states implies $0(2)^n \not\sim_L 0(2)^{n+1}$, contradicting the property of being respective of the right congruence. □

Corollary 21. $\mathbb{RX} \subsetneq \mathbb{IX}$ *for* $\mathbb{X} \in \{\mathbb{B}, \mathbb{C}, \mathbb{P}, \mathbb{M}, \mathbb{T}\}$.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

Proof. The language L accepted by $\mathcal{D}_{bad}$ is in $\mathbb{IB} \cap \mathbb{IC}$, and (by Proposition 10) is also in $\mathbb{IB}$, $\mathbb{IC}$, $\mathbb{IP}$, $\mathbb{IM}$ and $\mathbb{IT}$. Because L is not respective of its right congruence, it is not in $\mathbb{RB}$, $\mathbb{RC}$, $\mathbb{RP}$, $\mathbb{RM}$ or $\mathbb{RT}$. □

The following is a direct consequence of Proposition 10 and the definition of the $\mathbb{RX}$ classes.

Corollary 22. $(\mathbb{RB} \cup \mathbb{RC}) \subseteq \mathbb{RP} \subseteq \mathbb{RM} \subseteq \mathbb{RT}$.

The following two propositions complete the picture of the inclusions relating the $\mathbb{RX}$ classes.

Proposition 23. $(\mathbb{RB} \cap \mathbb{RC}) \subsetneq \mathbb{RB}$ *and* $(\mathbb{RB} \cap \mathbb{RC}) \subsetneq \mathbb{RC}$.

Proof. The inclusions are evident; we show they are proper. The language $[\![\mathcal{B}]\!]$ recognized by the DBA $\mathcal{B}$ in Fig. 3 (p. 7) was shown to be in $\mathbb{IB} \setminus (\mathbb{DB} \cap \mathbb{DC})$ in Proposition 12, and therefore is in $\mathbb{IB}$ but not in $(\mathbb{RB} \cap \mathbb{RC})$. To see that it is respective of its right congruence, assume $x(u)^\omega \in [\![\mathcal{B}]\!]$. Then no prefix of $x(u)^\omega$ reaches state 2. If the nonempty word u ends with the symbol 0 or 2, then $\mathcal{B}(x(u)^k)$ is state λ for all $k \geq 1$. If u ends with the symbol 1, then $\mathcal{B}(x(u)^k)$ is state 1 for all $k \geq 1$. In either case, $x(u)^n \sim_{[\![\mathcal{B}]\!]} x(u)^{n+1}$ for all $n \geq 1$.

Similarly, the proof of Proposition 12 showed that the language $[\![\mathcal{C}]\!]$ recognized by the DCA $\mathcal{C}$ in Fig. 3 (p. 7) is in $\mathbb{IC} \setminus (\mathbb{DB} \cap \mathbb{DC})$, and therefore in $\mathbb{IC} \setminus (\mathbb{RB} \cap \mathbb{RC})$. To see that $[\![\mathcal{C}]\!]$ is also respective of its right congruence, assume $x(u)^\omega \in [\![\mathcal{C}]\!]$. The set S of states visited infinitely often by $x(u)^\omega$ must either be $\{1\}$ or $\{2\}$. If it is $\{1\}$, then $u = 1^k$ for some positive integer k, and it must be that $\mathcal{C}(x(u)^k)$ is state 1 for every $k \geq 1$. However, if $S = \{2\}$, we claim that $\mathcal{C}(xuu)$ must be state 2. Consider the three states $\mathcal{C}(x)$, $\mathcal{C}(xu)$ and $\mathcal{C}(xuu)$. If at least one of these is state 2, then $\mathcal{C}(xuu)$ is state 2. If none of them is state 2, then one of the states λ and 1 must occur at least twice, which would imply that all prefixes of $x(u)^\omega$ would fail to reach state 2, contradicting the assumption that $S = \{2\}$. Hence, in either case, $x(u)^n \sim_{[\![\mathcal{C}]\!]} x(u)^{n+1}$ for all $n \geq 2$. □

Proposition 24. $(\mathbb{RB} \cup \mathbb{RC}) \subsetneq \mathbb{RP}$, $\mathbb{RP} \subsetneq \mathbb{RM}$ *and* $\mathbb{RM} \subsetneq \mathbb{RT}$.

Proof. The language L_{PM} recognized by the DPA $\mathcal{P}$ in Fig. 3 (p. 7) was shown to be in $\mathbb{IP}$ and not in $(\mathbb{IB} \cup \mathbb{IC})$ in the proof of Proposition 14. To see that it is respective of its right congruence, consider any $x(u)^\omega \in L_{PM}$. Then u must be a nonempty word that does not contain the symbol a. If u ends in the symbol b then $\mathcal{P}(x(u)^k)$ must be the state q_b for any $k \geq 1$, and if u ends in the symbol c then $\mathcal{P}(x(u)^k)$ must be the state q_λ for any $k \geq 1$, so in either case, $x(u)^n \sim_{L_{PM}} x(u)^{n+1}$ for any $n \geq 1$. Thus, L_{PM} is respective of its right congruence, and is in $\mathbb{RP}$ but not in $\mathbb{RB}$ or $\mathbb{RC}$.

The language $L = [\![\mathcal{M}']\!]$ accepted by the DMA $\mathcal{M}'$ in Fig. 3 (p. 7) was shown to be in $\mathbb{IM}$ and not in $\mathbb{IP}$ in the proof of Proposition 15. To see that L is respective of its right congruence note that $L = (a + bb^*c)^*(a^\omega + b^\omega)$. In particular if $x(u)^\omega \in L$ then u must be a^k or b^k for some positive integer k. In the former case, $\mathcal{M}'(x)$ must be state 1, and in the latter case, $\mathcal{M}'(x)$ must be state 1 or 2, and in either case, we have $xu^n \sim_L xu^{n+1}$ for all $n \geq 2$.

The language accepted by the DTA $\mathcal{T}$ in Fig. 2 (p. 3) is $L = (a+b)^*(a^\omega + b^\omega)$. It was shown to be in $\mathbb{IT}$ and not in $\mathbb{IM}$ in the proof of Proposition 16. L is respective of its right congruence because its right congruence is trivial. Thus, $L \in \mathbb{RT}$. □

Fig. 1 (p. 2) on the right summarizes the inclusion results for the $\mathbb{RX}$ classes. While it is notable that the requirement of being respective of the right congruence constitutes a restriction, there exist languages which are respective of their right congruence in every class of the Wagner Hierarchy. That is, the property of being respective of the right congruence is also "orthogonal" to the Wagner complexity measure.

Proposition 25. *Let n, m be two natural numbers and $p \in \{+, -, \pm\}$. There exists a language $L \subseteq \Sigma^\omega$ for $\Sigma = \{a, b\}$ such that $L \in \mathbb{RT} \cap \mathbb{RM} \cap \mathbb{RP} \cap \mathbb{DM}^p_{n,m}$ and $L \notin \mathbb{DM}^p_{n-1,m}$ and $L \notin \mathbb{DM}^p_{n,m-1}$.*

Proof. Consider again the DMA $\mathcal{D}^+_{n,m}$ in Fig. 4 (p. 10) and let $L^+_{n,m}$ be the language that it recognizes. We have established in the proof of Proposition 18 that $[\![\mathcal{D}^+_{n,m}]\!] \in \mathbb{IM} \cap \mathbb{IP} \cap \mathbb{IT}$, $[\![\mathcal{D}^+_{n,m}]\!] \notin \mathbb{DM}^p_{n-1,m}$ and $[\![\mathcal{D}^+_{n,m}]\!] \notin \mathbb{DM}^p_{n,m-1}$. Take $n_0 = (m+1)(n+1)$. Consider the state that $\mathcal{D}^+_{n,m}$ reaches after reading xu^{n_0}. If this state is n^m then clearly for any $n' > n_0$, $xu^{n'}$ also reaches states n^m. Otherwise there is an a following the longest subsequence of b^* in u. The state that $\mathcal{D}^+_{n,m}$ reaches after reading xu^{n_0} depends on (a) the longest subsequence of b's in x (b) the longest subsequence of b's in u and (c) the number of consecutive b's in the rightmost subsequence of b's in u. Since the parameter (a) depends only on x and the parameters (b) and (c) remain the same in u^i for any $i \in \mathbb{N}$ we have that $xu^{n_0+i} \sim_{L^+_{n,m}} xu^{n_0+i+1}$. Thus, the accepted language is respective of its right congruence. □

Relation to liveness properties Proposition 8 shows that a nonempty ω-language with a trivial right congruence is necessarily a liveness property, but there are liveness properties with nontrivial right congruences. We give a method of converting many ω-acceptors to related liveness properties. Let $\mathcal{A} = \langle \Sigma, Q, \lambda, \delta, \alpha \rangle$ be a DBA (resp., DCA, DPA, DMA, DTA). Let d be a

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

symbol not in Σ and define $\Sigma_d = \Sigma \cup \{d\}$. Let $L_d = \Sigma^* d \Sigma_d^\omega$ be the set of words over Σ_d that contain the symbol d. Let q_d be a state not in Q and define $Q_d = Q \cup \{q_d\}$. Let δ be extended to δ_d by defining $\delta_d(q, d) = q_d$ for all $q \in Q$ and $\delta_d(q_d, \sigma) = q_d$ for all $\sigma \in \Sigma_d$. Finally, let α_d be the acceptance condition augmented to accept the SCC $\{q_d\}$. For a DBA, this is achieved by marking q_d; for a DCA, by not marking q_d; for a DPA, by assigning q_d an odd color; for a DMA, by including $\{q_d\}$ as an accepting set of states; for a DTA, by including every non-empty subset of $\{(q_d, \sigma, q_d) \mid \sigma \in \Sigma_d\}$ as an accepting set of transitions. Then define the DBA (resp., DCA, DPA, DMA, DTA) $\mathcal{A}_d = \langle \Sigma_d, Q_d, \lambda, \delta_d, \alpha_d \rangle$. This construction has the following properties.

Proposition 26. *Let $\mathcal{A} = \langle \Sigma, Q, \lambda, \delta, \alpha \rangle$ be a DBA (resp., DCA, DPA, DMA, DTA) such that for no state $q \in Q$ is the set of ω-words accepted from q equal to Σ^ω. Then $\mathcal{A}_d$ is a DBA (resp. DCA, DPA, DMA, DTA) and $[\![\mathcal{A}_d]\!] = [\![\mathcal{A}]\!] \cup L_d$ is a liveness property. Moreover, if $\mathcal{A}$ is isomorphic to its rightcon automaton, then so is $\mathcal{A}_d$, and if $[\![\mathcal{A}]\!]$ is respective of its right congruence, then so is $[\![\mathcal{A}_d]\!]$.*

Proof. Let $L = [\![\mathcal{A}]\!]$ and $L' = [\![\mathcal{A}_d]\!]$. By construction, $\mathcal{A}_d$ is an ω-acceptor of the same type as $\mathcal{A}$. Let $w \in \Sigma_d^\omega$. If w does not contain the symbol d, then the run of $\mathcal{A}_d$ on w is the same as the run of $\mathcal{A}$ on w, and their acceptance conditions are the same for runs that do not involve state q_d, so $\mathcal{A}_d$ accepts w iff $\mathcal{A}$ accepts w. If w does contain the symbol d, then the prefix u of w before the first occurrence of d reaches some state $q \in Q$, and then the following d reaches q_d. Thus, the set of states visited infinitely often by the run of $\mathcal{A}_d$ on w is $\{q_d\}$, which is accepting by construction. Then L' is a liveness property, because for any $u \in \Sigma_d^*$, $ud^\omega \in L'$.

Assume $\mathcal{A}$ is isomorphic to its rightcon automaton. Then there are ω-words in Σ^ω distinguishing every pair of states in Q in $\mathcal{A}$, which continue to distinguish those states in $\mathcal{A}_d$. To distinguish q_d from an arbitrary $q \in Q$, we note that every element of Σ_d^ω is accepted from q_d, but there is by assumption, some $w \in \Sigma^\omega$ that is not accepted from q in $\mathcal{A}$, and therefore w is not accepted from q in $\mathcal{A}_d$. Thus all the pairs of $\mathcal{A}_d$ may be distinguished, and A_d is isomorphic to its rightcon automaton.

Assume L is respective of its right congruence with bound n_0. Note that if $u \in \Sigma^*$ then $[u]_L = [u]_{L'}$, and there is one more equivalence class, containing all the strings that contain the symbol d. Let $x(v)^\omega \in L'$ for some $x \in \Sigma_d^*$ and $v \in \Sigma_d^+$. If neither x nor v contains the symbol d, then for all $n \geq n_0$ we have that $x(v)^n \sim_L x(v)^{n+1}$ because L is respective of its right congruence, which implies $x(v)^n \sim_{L'} x(v)^{n+1}$. If either x or v contains the symbol d, then for all $n \geq 1$, the equivalence class of $x(v)^n$ is the set of all words that contain the symbol d, so $x(v)^n \sim_{L'} x(v)^{n+1}$. Thus, L' is respective of its right congruence. □

Using this proposition we can show that there are liveness properties with a fully informative right congruence, that are also respective of their right congruence in any class of the Wagner hierarchy. It can also be used to add the requirement of liveness to any previously obtained result distinguishing two classes. As one example, this construction could be applied to the DMA $\mathcal{M}'$ in Fig. 3 (p. 7) to yield a DMA $\mathcal{M}'_d$ recognizing a liveness property in $\mathbb{IM} \setminus \mathbb{IP}$.

Relation to non-counting languages The definition of respective of $\sim_L$ is reminiscent of the definition of non-counting languages [13]. A language $L \subseteq \Sigma^\omega$ is said to be *non-counting* iff $\exists n_0 \in \mathbb{N}.\ \forall n > n_0.\ \forall u, v \in \Sigma^*,\ w \in \Sigma^\omega.\ u(v)^n w \in L \iff u(v)^{n+1} w \in L$.

Proposition 27. *If L is non-counting then L is respective of its right congruence.*

Proof. Let $L \subseteq \Sigma^\omega$ be non-counting. Then $\exists n_0 \in \mathbb{N}.\ \forall n > n_0.\ \forall u, v \in \Sigma^*.\ \forall w \in \Sigma^\omega.\ u(v)^n w \in L \iff u(v)^{n+1} w \in L$. Assume towards contradiction that L is not respective of its right congruence. Thus, for all $n_0 \in \mathbb{N}$ there exist $n > n_0$ and $x, u \in \Sigma^*$ such that $x(u)^\omega \in L$ yet such that $x(u)^n \not\sim_L x(u)^{n+1}$. Hence, there exists $w \in \Sigma^\omega$ such that $x(u)^n w \in L$ iff $x(u)^{n+1} w \notin L$, contradicting that L is non-counting. □

The converse does not hold.

Proposition 28. *There exist languages that are respective of $\sim_L$ but are not non-counting.*

Proof. The language $L = (aa)^* b^\omega$ is respective of $\sim_L$ but is not non-counting. To see that it is not non-counting note that for every even n we have that $a^n b^\omega \in L$ while $a^{n+1} b^\omega \notin L$. To see that it is respective of $\sim_L$, let $x(u)^\omega \in L$. Then $u = b^k$ for some $k \geq 1$ and $x = (aa)^i b^j$ for some $i, j \in \mathbb{N}$. For all $w \in \Sigma^\omega$ and all $n \geq 1$, $x(u)^n w \in L$ iff $w = b^\omega$ iff $x(u)^{n+1} w \in L$. □

One of the most commonly used temporal logics is Linear temporal logic (LTL) [26]. LTL formulas are non-counting [18,33,13,27]. However, there are LTL formulas that characterize languages that are not in $\mathbb{IT}$ (and thus not in any of the $\mathbb{I}$ classes). Indeed, the formula $FG(a \vee Xa)$ characterizes the language $L = \Sigma^*(a + \Sigma a)^\omega$ and by Proposition 9, $|\sim_L| = 1$.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

Table 1
Boolean closure properties summary, with references to the respective Propositions.

Class	$\mathbb{DB} \cap \mathbb{DC}$	$\mathbb{IB}$	$\mathbb{IC}$	$\mathbb{IP}$	$\mathbb{IM}$	$\mathbb{IT}$	$\mathbb{RB}$	$\mathbb{RC}$	$\mathbb{RP}$	$\mathbb{RM}$	$\mathbb{RT}$
Complement	✓ Pr. 29	✗ Pr. 31	✗ Pr. 31	✓ Pr. 30	✓ Pr. 30	✓ Pr. 30	✗ Pr. 32	✗ Pr. 32	✗ Pr. 32	✗ Pr. 32	✗ Pr. 32
Union	✓ Pr. 29	✗ Pr. 34	✗ Pr. 33	✗ Pr. 33	✗ Pr. 33	✗ Pr. 38	✗ Pr. 39	✗ Pr. 35	✗ Pr. 35	✗ Pr. 35	✗ Pr. 38
Intersection	✓ Pr. 29	✗ Pr. 33	✗ Pr. 34	✗ Pr. 33	✗ Pr. 33	✗ Pr. 37	✗ Pr. 36	✗ Pr. 39	✗ Pr. 36	✗ Pr. 36	✗ Pr. 37

7. Closure properties

We examine what Boolean closure properties hold or do not hold for the classes of languages recognizable by an automaton isomorphic to the rightcon automaton, and by automata that are also respective of the right congruence. Table 1 summarizes the results for Boolean closure properties of the classes we consider.

It is a well known result that weak regular ω-languages are closed under all Boolean operations as stated by the following proposition.

Proposition 29 *(cf. [24]). The class $\mathbb{DB} \cap \mathbb{DC}$ is closed under the Boolean operations complementation, union and intersection.*

7.1. Complementation

Proposition 30. *$\mathbb{IT}$, $\mathbb{IM}$ and $\mathbb{IP}$ are closed under complementation.*

Proof. The result for $\mathbb{IM}$ was established in [31]. Let L be in $\mathbb{IP}$ (resp., $\mathbb{IM}$, $\mathbb{IT}$). Then there exists a DPA (resp., DMA, DTA) $\mathcal{A}$ recognizing L such that $\mathcal{A}$ is isomorphic to the rightcon automaton of L. By Lemma 3, by changing just the acceptance condition of $\mathcal{A}$ we may obtain an acceptor $\mathcal{A}^c$ of the same type for the complement language L^c. Then $\mathcal{A}^c$ is isomorphic to the rightcon automaton of L, which by Lemma 2 is the same as the rightcon automaton of L^c. Thus, L^c is in $\mathbb{IP}$ (resp., $\mathbb{IM}$, $\mathbb{IT}$). □

Proposition 31. *$\mathbb{IB}$ and $\mathbb{IC}$ are not closed under complementation.*

Proof. Consider the DBA $\mathcal{B}$ in Fig. 3 (p. 7) recognizing language L. When the same acceptor is considered as a DCA C, we have that C recognizes L^c by the proof of Lemma 3. In the proof of Proposition 12 it was shown that $L \in \mathbb{IB} \setminus \mathbb{DC}$ and $L^c \in \mathbb{IC} \setminus \mathbb{DB}$. Thus, $L \in \mathbb{IB}$ while $L^c \notin \mathbb{DB}$, and since $\mathbb{DB}$ subsumes $\mathbb{IB}$, $L^c \notin \mathbb{IB}$. Likewise $L^c \in \mathbb{IC}$ while $L \notin \mathbb{DC}$ and thus $L \notin \mathbb{IC}$, so neither $\mathbb{IB}$ nor $\mathbb{IC}$ is closed under complement. □

Because the definition of being respective of the right congruence is not symmetric for L and L^c, the $\mathbb{RX}$ classes behave somewhat differently from the $\mathbb{IX}$ classes with respect to complementation.

Proposition 32. *$\mathbb{RB}$, $\mathbb{RC}$, $\mathbb{RP}$, $\mathbb{RM}$ and $\mathbb{RT}$ are not closed under complementation.*

Proof. Consider the DBA $\mathcal{B}$ in Fig. 6 (p. 15). The ω-words $ba(ac)^\omega$, $a(ac)^\omega$, $(ac)^\omega$ and $(ca)^\omega$ distinguish its five states (the four shown in the figure, and the sink state). This shows that $[\![\mathcal{B}]\!] \in \mathbb{IB} \subset \mathbb{IP} \subset \mathbb{IM} \subset \mathbb{IT}$. To see that it is respective of its right congruence, if $x(y)^\omega$ is in $[\![\mathcal{B}]\!]$ then y must traverse the cycle of states 3 and 4. Thus for some n_0, $\mathcal{B}(x(y)^{n_0}) = 3$ and $y = (ac)^k$ for some $k \geq 1$, or $\mathcal{B}(x(y)^{n_0}) = 4$ and $y = (ca)^k$ for some $k \geq 1$. In either case, $\mathcal{B}(x(y)^n) = \mathcal{B}(x(y)^{n+1})$ for all $n \geq n_0$. However, its complement is not respective of its right congruence. The word b^ω is in the complement language $[\![\mathcal{B}]\!]^c$. But for every $n \in \mathbb{N}$ we have $b^n \nsim_{[\![\mathcal{B}]\!]^c} b^{n+1}$.

This shows that $\mathbb{RB}$, $\mathbb{RP}$, $\mathbb{RM}$ and $\mathbb{RT}$ are not closed under complementation. Consider the DCA $\mathcal{B}'$ obtained from $\mathcal{B}$ by marking the states $\{1, 2, 0\}$ where 0 is the sink state. Then $\mathcal{B}'$ recognizes the same language as $\mathcal{B}$. We get that $\mathcal{B}'$ is in $\mathbb{IC}$, is respective of its right congruence, yet its complement is not respective of its right congruence. □

7.2. Union and intersection

For classes that are closed under complementation, non-closure under one of union or intersection implies non-closure under the other operation. Classes not closed under complementation may require separate proofs of non-closure for the

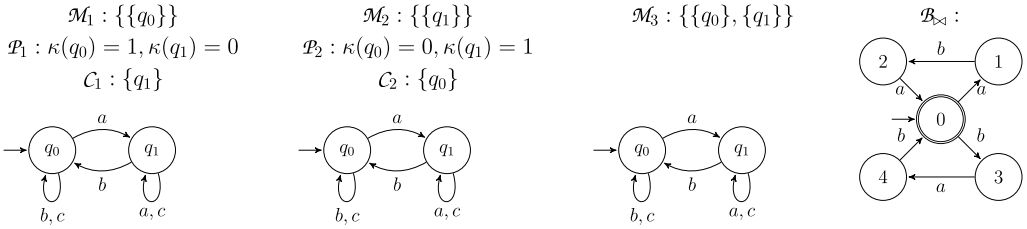

Fig. 5. On the left, examples for non-closure of union for $\mathbb{IM}$, $\mathbb{IP}$, and $\mathbb{IC}$. On the right $\mathcal{B}_{\bowtie}$. Note that the first and second automata define three acceptors each, a DMA by considering the acceptance condition specified by $\mathcal{M}_i$, a DPA by considering the acceptance condition in $\mathcal{P}_i$ and a DCA by considering the acceptance condition specified by $\mathcal{C}_i$ (for $i \in \{1, 2\}$).

two operations. Because complementation of languages in $\mathbb{IB}$ yields languages in $\mathbb{IC}$ and vice versa, non-closure of $\mathbb{IB}$ (resp. $\mathbb{IC}$) under one of union or intersection implies non-closure of $\mathbb{IC}$ (resp. $\mathbb{IB}$) under the other operation.

Proposition 33. *$\mathbb{IM}$ and $\mathbb{IP}$ are not closed under union or intersection. $\mathbb{IC}$ is not closed under union. $\mathbb{IB}$ is not closed under intersection.*

Proof. It was established in [31] that $\mathbb{IM}$ is not closed under intersection, and non-closure under union follows from closure under complementation (Proposition 30). We give a simple example of non-closure under union that also implies the non-closure under union of $\mathbb{IC}$ and $\mathbb{IP}$.

Consider the DMAs $\mathcal{M}_1$ and $\mathcal{M}_2$ from Fig. 5 (p. 14), and the DMA $\mathcal{M}_3$ recognizing the union of $[\![\mathcal{M}_1]\!]$ and $[\![\mathcal{M}_2]\!]$. The word c^ω distinguishes state q_0 from state q_1 in both $\mathcal{M}_1$ and $\mathcal{M}_2$, so both $[\![\mathcal{M}_1]\!]$ and $[\![\mathcal{M}_2]\!]$ are in $\mathbb{IM}$. However the language recognized by $\mathcal{M}_3$ is

$$(a + b + c)^*((b + c)^\omega + (a + c)^\omega),$$

which implies that its right congruence is trivial (Proposition 9). Therefore $[\![\mathcal{M}_3]\!]$ is not in $\mathbb{IM}$.

For $i = 1, 2$, the DPA $\mathcal{P}_i$ and the DCA $\mathcal{C}_i$ recognize the same language as $\mathcal{M}_i$. Thus, $[\![\mathcal{P}_1]\!]$ and $[\![\mathcal{P}_2]\!]$ are in $\mathbb{IP}$ and their union is not in $\mathbb{IP}$, and $[\![\mathcal{C}_1]\!]$ and $[\![\mathcal{C}_2]\!]$ are in $\mathbb{IC}$ and their union is not in $\mathbb{IC}$.

Non-closure of $\mathbb{IP}$ under intersection follows because $\mathbb{IP}$ is closed under complementation (Proposition 30). Non-closure of $\mathbb{IB}$ under intersection follows from non-closure of $\mathbb{IC}$ under union. □

Proposition 34. *$\mathbb{IB}$ is not closed under union and $\mathbb{IC}$ is not closed under intersection.*

Proof. We show that $\mathbb{IB}$ is not closed under union. From the duality of $\mathbb{IC}$ and $\mathbb{IB}$ it follows that $\mathbb{IC}$ is not closed under intersection.

Consider the languages L_1 and L_2 accepted by the DBA $\mathcal{B}_1$ and $\mathcal{B}_2$ in Fig. 6 (p. 15), respectively. We claim that $L_1, L_2 \in \mathbb{IB}$. To see that, note that $(ba)^\omega$ and $(ab)^\omega$ distinguish all states of $\mathcal{B}_1$, and b^ω, ab^ω distinguish all states of $\mathcal{B}_2$.

We claim that $L_1 \cup L_2$ consists of the language of all ω-words with infinitely many b's, and therefore has one right congruence class and is not in $\mathbb{IB}$. To see this, note first that $\mathcal{B}_1$ and $\mathcal{B}_2$ both reject words with a suffix a^ω (a run on such words eventually loops in state 2 in both machines). If a word w has infinitely many b's, then either (1) it ends in b^ω, and is accepted by $\mathcal{B}_1$, or (2) it has infinitely many a's and infinitely many b's, and is accepted by $\mathcal{B}_2$. □

Proposition 35. *The classes $\mathbb{RM}, \mathbb{RP}, \mathbb{RC}$ are not closed under union.*

Proof. Again, let us consider the DMAs $\mathcal{M}_1$ and $\mathcal{M}_2$ in Fig. 5 (p. 14). They are in $\mathbb{RM}$, $\mathbb{RP}$ and $\mathbb{RC}$. As shown in the proof of Proposition 33 their union (given by $\mathcal{M}_3$ in Fig. 5, p. 14) is not in $\mathbb{IM}$ or $\mathbb{IP}$ or $\mathbb{IC}$ and thus not in $\mathbb{RM}$ or $\mathbb{RP}$ or $\mathbb{RC}$. □

Proposition 36. *The classes $\mathbb{RM}, \mathbb{RP}, \mathbb{RB}$ are not closed under intersection.*

Proof. Consider DMAs $\mathcal{M}_1'$ and $\mathcal{M}_2'$ obtained from $\mathcal{M}_1$ and $\mathcal{M}_2$ of Fig. 5 (p. 14) by changing the acceptance conditions to $\{\{q_0\}, \{q_0, q_1\}\}$ and $\{\{q_1\}, \{q_0, q_1\}\}$, respectively. Both $\mathcal{M}_1'$ and $\mathcal{M}_2'$ are in $\mathbb{IM}$ since c^ω distinguishes the two states. In addition they are respective of their right congruence and are thus in $\mathbb{RM}$. Their intersection is the DMA $\mathcal{M}_3'$ obtained from $\mathcal{M}_3$ by changing the acceptance condition to $\{\{q_0, q_1\}\}$. The language recognized by $\mathcal{M}_3'$ is

$$(a + b + c)^*((b + c)^* a (a + c)^* b)^\omega$$

which has a trivial right congruence (Proposition 9), implying $\mathcal{M}_3'$ is not in $\mathbb{RM}$.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

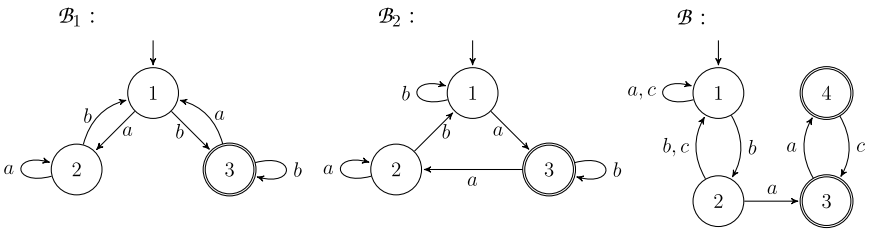

Fig. 6. On the left examples for non-closure of union for $\mathbb{IB}$. On the right, an example of a language in $\mathbb{RB}$ whose complement is not respective of its right congruence.

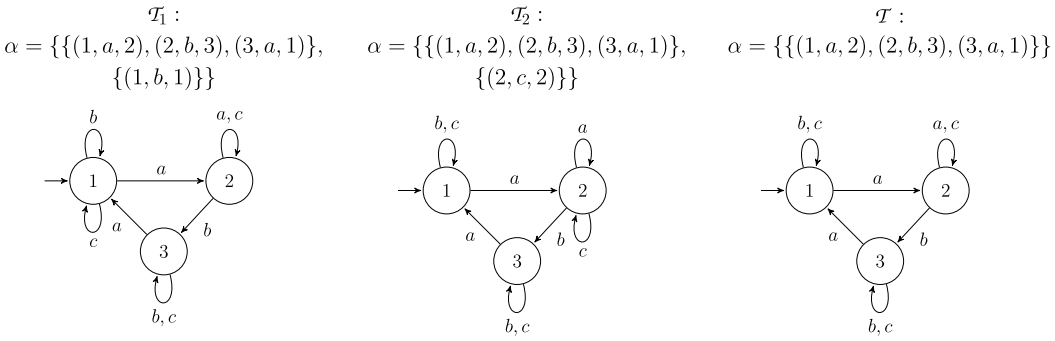

Fig. 7. Example for non closure of intersection for $\mathbb{IT}$ and $\mathbb{RT}$.

The result for $\mathbb{RB}$ follows because $\mathcal{M}_1'$ and $\mathcal{M}_2'$ can also be considered as DBAs with accepting sets $\{q_0\}$ and $\{q_1\}$, respectively. The result for $\mathbb{RP}$ follows because $\mathcal{M}_1'$ and $\mathcal{M}_2'$ can also be considered as DPAs with coloring functions $\kappa_1(q_0,q_1) = (1,2)$ and $\kappa_2(q_0,q_1) = (2,1)$, respectively. □

Proposition 37. *The classes $\mathbb{IT}$ and $\mathbb{RT}$ are not closed under intersection.*

Proof. Consider the DTAs $\mathcal{T}_1$ and $\mathcal{T}_2$ in Fig. 7 (p. 15). We claim that $[\![\mathcal{T}_1]\!]$ and $[\![\mathcal{T}_2]\!]$ are in $\mathbb{RT}$. The words b^ω and $(aba)b^\omega$ distinguish all states of $\mathcal{T}_1$ showing it is in $\mathbb{IT}$ and the words c^ω, $(aba)c^\omega$ distinguish all states of $\mathcal{T}_2$ showing it is in $\mathbb{IT}$. To see that $\mathcal{T}_1$ is respective of its right congruence note that $x(y)^\omega \in [\![\mathcal{T}_1]\!]$ if either y is a cyclic permutation of aba or a repetition thereof, or if y is in b^+ and $x \sim_{[\![\mathcal{T}_1]\!]} \epsilon$. In both cases for any n we have $x(y)^n \sim_{[\![\mathcal{T}_1]\!]} x(y)^{n+1}$. The argument for $\mathcal{T}_2$ being respective is similar. Clearly, $\mathcal{T}$ of the same figure recognizes the intersection of $\mathcal{T}_1$ and $\mathcal{T}_2$. However $\sim_{[\![\mathcal{T}]\!]}$ has only one equivalence class, showing $\mathcal{T} \notin \mathbb{IT} \supset \mathbb{RT}$. □

Proposition 38. *The classes $\mathbb{IT}$ and $\mathbb{RT}$ are not closed under union.*

Proof. Non-closure of $\mathbb{IT}$ under union follows from its non-closure under intersection (Proposition 37) and its closure under complementation (Proposition 30).

To see that $\mathbb{RT}$ is not closed under union consider the languages $T_1 = \Sigma^*(ba)^\omega + \Sigma^*ac^\omega + c^\omega$ and $T_2 = \Sigma^*(ba)^\omega + \Sigma^*bc^\omega + c^\omega$. It is easy to see that both T_1 and T_2 are in $\mathbb{IT}$ and are respective of their right congruence. However, their union $T = \Sigma^*((ab)^\omega + c^\omega))$ is not in $\mathbb{RT}$ since $\sim_T$ has a trivial right congruence (Proposition 9) but a DTA recognizing it requires at least 2 states (Proposition 11). □

Proposition 39. *The class $\mathbb{RB}$ is not closed under union and the class $\mathbb{RC}$ is not closed under intersection.*

Proof. Consider the DBAs $\mathcal{B}_1$ and $\mathcal{B}_2$ in Fig. 6 (p. 15). As shown in the proof of Proposition 34 they are both in $\mathbb{IB}$ but their union is not in $\mathbb{IB}$. It is easy to see that they are respective of their right congruence and thus in $\mathbb{RB}$, but since their union is not in $\mathbb{IB}$ it is not in $\mathbb{RB}$ either. □

8. A small experiment

We were curious to investigate the odds that a randomly generated Muller automaton will be isomorphic to its rightcon automaton, i.e., have a fully informative right congruence for DMAs. We ran a small experiment in which we generated a

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

Table 2
Results for 100 randomly generated DMAs with different numbers of states.

Number of states	5	6	7	8	9	10
Isomorphic	85	93	88	96	96	94
Not Isomorphic	15	7	12	4	4	6

random Muller automaton over an alphabet of cardinality 3, with 2 accepting strongly connected sets, and tested whether it turned out to be isomorphic to its rightcon automaton. The procedure was to try to distinguish states of the random DMA using 100,000 random ultimately periodic ω-words. If all states were successfully distinguished then the DMA is certainly isomorphic to its rightcon automaton, and was declared as such. If we failed to distinguish at least 2 states, we declared the DMA as non-isomorphic, though it might be that more tests would distinguish the undistinguished states and the DMA may in fact be isomorphic to its rightcon automaton. So the probability of a randomly generated DMA being isomorphic to its rightcon automaton may be higher than what is suggested by our results.

We generated DMAs with 5, 6, 7, 8, 9, and 10 states; 100 of each size. The results are summarized in Table 2. We find it interesting that in most of the cases a randomly generated DMA turns out to be isomorphic to its rightcon automaton, suggesting that this property is not rare. We defer a more careful study of the extent to which random automata have fully informative right congruences for further research.

9. Discussion

Considering regular ω-languages, we explored properties of their right congruences, characterized when a language has a trivial right congruence, defined classes of languages that have a fully informative right congruence, and defined an orthogonal property of a language being respective of its right congruence, which is implied by but does not imply the property of being non-counting. We showed that there are languages with fully informative right congruences in every class of the infinite Wagner hierarchy, and that this remains true if we consider languages that are also respective of their right congruences. The (mostly) non-closure results under Boolean operations are not necessarily inimical to learnability. Our hope is that future research will be able to take advantage of these properties in the search for efficient minimization and learning algorithms for regular ω-languages.

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.

Acknowledgments

This is a revised and expanded version of a paper of the same title presented at GandALF 2018. In the earlier paper, the right side of Fig. 1 (p. 2) did not faithfully represent the results; it has been corrected. We thank the reviewers for their thoughtful comments. This research was supported by grant #8758451 from the United States - Israel Binational Science Foundation (BSF), Jerusalem, Israel.

References

[1] F. Aarts, F. Vaandrager, Learning I/O automata, in: CONCUR 2010 - Concurrency Theory, 21st Inter. Conf., CONCUR 2010, Paris, France, August 31-September 3, 2010. Proc., 2010, pp. 71–85.
[2] D. Angluin, Learning regular sets from queries and counterexamples, Inf. Comput. 75 (2) (1987) 87–106.
[3] D. Angluin, U. Boker, D. Fisman, Families of DFAs as acceptors of omega-regular languages, in: 41st Inter. Symp. on Mathematical Foundations of Computer Science, MFCS, 2016, pp. 11:1–11:14.
[4] D. Angluin, U. Boker, D. Fisman, Families of DFAs as acceptors of ω-regular languages, Log. Methods Comput. Sci. 14 (1) (2018) 1–21.
[5] D. Angluin, D. Fisman, Learning Regular Omega Languages, in: Algorithmic Learning Theory - 25th Inter. Conf., ALT Proc., 2014, pp. 125–139.
[6] D. Angluin, D. Fisman, Learning regular omega languages, Theor. Comput. Sci. 650 (2016) 57–72.
[7] B. Balle, M. Mohri, Learning weighted automata, in: Algebraic Informatics - 6th International Conference, CAI 2015, 2015, pp. 1–21.
[8] F. Bergadano, S. Varricchio, Learning behaviors of automata from multiplicity and equivalence queries, SIAM J. Comput. 25 (6) (1996) 1268–1280.
[9] A.W. Biermann, J.A. Feldman, On the synthesis of finite-state machines from samples of their behavior, IEEE Trans. Comput. (ISSN 0018-9340) 21 (6) (1972) 592–597.
[10] M. Bojańczyk, Transducers with origin information, in: Automata, Languages, and Programming - 41st Inter. Colloq., ICALP, 2014, pp. 26–37.
[11] M. Botincan, D. Babic, Sigma*: symbolic learning of input-output specifications, in: 40th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Language (POPL13), 2013, pp. 443–456.
[12] H. Calbrix, M. Nivat, A. Podelski, Ultimately periodic words of rational w-languages, in: Proc. of the 9th Inter. Conf. on Mathematical Foundations of Programming Semantics, Springer-Verlag, ISBN 3-540-58027-1, 1994, pp. 554–566.
[13] V. Diekert, P. Gastin, First-order definable languages, in: Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas], 2008, pp. 261–306.
[14] D. Drews, L. D'Antoni, Learning symbolic automata, in: Tools and Algorithms for the Construction and Analysis of Systems - 23rd Inter. Conf., TACAS, 2017, pp. 173–189.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598

[15] A. Farzan, Y.C.E. Clarke, Y. Tsay, B. Wang, Extending automated compositional verification to the full class of omega-regular languages, in: Tools and Algorithms for the Construction and Analysis of Systems, in: LNCS, vol. 4963, Springer, Berlin, Heidelberg, 2008, pp. 2–17.
[16] D. Fisman, Inferring regular languages and ω-languages, J. Log. Algebraic Methods Program. 98C (2018) 27–49.
[17] M. Fliess, Matrices de Hankel, J. Math. Pures Appl. (1974) 197–224.
[18] D.M. Gabbay, A. Pnueli, S. Shelah, J. Stavi, On the temporal basis of fairness, in: Conf. Record of the 7th Annual ACM Symp. on Principles of Programming Languages, 1980, pp. 163–173.
[19] M. Leucker, Learning meets verification, in: Formal Methods for Components and Objects, 5th Inter. Symp., FMCO, 2006, pp. 127–151.
[20] C. Löding, W. Thomas, Alternating automata and logics over infinite words, in: IFIP TCS, in: Lecture Notes in Computer Science, vol. 1872, Springer, 2000, pp. 521–535.
[21] O. Maler, I. Mens, A generic algorithm for learning symbolic automata from membership queries, in: Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday, 2017, pp. 146–169.
[22] O. Maler, A. Pnueli, On the Learnability of infinitary regular sets, Inf. Comput. 118 (2) (1995) 316–326.
[23] O. Maler, L. Staiger, On syntactic congruences for omega-languages, Theor. Comput. Sci. 183 (1) (1997) 93–112.
[24] Z. Manna, A. Pnueli, A hierarchy of temporal properties, in: Proc. of the Ninth Annual ACM Symp. on Principles of Distributed Computing, 1990, pp. 377–410.
[25] I.-E. Mens, O. Maler, Learning regular languages over large ordered alphabets, Log. Methods Comput. Sci. 11 (3) (2015) 1–22.
[26] A. Pnueli, The temporal logic of programs, in: FOCS, 1977, pp. 46–57.
[27] A. Rabinovich, A proof of Kamp's theorem, Log. Methods Comput. Sci. 10 (1) (2014) 1–16.
[28] B.L. Saëc, Saturating right congruences, Inform. Théor. Appl. 24 (1990) 545–560.
[29] S. Schewe, Beyond hyper-minimisation—minimising DBAs and DPAs is NP-complete, in: IARCS Annual Conf. on Foundations of Software Technology and Theor. Comp. Science, FSTTCS, 2010, pp. 400–411.
[30] L. Staiger, Finite-state omega-languages, J. Comput. Syst. Sci. 27 (3) (1983) 434–448.
[31] D.L. Van, B.L. Saëc, I. Litovsky, Characterizations of rational omega-languages by means of right congruences, Theor. Comput. Sci. 143 (1) (1995) 1–21.
[32] K.W. Wagner, A hierarchy of regular sequence sets, in: MFCS, 1975, pp. 445–449.
[33] P. Wolper, Temporal logic can be more expressive, Inf. Control 56 (1/2) (1983) 72–99.

Please cite this article in press as: D. Angluin, D. Fisman, Regular ω-languages with an informative right congruence, Inf. Comput. (2020), https://doi.org/10.1016/j.ic.2020.104598