On the Learnability of Infinitary Regular Sets[*]

Oded Maler[†]
SPECTRE – VERIMAG[‡]
Miniparc-ZIRST
38330 Montbonnot
France
Oded.Maler@imag.fr

Amir Pnueli[§]
Department of Applied Mathematics
The Weizmann Institute
Rehovot 76100
Israel
amir@wisdom.weizmann.ac.il

October 11, 2006

[*] A preliminary version of this paper has been presented in the fourth annual workshop on *Computational Learning Theory*, Santa Cruz, 1991. To appear in *Information and Computation*

[†] The results presented in this paper has been obtained while the first author was at the Weizmann Institute working on Maler (1991).

[‡] VERIMAG is a joint laboratory of CNRS, INPG, UJF and VERILOG SA. SPECTRE is a project of INRIA.

[§] This research was supported in part by the European Community ESPRIT Basic Research Action project 3096 (SPEC).

Running head: *Learnability of Infinitary Regular Sets*

Addresss of contact author:

Oded Maler
SPECTRE – VERIMAG
Miniparc-ZIRST
38330 Montbonnot
France
`Oded.Maler@imag.fr`

Abstract

In this paper we extend the automaton synthesis paradigm to infinitary languages, that is, to subsets of the set Σ^ω of all infinite sequences over some alphabet Σ. Our main result is a polynomial algorithm for learning a sub-class of the ω-regular sets from membership queries and counter-examples based on the framework suggested by Angluin (Angluin, D., 1987, *Information and Computation* 75) for learning regular subsets of Σ^*.

1 Introduction

1.1 Motivation

The problem of sequence generalization is one of the fundamental problems in learning theory. Its automata-theoretic instance consists of the synthesis of a finite-state automaton whose behavior is compatible with a given sample of sequences over some alphabet (e.g., [Moore (1956)], [Trakhtenbrot and Barzdin (1973)], [Gold (1972)], [Biermann and Feldman (1972)], [Angluin (1987)], [Pitt (1989)]). Within this framework some aspects of real problems of learning from experience can be modeled quite naturally. One may think of an agent (a baby, a robot) thrown into some environment (a niche of the "real" world, an undocumented artifact) who tries to find some order in apparently meaningless sequences of actions and sensations (see [Rivest and Schapire (1987)] for an example of recent work under this paradigm).

The motivation behind this paper is humbler, coming from the area of system specification. Suppose a future customer of a computerized system is asked to specify the system requirements. According to these requirements some of the potential behaviors (sequences of events) of the system are considered as "good" and others are considered as "bad". Hence the requirements can be seen as nothing but a formal language over some alphabet. In many situations the customer of the system is not capable to express himself formally (e.g., give a logical formula whose set of models is exactly the set of good sequences), but rather can give examples of desired and undesired sequences. In such a situation, as suggested by [Maler (1987)] a learning algorithm can help the system designers to extract the formal specifications from the examples.

Infinitary languages can serve as models for the behavior of systems belonging to the class of *reactive* non-terminating programs in the sense of [Harel and Pnueli (1985)]. Such systems are supposed to maintain an ongoing interaction with their environments and many of their properties are best expressible when we consider infinite behavioral sequences. Thus our work is directed toward the generalization of samples consisting of good and bad infinite sequences into ω-reguler sets.[1] The main result of the paper is

[1] An alternative approach might be to learn the finitary set of prefixes as suggested in [Ulehla (1988)].

an algorithm for learning any ω-regular language U such that both U and $\Sigma^\omega - U$ are recognizable by a deterministic Büchi automaton.

1.2 Theoretical Significance

Extending the automaton synthesis framework to infinitary languages is not straightforward for the the following reasons:

1. The domain of the target function is not countable and most of its elements do not admit a finite representation. Such sample points cannot be used by any terminating learning algorithm. Thus we must learn these functions from samples consisting of "rational" points having a finite representation.[2] In the case of ω-regular sets we show that ultimately-periodic infinite sequences are sufficient for exact identification.

2. While in finitary languages, the Myhill-Nerode theorem tells us that every regular set has a unique minimal automaton isomorphic to its syntactic right-congruence, the situation in ω-automata is much more involved. Two states in an automaton can be indistinguishable by any "experiment" and still be unmergeable due to the role they play in the infinite behavior of the automaton. As a consequence we can see that the notion of a representative sample introduced by [Angluin (1981)] should be refined in the infinitary case to distinguish between examples of the finite and the infinite behavior of the automaton.

In section 2 we introduce the essentials of the theory of ω-regular sets while in section 3 we give a short survey of learnability results for the finitary case. In section 4 we discuss the technical problems associated with learning infinitary sets. The partial solution of these problems, the algorithm L^ω, is presented in section 5, followed by an example run in section 6.

[2] This observation might be relevant to other investigations of probabilistic learnability in uncountable domains, e.g., [Blumer et al. (1989)], because some pathological distributions must be excluded before distribution-independent results are discussed. Moreover, the finite precision of the sample points should be reflected in the quality of the approximating function.

2 Finite Automata and ω-Regular Sets

The theory of automata over infinite words (see [Trakhtenbrot and Barzdin (1973)], [Eilenberg (1974)], [Choueka (1974)], [Perrin (1985)], [Hoogeboom and Rozenberg (1985)], [Staiger (1987)], [Thomas (1990)] for expositions in various styles) deals with one of the simplest classes of infinitary languages, namely those that can be characterized by the infinite behavior they induce on a finite-state device. This theory is a useful tool for describing the behavior of non-terminating programs (called *reactive systems* by [Harel and Pnueli (1985)]) and for solving decision problems for various modal logics, see [Vardi and Wolper (1986)].

Let Σ denote a finite alphabet. An infinite sequence (or an ω-sequence) $\alpha \in \Sigma^\omega$ can be considered as a function $\alpha : I\!\!N \rightarrow \Sigma$. Some similarities and differences between Σ^* and the uncountable set Σ^ω of all infinite sequences are worth mentioning. The sequential concatenation operation is not defined on $\Sigma^\omega \times \Sigma^\omega$ but rather on $\Sigma^* \times \Sigma^\omega$. Every $\alpha \in \Sigma^\omega$ has infinitely many factorizations of the form $\alpha = u\beta$ into a finite prefix u and an infinite suffix β. Given a finite sequence u, u^ω is the infinite sequence obtained by concatenating infinitely many instances of u. An infinite sequence α that admits a factorization of the form uv^ω is said to be *ultimately-periodic*, where u is called the prefix and v is called the period of the factorization. Note also that being a suffix is not an order relation over Σ^ω: for example, $(ab)^\omega$ and $(ba)^\omega$ are suffixes of each other. An *infinitary set* is a subset of Σ^ω. There are several ways to relate infinitary sets to finitary ones:

Definition 1 *For every $U, V \subseteq \Sigma^*$ and $W \subseteq \Sigma^\omega$*

1. *$V^\omega \subseteq \Sigma^\omega$ denotes the set of all ω-sequences $\alpha = v_1 v_2 \ldots$ such that $v_i \in V$ for every i.*

2. *$UW \subseteq \Sigma^\omega$ denotes the set of all ω-sequences $\alpha = u\beta$ such that $u \in U$ and $\beta \in W$.*

3. *$\lim U \subseteq \Sigma^\omega$ denotes the set of all ω-sequences having infinitely-many prefixes in U.*

For example the infinitary set $0(0^*1)^\omega = \lim(0(0^*1)^*)$ consists of all the ω-words starting with 0 and containing infinitely many 1's. The infinitary counter-part of the regular sets are the ω-regular sets defined as:

Definition 2 (ω-regular sets) *A set $W \subseteq \Sigma^\omega$ is ω-regular if*

$$W = \bigcup_{i=1}^{k} U_i V_i^\omega$$

where U_i and V_i are finitary regular sets.

The following self-evident lemma is due to Büchi ([Buc62]):

Claim 1 *Every non-empty ω-regular set contains an ultimately-periodic sequence.*

Proof: Take $\alpha = uv^\omega$, $u \in U_i, v \in V_i$ for some $i \leq k$. ■

The underlying computational structure that reads an infinite sequence is the same as in the finitary case:

Definition 3 (Transition graphs) *A transition graph (transition structure, semi-automaton) is a system (Σ, Q, δ, q_0) where Σ is a finite alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \rightarrow Q$ is the transition function*[3] *and $q_0 \in Q$ is the initial state.*

When an infinite word $\alpha = \alpha_1, \alpha_2, \ldots \in \Sigma^\omega$ is fed into such a graph it induces an infinite run $\xi(\alpha) = \xi_0, \xi_1 \ldots \in Q^\omega$ through its states, such that $\xi_0 = q_0$ and for every $i > 0$, $\xi_i = \delta(\xi_{i-1}, \alpha_i)$. As Q is finite, some states occur *infinitely often* in $\xi(\alpha)$ and we denote the set of these states by $Inf(\alpha)$. In order to use automata as acceptors of ω-sets various kinds of *acceptance conditions* based on the induced run can be defined. The most commonly-used conditions are based on a relation (inclusion, disjointness or equality) between $Inf(\alpha)$ and some subsets of Q that stand for "good" or "bad" states. The following two conditions classify infinite sequences according to whether good states are visited infinitely many times or bad states are visited only finitely many times:

Definition 4 (B and $\overline{\mathbf{B}}$ automata) *Let (Σ, Q, δ, q_0) be a transition graph and let F be a subset of Q.*

- *A **B**-automaton $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ accepts the set*

$$L_{\mathcal{A}} = \{\alpha \in \Sigma^\omega : Inf(\alpha) \cap F \neq \emptyset\} \quad (1)$$

[3] All transition graphs considered in this paper are deterministic.

- *A $\overline{\mathbf{B}}$-automaton $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ accepts the set*

$$L_{\mathcal{A}} = \{\alpha \in \Sigma^\omega : Inf(\alpha) \subseteq F\} \tag{2}$$

The first acceptance condition (introduced by Büchi) states that $L_{\mathcal{A}}$ consists of all sequences that pass infinitely many times through one of the members of F. The second condition defines the set of sequences that after some point stay within F (or visit the bad states in $Q - F$ only finitely many times). Since, syntactically, both types of automata are equivalent to acceptors of finitary sets, we will sometimes refer to $\mathbf{B}$ and $\overline{\mathbf{B}}$-acceptance by the same automaton $\mathcal{A}$. The classes $\mathbf{B}$ and $\overline{\mathbf{B}}$ denote respectively the set of all subsets of Σ^ω that can be accepted by some automaton using $\mathbf{B}$ and $\overline{\mathbf{B}}$ conditions. Clearly, if $U \in \mathbf{B}$ then $\Sigma^\omega - U \in \overline{\mathbf{B}}$ and vice versa.

Example 1 *The automaton in figure 1 with $F = \{q_2\}$ accepts the finitary set $\Sigma^* b$. Viewed as an ω-automaton it $\mathbf{B}$-accepts $(\Sigma^* b)^\omega$ (all sequences containing infinitely many b's), and it $\overline{\mathbf{B}}$-accepts the set $\Sigma^* b^\omega$ (sequences containing only finitely many a's).*

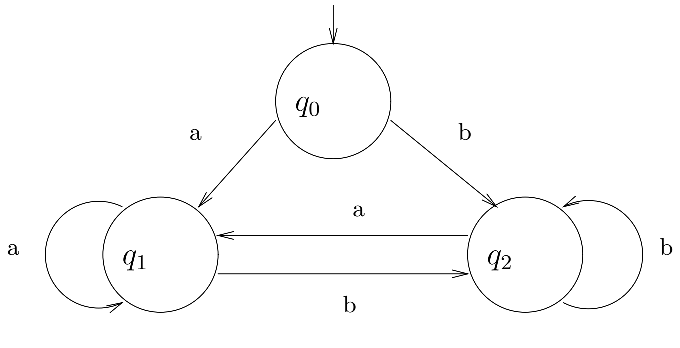

Figure 1: *An automaton.*

This is an instance of the following claim:

Claim 2 (Characterization of B and $\overline{\mathbf{B}}$) *If $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ accepts the finitary set U then it $\mathbf{B}$-accepts $\lim U$ and $\overline{\mathbf{B}}$-accepts $\Sigma^\omega - \lim(\Sigma^* - U)$.*

Not every ω-regular set is of the form $\lim U$ or $\Sigma^\omega - \lim U$. In other words, there are certain ω-regular sets that cannot be recognized by either deterministic $\mathbf{B}$-automata or deterministic $\overline{\mathbf{B}}$-automata. A more general acceptance condition due to [Muller (1963)], allows one to be more specific about $Inf(\alpha)$:

Definition 5 (M-automata) *An* $\mathbf{M}$*-automaton* $\mathcal{A} = (\Sigma, Q, \delta, q_0, \mathcal{F})$ *is a transition graph augmented with a family of subsets* $\mathcal{F} \subseteq 2^Q$. *The language accepted by* $\mathcal{A}$ *is defined as*

$$L_{\mathcal{A}} = \{\alpha \in \Sigma^\omega : Inf(\alpha) = F \text{ for some } F \in \mathcal{F}\} \tag{3}$$

The celebrated Büchi-McNaughton theorem, [McNaughton (1966)], states that $\mathbf{M}$-automata accept exactly the ω-regular sets.

Example 2 *: The automaton in figure 1 with* $\mathcal{F} = \{\{q_1, q_2\}\}$ *accepts all the sequences that pass infinitely often through both* q_1 *and* q_2*, that is, the set* $\Sigma^*(a^+b^+)^\omega = \Sigma^\omega - \Sigma^*(a^\omega + b^\omega)$.

When we consider $Inf(\alpha)$ we can refer only to states that reside in strongly-connected components (SCCs) of the transition graph – other states cannot be visited infinitely often. Let $\mathcal{S}$ be the set of all SCCs of a given automaton, partially ordered according to set inclusion. Muller condition can be viewed as partitioning $\mathcal{S}$ into good (accepting) and bad (rejecting) SCCs. An important complexity measure for ω-regular sets is the number of alternations between good and bad SCCs along the inclusion chains. For example, the automaton in figure 2, whose corresponding partially-ordered marked SCCs appear in figure 3, has three alternations along the chain $\{2\} \subseteq \{1,2\} \subseteq \{1,2,3\} \subseteq \{1,2,3,4\}$.

From this point of view $\mathbf{B}$ and $\overline{\mathbf{B}}$ automata can be considered as restricted forms of $\mathbf{M}$-automata, where in $\mathbf{B}$-automata no superset of an accepting SCC can be rejecting, and in $\overline{\mathbf{B}}$-automata no subset of an accepting SCC can be rejecting. [Wagner (1979)] has shown that this complexity measure is language-specific and is invariant over all automata accepting the same language. This means, for example, that if $U \in \mathbf{B}$ then for any $\mathbf{M}$-automaton accepting U the family $\mathcal{F}$ is upward-closed and can be expressed as the set of reachable SCCs having a non-empty intersection with some fixed $F \subseteq Q$.

An important (at least for this paper) sub-class of the ω-regular sets is the class $\mathbf{B} \cap \overline{\mathbf{B}}$ consisting of sets recognizable both by $\mathbf{B}$ and $\overline{\mathbf{B}}$-automata.

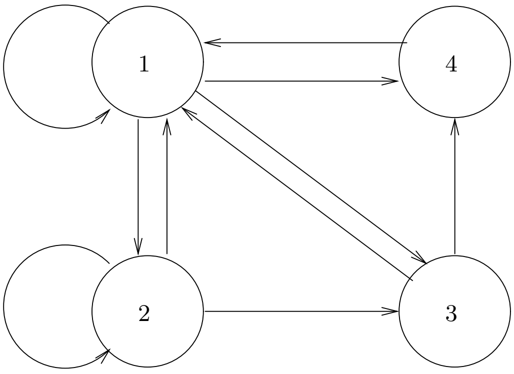

Figure 2: *An* $\mathbf{M}$*-automaton with an accepting family* $\mathcal{F} = \{\{1\},\{2\},\{1,4\},\{1,2,3\}\}$.

From the above discussion it follows that in any $\mathbf{M}$-automaton recognizing a $\mathbf{B}\cap\overline{\mathbf{B}}$ set, the family $\mathcal{F}$ is both upward-closed and downward-closed. In other words, no alternation between good and bad SCCs along any inclusion chain is allowed and all the SCCs that are contained in the same maximal SCC (MSCC) should be uniformly marked as accepting or as rejecting. Every $\mathbf{B}\cap\overline{\mathbf{B}}$-set can be accepted by a $\mathbf{B}\cap\overline{\mathbf{B}}$-automaton which is a $\mathbf{B}$-automaton whose set of accepting states F is a union of MSCCs (or equivalently a $\overline{\mathbf{B}}$-automaton with a set of rejecting states $Q-F$). The structure of a typical $\mathbf{B}\cap\overline{\mathbf{B}}$-automaton is depicted in figure 4.

For the sake of completenss we mention some connections between acceptance conditions and topological properties – they are not essential for the rest of the paper. There are various ways to make Σ^ω a topological space, either by letting the sets of the form $L\Sigma^\omega$, for some $L\subseteq\Sigma^*$, be the open sets or by defininig a metric (e.g., $d(\alpha,\beta)=\frac{1}{2^k}$ where k is the least integer such that $\alpha_k\neq\beta_k$). In this space one can see that $\mathbf{B}$ and $\overline{\mathbf{B}}$ are included in the Borel classes F_σ and G_δ respectively. The class of ω-regular sets, which is the boolean closure of $\mathbf{B}$, is contained in $F_{\sigma\delta}\cap G_{\delta\sigma}$. A fully detailed account of the topological complexity hierarchy of the ω-regular sets, its refinements and its correspondence with other complexity measures can be found in the works of [Wagner (1979)], [Kaminski (1985)], [Hoogeboom and Rozenberg (1985)], [Manna and Pnueli (1991)] and [Staiger (1987)].

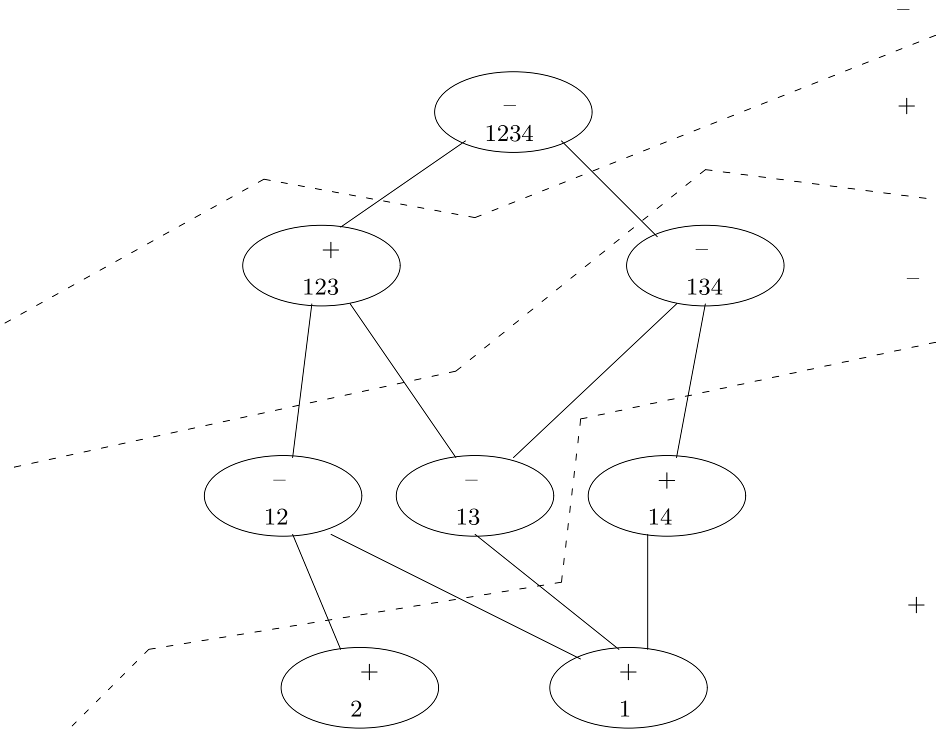

Figure 3:
The partially-ordered set of marked SCCs for the automaton in figure 2.

3 The Synthesis of Automata from Examples

The phenomenon investigated under titles such as *machine learning* and *inductive inference* is the reconstruction of some functions from a sample of their graphs; see [Angluin and Smith (1984)] for a survey. In the most general setting, we have some source of information that tells us for every sample point $w \in V \subseteq \Sigma^*$ whether or not it is a member of some set U, and the goal of learning is to build an automaton that accepts U, or, at least, an automaton compatible with the sample V).

Several variants of this problem can be charaterized by different assumptions concerning the presentation of the examples (e.g., whether they are arbitrarily given or are selected by the learning algorithm) and by different

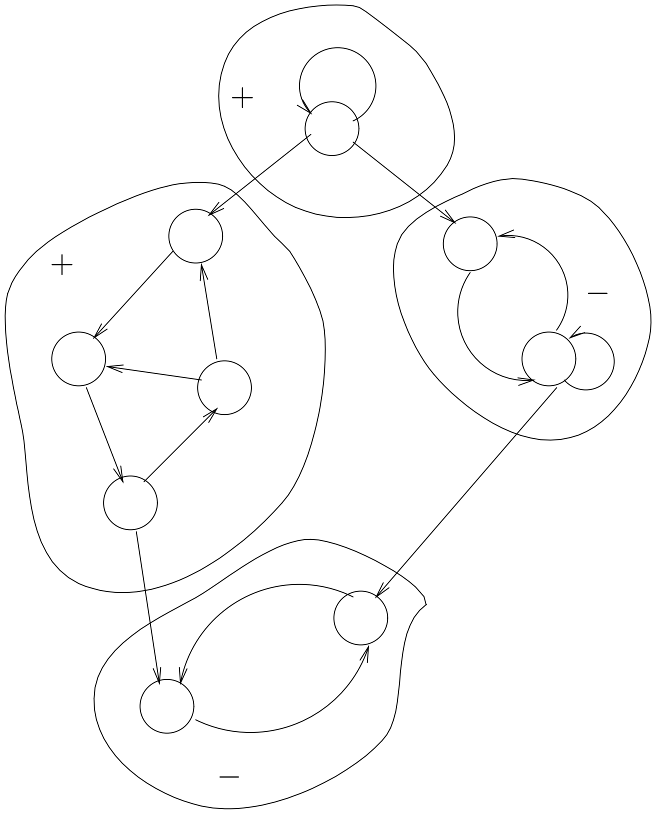

Figure 4: *A* $\mathbf{B} \cap \overline{\mathbf{B}} - \omega$*-automaton.*

criteria for the success of the learning process.

The simplest method of inductive inference is *identification by enumeration*, introduced by [Gold (1967)]. This method requires that the hypothesis space (or, more precisely, the set of their effective representations) can be effectively enumerated, and that the consistency of a hypothesis with the sample can be checked in finite time. Regular sets clearly satisfy these requirements: it is possible to enumerate all automata (or regular expressions) and it is decidable whether a word v belongs to such set. The algorithm employs an "oracle" EX that upon request produces a pair $(v, b) \in \Sigma^* \times \{0, 1\}$ such that $b = 1$ if $v \in U$ (a positive example) and $b = 0$ otherwise (a negative example). The algorithm works as follows (sets are represented by, say, deterministic automata):

Algorithm 1 (Identification by Enumeration) *Input: an oracle EX producing an infinite sequence of examples. Output: an infinite sequence of conjectured automata, each of them compatible with the examples presented so far.*

1. *Let the current hypothesis H be the automaton $\mathcal{A}_0$ (the first according to the enumeration) and let the set V of examples be $\emptyset$.*
2. *Get a new example from EX and add it to V.*
3. *While $H = \mathcal{A}_j$ contradicts V ($\mathcal{A}_j$ accepts a negative example or rejects a positive one) let H be $\mathcal{A}_{j+1}$*
4. *Goto 2.*

This algorithm has been shown by [Gold (1967)] to identify all the regular sets in the limit, i.e., for any regular set and every infinite sequence of examples, there will be some stage where the hypothesis H will stop changing. Such methods are inherently exponential (the number of automata we check is exponential in the size of the smallest accepting automaton), no matter how smart an enumeration scheme we may employ.

Practical algorithms are based on a sample-driven synthesis of automata. Most algorithms are variants of the state-characterization method of [Gold (1972)], based on the Myhill-Nerode theorem which relates input-output-based and state-based descriptions of systems:

Theorem 3 (Myhill-Nerode) *For every finitary regular set U there exists a unique minimal automaton isomorphic to the set of equivalence classes of the following right-congruence relation (the syntactic right congruence):*

$$w \approx w' \text{ iff } \forall u \in \Sigma^* : wu \in U \iff w'u \in U \tag{4}$$

The essence of this theorem for the purpose of learning is that internal states can be characterized according to their input-ouput behavior, or in other words, two states that accept exactly the same language can be merged into a single state in the minimal automaton.

There have been several results concerning the complexity of reconstructing the relation (4) from the partial information given in the sample. In particular [Gold (1978)] showed that finding the minimal automaton compatible with a sample is NP-complete. [Pitt and Warmuth(1993)] showed that this hardness result holds also for the problem of approximating the minimal automaton. On the other hand, for the case where the sample is uniform-complete (i.e., consists of all words up to a given length), [Trakhtenbrot and Barzdin (1973)] gave a polynomial algorithm for this problem. [Angluin (1987)] extended the negative results to samples that are almost (up to polynomial proportion of missing entries) uniform-complete. The positive results were extended by [Angluin (1981)] to representative samples, which include words that excercise every transition in the minimal automaton of the set. A more recent survey of complexity issues appears in [Pitt (1989)].

The algorithm L^*, due to [Angluin (1987)], is an interactive algorithm which uses two oracles: $MEMBER(u)$ which answers the question whether u is a member of the set, and an additional oracle, $EQUIV(\mathcal{A})$, which accepts as input a conjectured automaton, confirms this conjecture if it is correct, or otherwise supplies a *counter-example*, i.e., a word misclassified by this automaton. The second oracle prevents the learning algorithm from the potential need to ask exponentially many membership queries. Within this setting Angluin proved that every language accepted by an n-state automaton can be learned using polynomially many queries and computation steps. In this paper we give a similar result for $\mathbf{B} \cap \overline{\mathbf{B}}$ ω-regular sets.

4 Difficulties in Learning Infinitary Regular Sets

When we try to adapt the learning-from-examples framework to the infinitary case we face several conceptual and technical problems.

4.1 Presentation of Infinite Examples

The first problem is concerned with the availability of examples: how can one observe an infinite object? So we are restricted to those ω-words that have a finite representation. The most natural choice involves ultimately-periodic words of the form uv^ω, and, indeed, one can come out easily with our first result:

Claim 4 *The class of ω-regular sets is identifiable in the limit from ultimately-periodic examples.*

Proof: We show that the *identification by enumeration* algorithm of [Gold (1967)] is applicable. Clearly, ω-automata are effectively enumerable. The set $Inf(uv^\omega)$ can be computed for every automaton and every $u, v \in \Sigma^*$. Thus, the membership of an ultimately-periodic sequence in an ω-regular set is decidable and, consequently, the compatibility of a hypothesis with a sample can be verified. Finally, since ω-regular sets are closed under Boolean operations, the symmetric difference between the "real" set and any false ω-regular conjecture is by itself a non-empty ω-regular set and, as such, it contains an ultimately-periodic sequence. Such a sequence will eventually be presented and refute any false conjecture. ▟

Claim 5 *If two n-state $\mathbf{B} \cap \overline{\mathbf{B}}$-automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are distinguishable ($L_{\mathcal{A}_1} \neq L_{\mathcal{A}_2}$) then they are distinguishable by an ultimately-periodic ω-word α such that the size*[4] *of α is at most n^2.*

Proof: Every n-state $\mathbf{B} \cap \overline{\mathbf{B}}$-automaton $\mathcal{A}$ such that $L_{\mathcal{A}} \neq \emptyset$ accepts an ω-word uv^ω such that $|u| + |v| \leq n$ (let u be the shortest word leading to an accepting MSCC and let v be the word inducing the shortest cycle in this MSCC). The automaton accepting the symmetric difference of $L_{\mathcal{A}_1}$ and $L_{\mathcal{A}_2}$ is constructed by a cartesian product, and since it has n^2 states it accepts an ω-word of size n^2. ▟

4.2 "Unobservable" States

In order to apply the state-characterization method to infinitary languages we need some version of the Myhill-Nerode theorem. First we define the following equivalence on Σ^ω relative to an ω-regular set U:

$$\alpha \sim \alpha' \text{ iff } \alpha \in U \iff \alpha' \in U \tag{5}$$

The infinitary analogue of the syntactic right-congruence (4) is the following:

$$w \approx w' \text{ iff } \forall \alpha \in \Sigma^\omega : w\alpha \sim w'\alpha \tag{6}$$

[4] The size of an ultimately-periodic word α is defined as the minimum of $|u| + |v|$ over all $u, v \in \Sigma^*$ such that $\alpha = uv^\omega$.

It follows from claim 1 and from the fact that the set $\{\alpha : u\alpha \in U\}$ is also ω-regular, that the above formulation of $\approx$ in terms of arbitrary infinite continuations is equivalent to a formulation using ultimately-periodic words:

$$w \approx w' \text{ iff } \forall u \in \Sigma^*, v \in \Sigma^+ : wuv^\omega \sim w'uv^\omega \tag{7}$$

Unlike the finitary case not every ω-regular set U can be accepted by an automaton whose states are representable by the congruence classes of this relation. Let us look, for example, at the right-congruence associated with the set $U = (0{+}1)^*1^\omega$. Clearly, it consists of exactly one equivalence class because all the finite prefixes "accept" the same infinite suffixes, namely those that eventually consist only of 1's. The automaton associated with this relation is the trivial one-state automaton but this automaton can accept nothing but Σ^ω or the empty set. So every automaton for this set must have at least two states which are not necessarily distinguishable by $\approx$. Moreover, for some ω-regular sets *there exists no unique canonical ω-automaton* which is a homomorphic image of every automaton accepting those sets. Three 2-state mutually non-homomorphic and irreducible automata for $U = (0 + 1)^*1^\omega$ are depicted in figure 5.

The existence of such "non-observable states" implies that we should devise an algorithm for detecting the necessity of additional "hidden" states in any sample-compatible automaton and for deciding how to split some of the observed states into two or more copies. Since there is no unique minimal automaton accepting a given ω-language, the choice of this algorithm will dictate to which of the several irreducible non-isomorphic automata will the synthesis algorithm converge. There is no guarantee that this automaton will not be exponentially larger than some other automaton accepting the same set, and no other solution better than exhaustive search is currently known. The largest sub-class of the ω-regular sets which is known to "behave properly" in this sense is the class $\mathbf{B} \cap \overline{\mathbf{B}}$ according to the following result of [Staiger (1983)]:

Theorem 6 (Staiger) *Every $\mathbf{B} \cap \overline{\mathbf{B}}$ ω-regular set is accepted by an automaton isomorphic to its syntactic right-congruence.*

As a corollary, the states of the minimal automaton for $\mathbf{B} \cap \overline{\mathbf{B}}$ sets can be identified by unique entries in the observation table (see definition below). By restricting our algorithm to this non-trivial class we defer the removal

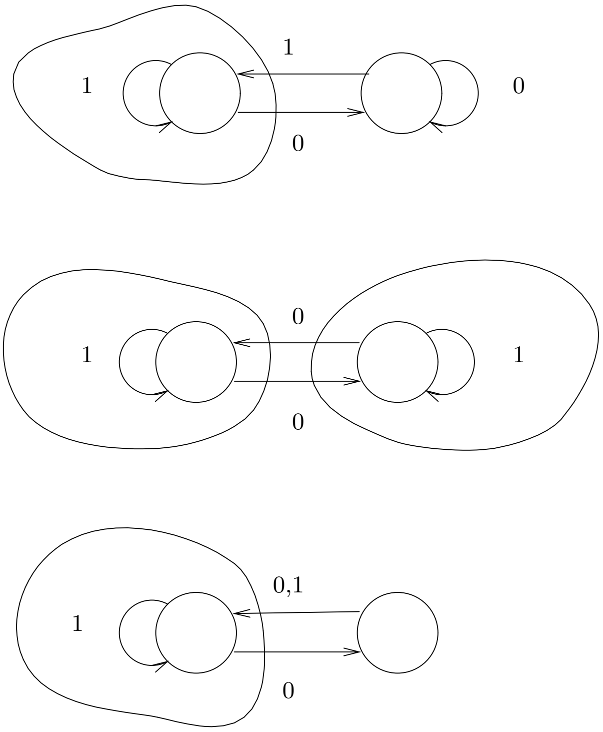

Figure 5: *Three “minimal” automata for* $U = (0+1)^*1^\omega$

of the main obstacle preventing the application of learning algorithms to ω-regular languages.

5 The Algorithm L^ω

Here we describe an algorithm for synthesizing $\mathbf{B} \cap \overline{\mathbf{B}}$-automata from queries and counter-examples. The algorithm is an extension of the L^* algorithm of [Angluin (1987)].

5.1 Definitions and Setting

We redefine Angluin’s observation tables in a manner suitable for identifying the syntactic right-congruence associated with ω-regular sets.

Definition 6 (ω-observation tables) *Let S be a finite prefix-closed subset of Σ^*, let E be a finite suffix-closed set of ultimately-periodic words over Σ, and let $S' = S \cup S\Sigma$. An ω-observation table for an ω-regular set U is a triple (S, E, T) where $T : S' \times E \longrightarrow \{0, 1\}$ is defined by: $T(s, \alpha) = 1$ iff $s\alpha \in U$. For each $s \in S'$, we denote by $f_s : E \longrightarrow \{0, 1\}$ the function given by $f_s(\alpha) = T(s, \alpha)$.*

Whenever an ultimately-periodic word α is classified as a member or a non-member of U, its classification is written in all entries (s, β) such that $\alpha = s\beta$. The notions of a closed table and its associated transition graph, which are identical to the same[5] notions for the finitary case presented by [Angluin (1987)], are defined as follows:

Definition 7 (Closed tables) *An observation table is closed if for every $s' \in S\Sigma$ there exists some $s \in S$ such that $f_s = f_{s'}$. With every closed observation table we can associate a transition graph (Σ, Q, δ, q_0) with $Q = \{f_s : s \in S\}$, $q_0 = f_\lambda$ and $\delta(f_s, \sigma) = f_t$ for some $t \in S$ satisfying $f_t = f_{s\sigma}$.*

We can extend the f_s-notation to every $u \notin S$ by identifying f_u with $\delta(f_\lambda, u) = f_t$ for some $t \in S$. This notation is justified by the easily-verified fact that for every $u \in \Sigma^*, \alpha \in \Sigma^\omega$ such that $u\alpha$ is in the sample, $f_t(\alpha) = f_\lambda(u\alpha)$.

Finally the sources of information are the oracles $MEMBER(u, v)$ and $EQUIV(\mathcal{A})$, the former returning the membership of uv^ω in U and the latter either confirming the correctness of a conjectured ω-automaton $\mathcal{A}$, or returning an ultimately-periodic counter-example otherwise. A counter-example is an ultimately periodic word which is either in $U - L_\mathcal{A}$ or in $L_\mathcal{A} - U$.

When we deal with automata over finite words, after we discover the underlying transition structure of the automaton it is straightforward to identify the accepting states. This is done according to their behavior on λ, i.e., $f_s \in F$ iff $f_s(\lambda) = 1$. In the infinitary case this is not as simple. A cycle in the transition graph need not be exercised *infinitely often* by a word in the sample, although each of its constituting transitions have been exercised at least once. Thus, the notion of a *representative sample* as defined by

[5] In fact, Angluin uses an additional notion of a *consistent* table, but this can be eliminated by a slight modification of L^*: instead of adding the prefixes of a counter-example to S, add their suffixes to E. This way the table is always consistent.

[Angluin (1981)] should be refined to cover infinite behavior. Moreover, if we restrict ourselves to certain sub-classes of automata, it might be impossible to augment the transition graph with appropriate acceptance conditions that yield a compatible automaton. We will give an example of this phenomenon in the sequel.

Our algorithm solves this problem for $\mathbf{B} \cap \overline{\mathbf{B}}$-automata in two stages. During the first stage it determines the transition graph in a manner similar to the finitary case. The second stage consists of an attempt to identify the accepting states. If the process succeeds we can construct a minimal compatible $\mathbf{B} \cap \overline{\mathbf{B}}$-automaton as our current conjecture. Otherwise the algorithm tells us how to extend E in order to discover hidden states, whose addition leads to a consistent identification of the accepting states.

5.2 From Transition Graphs to Automata

After determining the transition graph, we should mark the various states as accepting or as rejecting. In $\mathbf{B} \cap \overline{\mathbf{B}}$-automata no "good" and "bad" SCCs intersect. Thus, if the current graph can support such an automaton the states must be non-ambiguously identifiable as good or bad according to the cycles they participate in. In case this cannot be done we discover a *conflict* which is defined as:

Definition 8 (Conflict) *For an ω-observation table, a conflict is a triple (s, u, v) of words, such that $f_{su} = f_{sv} = f_s$, but $f_s(u^\omega) \neq f_s(v^\omega)$.* (See figure 6).

The following algorithm attempts to mark each state as accepting or rejecting and detects a conflict if this is impossible:

Procedure 2 (Mark or Detect) *Input: a complete ω-observation table. Output: either a consistent marking of the states as accepting or rejecting or a conflict.*

1. *Mark all the states in S as "indefinite".*

2. *For every ω-word α in the sample, i.e., $\alpha = s\beta$ for some $s \in S$ and $\beta \in E$, compute $Inf(\alpha)$ and mark every $q \in Inf(\alpha)$ according to whether α is a positive or a negative example.*

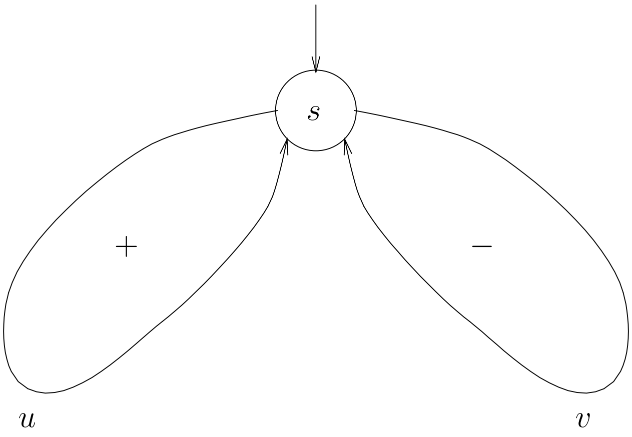

Figure 6: *A conflict* (s, u, v).

3. *Compute the set* $\mathcal{S}$ *of MSCCs of the graph. If no two states that belong to the same MSCC are marked by conflicting signs, mark the remaining unmarked states according to their marked neighbors in the MSCC and stop.*[6]

4. *Otherwise take two strongly-connected states* f_s *and* f_t *such that* sx^ω *is a positive example and* ty^ω *is a negative example for some* $x, y \in \Sigma^+$. *Find two words* z, w *such that* $f_{sz} = f_t$ *and* $f_{tw} = f_s$ *(such words must exist because* f_s *and* f_t *are strongly-connected) and ask for the membership of* $s(zw)^\omega$ *in* U. *If the answer is negative there exists a conflict* (s, x, zw)*, otherwise a conflict* (t, y, wz) *is detected.*[7]

Conflict detection is illustrated in figure 7. Sometimes $f_s = f_t$ and the conflict is detected already in steps 2-3 of the procedure (as in figure 6 and in the example presnted in section 6).

Claim 7 *The marking procedure always terminates after polynomially many (in the size of the table) steps.*

[6] MSCCs such that none of their states is visited infinitely-often by the sample can be marked arbitrarily.

[7] Recall that $s(zw)^\omega = sz(wz)^\omega = t(wz)^\omega$.

Proof: The two main computational tasks in this procedure are the computation of $Inf(uv^\omega)$ for every ω-word in the sample, and finding MSCCs in a graph. The former can be done by looking at all states visited by the word uv^n (whose size is $|u| + n|v|$) and the latter is known to be polynomial. ▟

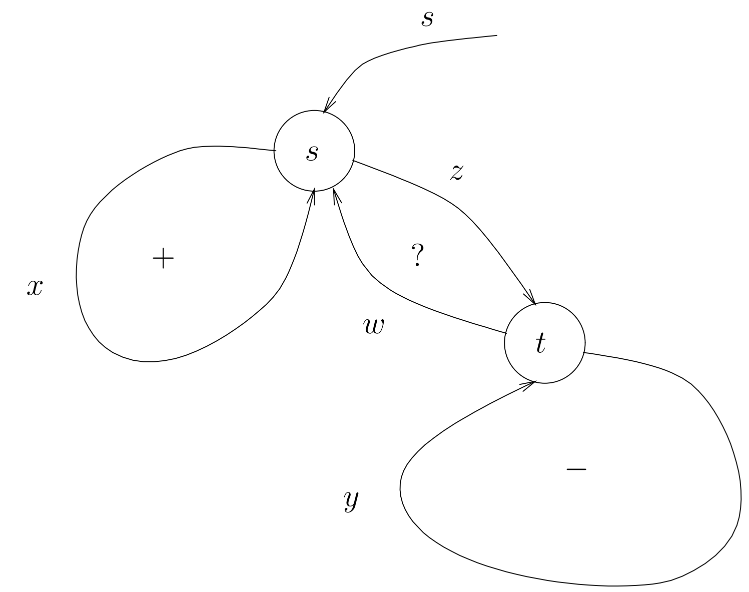

Figure 7: *Conflict detection.*

5.3 Conflict Resolution

A conflict (s, u, v) means that, according to our state characterization, both u and v induce a cycle from f_s to itself but the marks computed for these cycles contradict each other. Thus, in the table corresponding to the "real" automaton at least one of $\{f_{su}, f_{sv}\}$ must be distinct from f_s and there should be a word establishing this distinction. The procedure *conflict(*s, u, v*)* finds such a distinguishing experiment using a certain property of $\mathbf{B} \cap \overline{\mathbf{B}}$ automata that we prove.

Definition 9 *For finite words* $u, v \in \Sigma^+$ *and* $n > 0$*, let* $\Delta^n_{u,v}$ *be the set of words of the form* xu^ω *or* xv^ω*, where* x *is a prefix of* $(u^n v^n)^n$ *or a prefix of* $(v^n u^n)^n$*.*

Claim 8 *Let $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ be an n-state $\mathbf{B} \cap \overline{\mathbf{B}}$-automaton accepting U such that $su^\omega \in U$ and $sv^\omega \notin U$. Obviously, one of $\{\delta(q_0, su), \delta(q_0, sv)\}$ must be distinct from $\delta(q_0, s)$. Let $t \in \{su, sv\}$ be a finite word such that $\delta(q_0, t) \neq \delta(q_0, s)$. Then there exists a word $\alpha \in \Delta^n_{u,v}$ distinguishing between t and s, i.e., such that $t\alpha \not\sim s\alpha$.*

Proof: Suppose, without loss of generality, that $t = sv$, i.e., $\delta(q_0, sv) \neq \delta(q_0, s)$. The sequence sv^n clearly leads to the rejecting SCC visited infinitely often by sv^ω – otherwise we had more than n states (see figure 8). There are two possibilities concerning $sv^n u^\omega$:

1. $sv^n u^\omega$ is rejected, in which case we are done because there exists some k, $0 \leq k < n$ such that $sv^k u^\omega \in U$ and $sv^{k+1}u^\omega \notin U$ and thus $\alpha = v^k u^\omega$ distinguishes between s and sv.

2. $sv^n u^\omega$ is accepted, so $sv^n u^n$ leads to a new accepting SCC, and so on. As the number of SCCs is bounded by n, the number of alternations is bounded as well (see figure 9) and there is some distinguishing α, such that $\alpha = v^n u^n \ldots u^n v^k u^\omega$ or $\alpha = v^n u^n \ldots v^n u^k v^\omega$. ∎

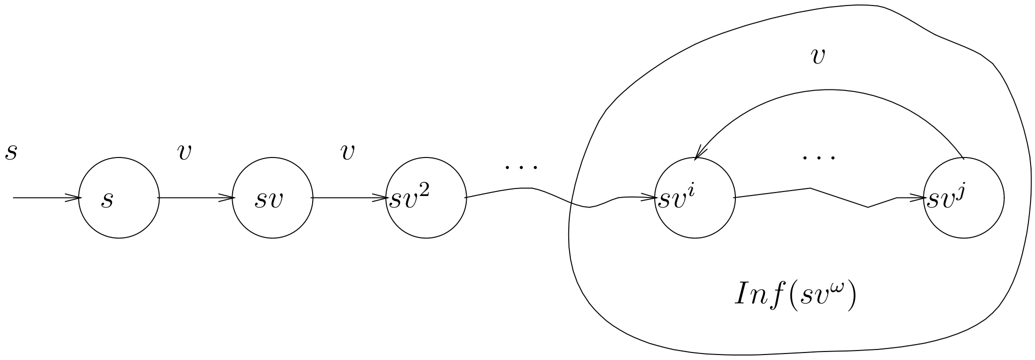

Figure 8: *sv^n leads to $Inf(sv^\omega)$.*

Procedure 3 (Conflict Resolution) *Input: a conflict (s, u, v). Output: an ω-word $x\alpha$ such that $x \in \{u, v\}$ and $s\alpha \not\sim sx\alpha$.*
For every $\alpha \in \Delta^n_{u,v}$ (enumerated in increasing-size order) ask membership queries about $s\alpha$, $sv\alpha$ and $su\alpha$ until either $s\alpha \not\sim sv\alpha$ or $s\alpha \not\sim su\alpha$.

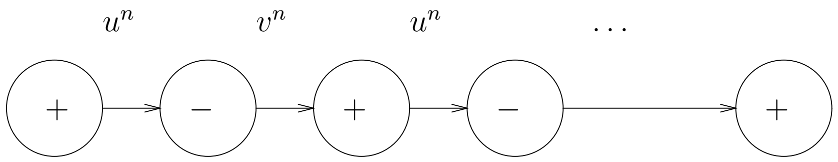

Figure 9: *Alternating SCCs.*

Claim 9 *The conflict resolution procedure always terminates after polynomially many steps and queries.*

Proof: Termination is guaranteed by claim 8. The length of the elements in $\Delta^n_{u,v}$ is $O(n^2)$ and $|\Delta^n_{u,v}| = O(n^2)$. ▟

5.4 The Algorithm

The complete algorithm is described below:

Algorithm 4 **(L^ω)** *Input: the oracles MEMBER and EQUIV for a* $\mathbf{B}\cap\overline{\mathbf{B}}$ *ω-regular set U. Output: the minimal* $\mathbf{B}\cap\overline{\mathbf{B}}$*-automaton accepting U.*

```
$S := \{\lambda\}$; $E := \{\sigma^\omega : \sigma \in \Sigma\}$; Confirmed:=False
while $\neg$Confirmed do
        while the table is not closed do
                add the new state $s\sigma$ to $S$
                and extend the table accordingly.
        end
        construct the transition graph
        call the marking algorithm (procedure 2)
        if the marking fails then
                call the conflict resolution procedure (procedure 3)
                add the outcome and its suffixes to $E$.
        else
                let the current conjecture be the marked automaton
                call the EQUIV oracle
                if the answer is positive then
                    Confirmed:=True
```

```
            else
                add the counter-example and its suffixes to $E$.
    end
```

5.5 Properties of L^ω

Theorem 10 *The algorithm L^ω always terminates after polynomially many (in the number of states, and the size of counter-examples) membership queries, equivalence queries and computation steps. It always produces the (unique) minimal $\mathbf{B} \cap \overline{\mathbf{B}}$-automaton compatible with the sample.*

Proof: The proof is along the same lines as the corresponding proof for L^* given by [Angluin (1987)]. In fact L^* can be viewed as a degenerate case of L^ω where the marking procedure is trivial and always succeeds and thus procedure 3 is never called.

One obvious invariance of the algorithm is that $|S| \leq n$ where n is the number of states in the minimal automaton for U – we add states only when they are dicovered. Thus if the program terminates the current conjecture is indeed the minimal automaton for U.

We will now show that the main loop can repeat only $O(n^2)$ times. Each time the loop repeates it is due to one of the following two reasons:

1. The marking fails – in this case a conflict has been detected, and after adding the outcome of procedure 3 to E, a new state is discovered in the next iteration of the main loop. Hence this can happen at most $n - 1$ times.

2. The marking succeeds but the $EQUIV$ oracle presents a counter-example which is added to the table. In the next iteration, there are three possibilities: 1) the table is not closed anymore and a new state is added, 2) the table is closed but the marking fails and thus a new state must be discovered in the next iteration, and 3) the table is closed and the marking succeeds. This can happen only if the counter-example β had $Inf(\beta)$ belonging to a MSCC which was not visited infinitely often by any of the previous examples. For a given transition graph of size m, the number of times this might happen is bounded by the number of MSCCs which is at most m. Since at all stages the size of the

transition graph is not greater then n we can conclude that the number of counter-examples that can be presented between two consecutive discoveries of new states is at most n.

Hence the algorithm terminates after $O(n^2)$ iterations of the main loop.

As for complexity, we will first show that the size of the table (and the total number of queries) is always kept polynomial. The number of rows is $O(n)$. The set of columns is extended at most n^2 times in one of the following two occasions: 1) when a conflict is detected and resolved, and 2) when a counter-example is presented. In both cases we add an utlimately-periodic ω-word β to E together with its suffixes[8] and both their number and size are $O(|\beta|)$. In the first case β is an element of $\Delta^n_{u,v}$, and thus $|\beta| = O(n^2)$ and in the second case if we assume that counter-examples are of minimal length, it follows from claim 5 that $|\beta| = O(n^2)$.

This fact and together with claim 7 implies that each call to the marking procedure takes polynomially many steps and the proof is concluded. ▟

Corollary 11 (Main result) *The class of* $\mathbf{B} \cap \overline{\mathbf{B}}$ *ω-regular sets is polynomially learnable from queries and counter-examples.*

6 An Example Run of L^ω

Let $U = (01)^*(10)^\omega$. The initial observation table is completely negative because no 1-period word is accepted. Thus the first conjecture is the empty language accepted by the trivial 1-state automaton, for which $+(10)^\omega$ is a counter-example. Adding $(10)^\omega$, with its "co-suffix" $(01)^\omega$ to E leads to the table and transition graph of figure 10.

All the states in this graph belong to the same maximal SCC and they cannot be marked consistently. The state f_λ accepts $(10)^\omega$ and rejects $(01)^\omega$ while $f_\lambda = f_{01} = f_{10}$. So we call *conflict*$(\lambda, 01, 10)$ which returns the experiment $01(10)^\omega$ By adding $1(10)^\omega$ and $01(10)^\omega$ to E we discover the states $f_{00} = f_{11}$ and f_{10}. The final table and its corresponding transition graph are depicted in figure 11. There are three maximal strongly-connected components in this graph and each of them can be marked consistently.

[8] Since the suffix relation over Σ^ω is not an order relation, being suffix-closed for $E \subseteq \Sigma^\omega$ means not only that membership of σuv^ω in E implies membership of uv^ω but also that membership of v^ω implies the membership of t^ω for every t which is a rotation of v.

7 Conclusion

We have shown that the state-characterization approach to automaton synthesis can be applied to a "well-behaving" sub-class of the ω-regular sets. Concerning the rest of the ω-regular sets, where a unique minimal automaton does not exist, [Arnold (1985)] introduced an alternative canonical object. Unfortunately this object is a two-sided congruence (semigroup) whose size for a given language can be considerably larger than the size of an accepting automaton. Thus, although Arnold's congruence can be learned using two 3-dimensional observation tables, this option is not very attractive. Some investigations of a "one-sided" analogue of Arnold's congruence were made by [Le Saec (1990)], and recently [Maler and Staiger (1993)] defined a much smaller canonical object, but its usefulness for efficient learning has not been investigated yet.

Acknowledgements

We would like to thank A. Arnold for many useful comments. We thank one anonymous referee for noting a small bug in a previous version of the algorithm, and a second referee for demanding an improvement in the presentation of theorem 10.

References

[Angluin (1987)] Angluin, D. (1978), On the Complexity of Minimum Inference of Regular Sets, *Inform. and Control* 39, 337-350.

[Angluin (1981)] Angluin, D. (1981), A Note on the Number of Queries Needed to Identify Regular Languages, *Inform. and Control* 51, 76-87.

[Angluin (1987)] Angluin, D. (1987) Learning Regular Sets from Queries and Counter-Examples, *Inform. and Comput.* 75, 87-106.

[Angluin and Smith (1984)] Angluin D., and Smith, C.H. (1984), Inductive Inference: Theory and Methods, *ACM Computing Surveys* 15, 237-269.

[Arnold (1985)] Arnold, A. (1985), A Syntactic Congruence for Rational ω-Languages, *Theoret. Comput. Science* 39, 333-335.

[Biermann and Feldman (1972)] Biermann, A.W., and Feldman, J.A. (1972), On the Synthesis of Finite-State Machines from Samples of their Behavior, *IEEE Trans. on Computers* C-21, 592-597.

[Blumer et al. (1989)] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M.K. (1989), Learnability and the Vapnik-Chervonenkis Dimension, *J. of the ACM* 36, 929-965.

[Büchi (1962)] Büchi, J.R. (1962), On a Decision Method in Restricted Second Order Arithmetic, *Proc. of the 1960 Int. Congr. for Logic*, 1-11, Stanford University Press, Stanford.

[Choueka (1974)] Choueka, Y. (1974), Theories of Automata on ω–Tapes: A Simplified Approach, *J. of Computer and System Sciences* 8, 117-141.

[Eilenberg (1974)] Eilenberg, S. (1974), *Automata, Languages and Machines, Vol. A*, Academic Press, New-York.

[Gold (1967)] Gold, E.M. (1967), Language Identification in the Limit, *Inform. and Control* 10, 447-474.

[Gold (1972)] Gold, E.M. (1972), System Identification via State Characterization, *Automatica* 8, 621-636.

[Gold (1978)] Gold, E.M. (1978), Complexity of Automaton Identification from Given Data, *Inform. and Control* 37, 302-320.

[Harel and Pnueli (1985)] Harel, D., and Pnueli, A. (1985), On the Development of Reactive Systems, in *Logics and Models of Concurrent Systems* (K.R. Apt, Ed.), 477-498, Springer, Berlin.

[Hoogeboom and Rozenberg (1985)] Hoogeboom, H.J. and Rozenberg G. (1985), Infinitary Languages – Basic Theory and Applications to Concurrent Systems, in *Current Trends in Concurrency* (J.W. de-Bakker et al., Eds.), Lecture Notes in Computer Science 224 , 266-342, Springer, Berlin.

[Kaminski (1985)] Kaminski, M. (1985), A Classification of ω-Regular Languages, *Theoret. Comput. Science* 36, 217-229.

[Le Saec (1990)] Le Saec, B. (1990), Saturating Right Congruences, *Theor. Inf. et Appl.* 24, 545-560.

[Maler (1987)] Maler, O. (1987), *Detecting Regularities in Computations*, Unpublished Ph.D. Proposal, Weizmann Institute, Rehovot.

[Maler (1991)] Maler, O. (1991), *Finite Automata: Infinite Behavior, Learnability and Decomposition*, Unpublished Ph.D. Dissertation, Weizmann Institute, Rehovot.

[Maler and Staiger (1993)] Maler, O., and Staiger, L. (1993), On Syntactic Congruences for ω-Languages, in *Proc. of the 10th Ann. Symp. on Theoreticl Aspects of Computer Science*, (P. Enjalbert, A. Finkel and K.W. Wagner, Eds.) Lecture Notes in Computer Science 665, Springer, Berlin.

[Manna and Pnueli (1991)] Manna, Z., and Pnueli, A. (1991), Completing the Temporal Picture, *Theoret. Comput. Science* 83, 97-130.

[McNaughton (1966)] McNaughton, R. (1966), Testing and Generating Infinite Sequences by a Finite Automaton, *Inform. and Control* 9, 521-530.

[Moore (1956)] Moore, D.E. (1956), Gedanken Experiments on Sequential Machines, in *Automata Studies* (C. Shannon and J.M. McCarthy, Eds.), 129-153, Princeton University Press, Princeton.

[Muller (1963)] Muller, D.E. (1963), Infinite Sequences and Finite Machines, *AIEE Proc. 4th Ann. Symp. on Switching Theory and Logical Design*, Chicago, 3-16.

[Perrin (1985)] Perrin, D. (1985), An Introduction to Finite Automata on Infinite words, in *Automata on Infinite Words* (D. Perrin, Ed.), 2-17, Lecture Notes in Computer Science 192, Springer, Berlin.

[Pitt (1989)] Pitt, L. (1989), Inductive Inference, DFAs, and Computational Complexity, in *Proc. of the 2nd International Workshop on Analogical and Inductive Inference* (K. Jantke, Ed.), 18-44, Lecture Notes in Computer Science 397, Springer, Berlin.

[Pitt and Warmuth(1993)] Pitt, L. and Warmuth, M.L. (1993), The Minimum Consistent DFA Problem Cannot be Approximated within any Polynomial, *Journal of the ACM* 40, 95-142.

[Rivest and Schapire (1987)] Rivest, R.L., and Schapire, R.E. (1987), Diversity-Based Inference of Finite Automata, *Proc. of the IEEE 28th Symposium on the Foundations of Computer Science*, 78-87.

[Staiger (1983)] Staiger, L. (1983), Finite-State ω-Languages, *J. of Computer and System Sciences* 27, 434-448.

[Staiger (1987)] Staiger, L. (1987), Research in the Theory of ω-languages, *J. Inf. Process. Cybern. EIK* 23, 415-439.

[Thomas (1990)] Thomas, W. (1990), Automata on Infinite Objects, in *Handbook of Theoretical Computer Science, Vol. B* (J. Van Leeuwen, Ed.), 133-191, Elsevier, Amsterdam.

[Trakhtenbrot and Barzdin (1973)] Trakhtenbrot, B.A. and Barzdin, Y.M. (1973), *Finite Automata: Behavior and Synthesis*, North-Holland, Amsterdam, 1973.

[Ulehla (1988)] Ulehla, J. (1988), *Over the Draft of Detecting Regularities in Computations*, Report 88-78, Charles University, Prague.

[Vardi and Wolper (1986)] Vardi. M.Y., and Wolper, P. (1986), Automata-Theoretic Techniques in Modal Logics of Programs, *J. of Computer and Systems Science* 32, 183-221.

[Wagner (1979)] Wagner, K. (1979), On ω-Regular Sets, *Inform. and Control* 43, 123-177.

	0^ω	1^ω	$(01)^\omega$	$(10)^\omega$
λ	$-$	$-$	$-$	$+$
0	$-$	$-$	$-$	$-$
1	$-$	$-$	$+$	$-$
00	$-$	$-$	$-$	$-$
01	$-$	$-$	$-$	$+$
10	$-$	$-$	$-$	$+$
11	$-$	$-$	$-$	$-$

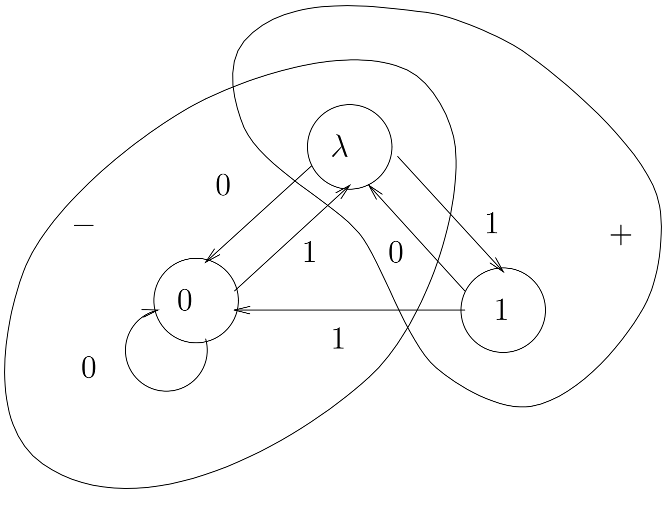

Figure 10: *A closed table and its associated transition-graph with no compatible* $\mathbf{B} \cap \overline{\mathbf{B}}$ *acceptance condition.*

	0^ω	1^ω	$(01)^\omega$	$(10)^\omega$	$1(10)^\omega$	$01(10)^\omega$
λ	$-$	$-$	$-$	$+$	$-$	$+$
0	$-$	$-$	$-$	$-$	$+$	$-$
1	$-$	$-$	$+$	$-$	$-$	$-$
00	$-$	$-$	$-$	$-$	$-$	$-$
10	$-$	$-$	$-$	$+$	$-$	$-$
01	$-$	$-$	$-$	$+$	$-$	$+$
11	$-$	$-$	$-$	$-$	$-$	$-$
000	$-$	$-$	$-$	$-$	$-$	$-$
001	$-$	$-$	$-$	$-$	$-$	$-$
100	$-$	$-$	$-$	$-$	$-$	$-$
101	$-$	$-$	$+$	$-$	$-$	$-$

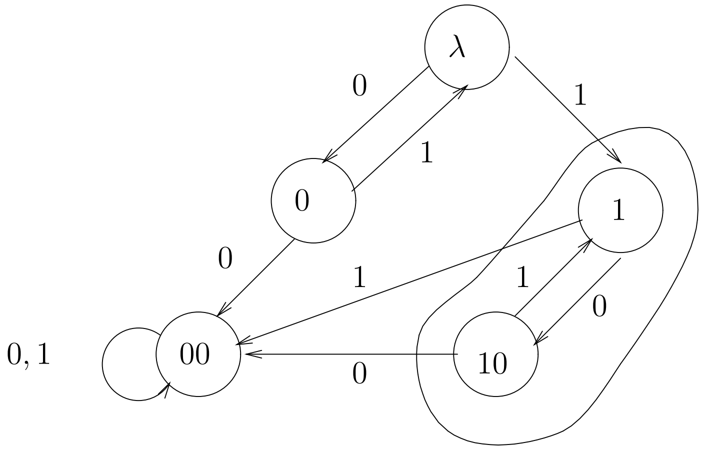

Figure 11: *The final table and its corresponding automaton.*