Learning deterministic finite-state machines from the prefixes of a single string is NP-complete

Radu Cosmin Dumitru R.C.DUMITRU@STUDENT.TUDELFT.NL
Delft University of Technology, Delft, Netherlands
Ryo Yoshinaka RYOSHINAKA@TOHOKU.AC.JP
Ayumi Shinohara AYUMIS@TOHOKU.AC.JP
Tohoku University, Sendai, Japan

Editor: Editor's name

Abstract

Computing a minimum deterministic finite automaton consistent with a given set of positive and negative examples is known to be NP-hard. Previous work has identified conditions on the input sample under which the problem becomes tractable or remains hard. In this paper, we study the computational complexity of the case where the input sample is prefix-closed. This formulation is equivalent to computing a minimum Moore machine consistent with observations along its runs. We show that the problem is NP-hard to approximate when the sample set consists of all prefixes of binary strings. Furthermore, we show that the corresponding decision problem remains NP-complete even when the sample set consists of all prefixes of a single binary string. Our argument also extends to the corresponding problem for Mealy machines.

Keywords: automata learning, learning complexity, problem approximability, prefix-closed sample

1. Introduction

The problem of learning a deterministic finite automaton (DFA) from a labeled sample set containing positive and negative examples has been a central problem in grammatical inference. In particular, the MinConDFA problem, which asks for a smallest DFA consistent with the given sample, has been extensively studied. It was shown to be NP-hard by Gold (1978) and Angluin (1978).[1] Furthermore, Pitt and Warmuth (1993) showed that MinConDFA is not only NP-hard, but also hard to approximate within any polynomial ratio. Assuming P $\neq$ NP, they proved that for any constant c, no polynomial-time algorithm can guarantee a consistent DFA of size at most m_*^c, where m_* denotes the size of a smallest consistent DFA. More recently, Chalermsook et al. (2014) proved that MinConDFA is NP-hard to approximate within a factor of $n^{1-\epsilon}$ for any constant $\epsilon > 0$, where n is the sample size; this bound is tight, since a trivial $O(n)$-approximation is achieved by the prefix-tree acceptor.

On the other hand, the literature has proposed several polynomial-time algorithms (Oncina and García, 1992; Lang et al., 1998, etc.) that compute a smallest DFA when

1. Precisely speaking, the model considered in these works corresponds to a variant of Mealy machines that report the label on the last edge traversed by each input string. Lingg et al. (2024) gave a DFA-adaptation of Gold's proof.

© 2026 R.C. Dumitru, R. Yoshinaka & A. Shinohara.

the input sample satisfies a property that is *characteristic* of the target DFA (Gold, 1967; de la Higuera, 1997). This contrasts with Trakhtenbrot and Barzdin (1973)'s algorithm which returns a smallest DFA provided that the input sample is *uniformly complete*, which is a "structural property", independently of the target DFA to be learned. Along this line, Vázquez de Parga et al. (2016) showed that Trakhtenbrot and Barzdin's algorithm outputs a minimal DFA when the sample is *reasonably complete*, which is a weaker property than the uniform completeness. Furthermore, Zhang (2020) introduced an even weaker condition, called *semantic completeness*, and showed that if the input sample is prefix-closed and semantically complete, then the MinConDFA problem can be solved in polynomial time. These positive results emphasize the role of prefix-closure as a key structural property. However, Zhang (2020) also proved that MinConDFA remains NP-hard for prefix-closed samples that are not semantically complete. In a different but related direction, Ueno et al. (2013) studied the case where the input sample consists of some prefixes of a single string and showed that the problem remains NP-hard in this case.

In this paper, we combine the restrictions studied by Zhang (2020) and Ueno et al. (2013) and consider samples that consist of all prefixes of a single string over a binary alphabet. We show that even under this unified restriction, the problem remains NP-hard. For Moore machines, this setting corresponds to observing the input/output labels along a single run, where both the input and the output alphabets are binary. This models scenarios in which one has trace information from a single execution — e.g., in black-box testing, runtime monitoring, or trace-based reverse engineering — but has no ability to reset or systematically explore the state space. The same result also holds for Mealy machines, where outputs are placed on transitions instead of states. Our results show that the MinConDFA problem stays computationally intractable despite the highly constrained structure of the input data.

We also show that the problem remains hard to approximate within any polynomial ratio when the input consists of a prefix-closed set of multiple runs over a binary alphabet, for finding a smallest DFA, acyclic DFA, Moore machine, or Mealy machine. This strengthens the hardness-of-approximation results of Shimozono et al. (1998), which were established for acyclic DFAs under general input samples. Whether such approximation hardness still holds when the input sample is restricted to the prefix-closed sample of a single run remains an open question for future work. We note that Ueno et al. (2013) obtained a hardness-of-approximation result for DFAs in the setting where the input sample consists of some, but not necessarily all, prefixes of a single run.

2. Preliminaries

We denote the interval between two integers i and j by $[i, j] = \{ k \in \mathbb{Z} \mid i \leq k \leq j \}$.

For an alphabet Σ, the set of strings over Σ is denoted by Σ^*. We write ε for the empty string. The concatenation $s \cdot t$ of two strings s and t is often denoted by st. A string s is a *prefix* of a string w if $w = st$ for some string t. By s^k we denote the k-fold repetition of s. The set of prefixes of a string s is denoted by $\mathrm{Pref}(s)$, which is extended for sets S of strings by $\mathrm{Pref}(S) = \bigcup_{s \in S} \mathrm{Pref}(s)$. The set S is said to be *prefix-closed* if $S = \mathrm{Pref}(S)$. It is *almost prefix-closed* if $S = \mathrm{Pref}(S) - \{\varepsilon\}$. String concatenation is generalized for sets by

$ST = \{\, st \in \Sigma^* \mid s \in S \text{ and } t \in T \,\}$. For a string s and a string set T, define $sT = \{s\}T$. We write $S^* = \{\, s_1 \dots s_k \mid k \geq 0 \text{ and } s_i \in S \text{ for all } i \in [1,k] \,\}$.

A *deterministic finite automaton (DFA)* is a tuple $M = (Q, \Sigma, \delta, F, q_0)$, where Q is the finite set of states, Σ is the input alphabet, $\delta\colon Q \times \Sigma \rightharpoonup Q$ is the state transition function, $F \subseteq Q$ is the set of accepting states, and $q_0 \in Q$ is the initial state. It is *total* if $\delta(q,a)$ is defined for all $q \in Q$ and $a \in \Sigma$. Unless stated otherwise, DFAs are assumed to be total. We write *partial DFA* when totality is not assumed. The state transition function δ is extended to $\delta^*\colon Q \times \Sigma^* \rightharpoonup Q$ by $\delta^*(q,\varepsilon) = q$ and $\delta^*(q,wa) = \delta(\delta^*(q,w),a)$ for $q \in Q$, $w \in \Sigma^*$, and $a \in \Sigma$. We identify the DFA M with its induced partial function from Σ^* to $\{+,-\}$, defined by $M(w) = +$ if and only if $\delta^*(q_0,w) \in F$. We say that M *accepts* a string w if $M(w) = +$ and otherwise *rejects* w. A *DFA sample* is a pair (D_+, D_-) of disjoint finite sets of strings. The sample (D_+, D_-) is *(almost) prefix-complete* if $D_+ \cup D_-$ is (almost) prefix-closed. A DFA M is *consistent* with (D_+, D_-) if and only if M accepts all $s \in D_+$ and rejects all $s \in D_-$.

An *acyclic DFA (ADFA)* is a partial DFA in which the state transition graph is acyclic. That is, there are no state q and nonempty string s such that $\delta^*(q,s) = q$.

A *Moore machine* is a tuple $M = (Q, \Sigma, \Delta, \delta, \rho, q_0)$, where Q is the finite set of states, Σ is the input alphabet, Δ is the output alphabet, $\delta\colon Q \times \Sigma \to Q$ is the state transition function, $\rho\colon Q \to \Delta$ is the (state) output function, and $q_0 \in Q$ is the initial state. The function δ is extended to $\delta^*\colon Q \times \Sigma^* \to Q$ in the same way as in DFAs. We identify the machine M with its induced function from Σ^* to Δ^*, defined by $M(\varepsilon) = \varepsilon$ and $M(wa) = M(w) \cdot \rho(\delta^*(q_0, wa))$ for $w \in \Sigma^*$ and $a \in \Sigma$. A *run* of M over $s \in \Sigma^*$ is the pair $(s, M(s))$.

A *Mealy machine* $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ is defined similarly to a Moore machine, except for the (transition) output function $\lambda\colon Q \times \Sigma \to \Delta$. Again we identify M with its induced function, defined by $M(\varepsilon) = \varepsilon$ and $M(wa) = M(w) \cdot \lambda(\delta^*(q_0,w),a)$. A *run* of M over $s \in \Sigma^*$ is the pair $(s, M(s))$.

A *machine sample set* D is a finite set of string pairs $(s,t) \in \Sigma^* \times \Delta^*$ such that $|s| = |t|$. The set D is *prefix-consistent* if for any $(s_1,t_1), (s_2,t_2) \in D$ and $s \in \mathrm{Pref}(s_1) \cap \mathrm{Pref}(s_2)$, the prefixes of length $|s|$ of t_1 and t_2 coincide. A Moore/Mealy machine M is said to be *consistent* with D if $M(s) = t$ for all $(s,t) \in D$.

Problem 1 (ConDFA, ConADFA, ConMoore, ConMealy) Given a natural number m and a DFA sample D, the problem ConDFA is to decide whether there exists a DFA with at most m states consistent with D. The problems ConADFA, ConMoore, and ConMealy are similarly defined.

Those problems are obviously in NP.

MinConDFA is the optimization version of ConDFA, which is to find a DFA with the minimum number of states consistent with a DFA sample. MinConADFA, MinConMoore, and MinConMealy are similarly defined. Since the number of states of a smallest consistent automaton is bounded by the size of D, MinConDFA and ConDFA are polynomial-time equivalent. We establish NP-hardness for the decision versions and inapproximability for the optimization versions.

In this paper, we are concerned with Moore/Mealy machines whose output alphabet Δ is binary, i.e., $\Delta = \{+,-\}$. The problem ConDFA with prefix-complete instances is

almost identical to ConMoore with a binary output alphabet, except that the output symbol assigned to the initial state is not included in instances of ConMoore. Any prefix-consistent machine sample D can be converted in polynomial time into an equivalent DFA sample (D_+, D_-), which is almost prefix-complete, and vice versa.

Next, we define the graph coloring problem, one of Karp's original 21 NP-complete problems (Karp, 1972). We will reduce this problem to the DFA learning problems in order to prove their NP-hardness.

A *K-labeling* of a simple undirected graph $G = (V, E)$ is a map $\Phi\colon V \to [1, K]$. It is called a *K-coloring* if $\{u, v\} \in E$ implies $\Phi(u) \neq \Phi(v)$. The least number K such that G admits a K-coloring is called the *chromatic number* of G.

Problem 2 (Graph Coloring) Given a simple undirected graph $G = (V, E)$ and an integer K, GraphColoring is to decide whether it admits a K-coloring.

Throughout this paper, we assume without loss of generality that $|E| \geq 2$ and that G has no isolated vertex, since instances with $|E| \leq 1$ are trivially decidable and isolated vertices are colorable with any color.

3. Zhang's reduction

Zhang (2020) has shown that ConDFA is NP-complete when the input sample set over an unbounded alphabet is prefix-closed. We briefly review his proof with a slight modification.

Suppose we are given a graph $G = (V, E)$ and an integer K as an instance of GraphColoring. For $V = \{v_1, \ldots, v_{|V|}\}$, we write $e_{ij} = e_{ji} = \{v_i, v_j\}$. Define a DFA sample (Z_+, Z_-) over $\Sigma = V \cup E$ by

$$Z_+ = \{\varepsilon\} \cup \{\, v_i e_{ij} \in VE \mid i < j \,\},$$
$$Z_- = V \cup \{\, v_j e_{ij} \in VE \mid i < j \,\}.$$

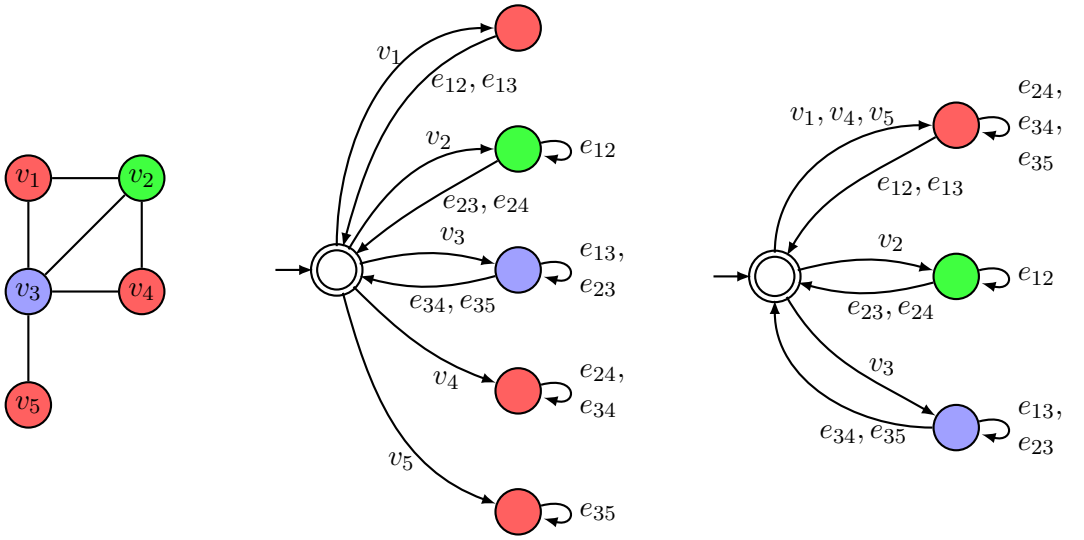
The set $Z_+ \cup Z_-$ is prefix-closed and the total description size is polynomially bounded in the size of the graph. Then, the input graph is K-colorable if and only if there is a DFA with $K + 1$ states consistent with (Z_+, Z_-).

Example 1 Consider the graph shown in Figure 1(*a*) whose chromatic number is $K = 3$. The partial DFAs in Figures 1(*b*) and 1(*c*) are consistent with the sample (Z_+, Z_-) obtained from the graph.

Suppose the input graph G admits a K-coloring function $\Phi\colon V \to [1, K]$. Then, we define a partial DFA $(Q, \Sigma, \delta, F, 0)$ by $Q = [0, K]$, $F = \{0\}$, and

$$\delta(0, v_i) = \Phi(v_i),$$
$$\delta(k, e_{ij}) = \begin{cases} 0 & \text{if } \Phi(v_i) = k \text{ and } i < j, \\ k & \text{if } \Phi(v_j) = k \text{ and } i < j, \end{cases}$$

for all $k \in [1, K]$, $v_i \in V$, and $e_{ij} \in E$. The transition function δ is well-defined since $\Phi(v_i) \neq \Phi(v_j)$ for all $e_{ij} \in E$. Clearly, the DFA is consistent with (Z_+, Z_-).



(*a*) 3-colorable graph (*b*) Consistent partial DFA with $|V| + 1$ states (*c*) Consistent partial DFA with $K + 1$ states

Figure 1: Example of (a modification of) Zhang's reduction.

Conversely, suppose that (Z_+, Z_-) admits a consistent DFA $M = (Q, \Sigma, \delta, F, q_0)$ with $K + 1$ states. Since M accepts some strings, there are at most K rejecting states. We may assume without loss of generality that the rejecting states of M are named $1, \ldots, K'$ for some $K' \leq K$. Since every v_i is rejected, $\delta(q_0, v_i)$ is a rejecting state. Define a function $\Phi \colon V \to [1, K']$ by $\Phi(v_i) = \delta(q_0, v_i)$. For each edge e_{ij} of G with $i < j$, since $v_i e_{ij} \in Z_+$ and $v_j e_{ij} \in Z_-$, we must have $\Phi(v_i) = \delta(q_0, v_i) \neq \delta(q_0, v_j) = \Phi(v_j)$. Thus, Φ is a proper K-coloring.

For more details, see Zhang's original paper.

4. Learning DFAs from prefix-complete samples of binary strings

Based on the reduction presented in the previous section, this section shows that ConDFA and ConADFA are NP-complete when the input alphabet is binary and the sample is prefix-complete, and moreover that MinConDFA and MinConADFA are hard to approximate within any polynomial ratio. Our argument applies to Moore and Mealy machines as well.

A naive attempt to obtain the binary case is to encode the symbols of a large alphabet Σ in binary and to reuse an existing reduction verbatim. Each symbol is replaced by a binary string of length $\lceil \log_2 |\Sigma| \rceil$, so the transitions leaving a state are replaced by a binary tree of auxiliary states. Note, however, that encoding an alphabet in binary does not determine the size of the resulting automaton (Geffert et al., 2024). Two minimal DFAs with the same number of states over the same alphabet may thus be encoded into binary automata whose minimal sizes differ. For example, consider two minimal DFAs $M = (Q, \Sigma, \delta, F, q_0)$ and $M' = (Q, \Sigma, \delta', F, q_0)$ where $Q = \Sigma = [0, 2^n - 1]$, $F = \{0\}$, $q_0 = 0$,

and $\delta(q,i) = (q+i) \bmod 2^n$, while $\delta'(q,2i) = (q+i) \bmod 2^n$ and $\delta'(q,2i+1) = i$. After minimization, the binary encodings of those automata have $|Q|\log_2|\Sigma|$ and $|Q|(|\Sigma|-1)$ states, respectively. The size of a smallest automaton consistent with the encoded sample is thus not determined by that of a smallest one for the original sample. This is harmless when a claim tolerates a factor of $|\Sigma|$, as in the binary-alphabet case of Pitt and Warmuth (1993), but fatal for an argument that turns on an exact bound on the number of states.

4.1. NP-hardness of the decision problem

For an input graph $G = (V, E)$ of GraphColoring, let $\tilde{v}_i, \tilde{e}_{ij} \in \{0,1\}^*$ be binary encodings[2] of lengths $\lceil \log_2 |V| \rceil$ and $\lceil \log_2 |E| \rceil$ for $v_i \in V$ and $e_{ij} \in E$, respectively. The binary encoding involves auxiliary states that disturb the neat correspondence between the number of colors and the number of states in Zhang's reduction. We make the states reached by reading $\tilde{v}_i$ "heavier" than those auxiliary states by inserting many 0's between $\tilde{v}_i$ and $\tilde{e}_{ij}$. Let L be any integer polynomially bounded in $|V|$ and greater than $4|V| + 2|E|\lceil \log_2 |E| \rceil$, and define

$$
\begin{aligned}
S &= \{\, \tilde{v}_i 0^L \tilde{e}_{ij} \mid v_i \in V,\ e_{ij} \in E \,\},\\
S_+ &= \{\, \tilde{v}_i 0^L \mid v_i \in V \,\} \cup \{\, \tilde{v}_i 0^L \tilde{e}_{ij} \mid v_i \in V,\ e_{ij} \in E,\ i < j \,\},\\
S_- &= \mathrm{Pref}(S) - S_+.
\end{aligned}
$$

The sample (S_+, S_-) is prefix-complete and the total description size is polynomially bounded in the size of the graph. We note that the above construction is independent of the given number K of colors. We call the three substrings $\tilde{v}_i$, 0^L, and $\tilde{e}_{ij}$ of $\tilde{v}_i 0^L \tilde{e}_{ij} \in S$ the *head*, the *body*, and the *tail*, respectively.

The tree-shaped automaton in Figure 2 represents the sample obtained from the graph in Figure 1(*a*), where vertices and edges are encoded as $\tilde{v}_1 = 000$, $\tilde{v}_2 = 001$, $\tilde{v}_3 = 010$, $\tilde{v}_4 = 011$, $\tilde{v}_5 = 100$, $\tilde{e}_{12} = 000$, $\tilde{e}_{13} = 001$, $\tilde{e}_{23} = 010$, $\tilde{e}_{24} = 011$, $\tilde{e}_{34} = 100$, and $\tilde{e}_{35} = 101$. The automaton is divided into three parts. The strings of S correspond one-to-one with the paths from the initial state to leaf states. The group of states reached by reading some prefixes of $\tilde{v}_i$ are called the *head*, those reached by $\tilde{v}_i 0^h$ for $h \in [1, L]$ constitute the *body*, and the rest is the *tail*. Let us call a sequence of distinct states $q_1, \ldots, q_k$ a 0-*chain* if $\delta(q_h, 0) = q_{h+1}$ for all $h \in [1, k-1]$. The body of the automaton consists of $|V|$ 0-chains. Each 0-chain is bigger than the total size of the head and the tail, by the choice of L. Therefore, to substantially reduce the size of the automaton, we must merge the states in the body. Recall that $\tilde{v}_i 0^h$ is rejected for every $h < L$ but $\tilde{v}_i 0^L$ is accepted. This means that each 0-chain cannot be folded. We must unify 0-chains. Just like Zhang's reduction, this corresponds to grouping vertices of the graph by distinct color classes. By merging the states of the automaton in Figure 2, we obtain the ADFA in Figure 3. This ADFA has fewer than $(K+1)L = 4L$ states. Some of its states can be further merged to obtain an even smaller automaton, but the size cannot be reduced to $3L$ or less. The number of 0-chains is the dominant factor.

Lemma 1 *A graph G admits a K-coloring if and only if (S_+, S_-) admits a consistent DFA with fewer than $m = (K+1)L$ states.*

2. Collision between $\tilde{v}_i$ and $\tilde{e}_{jk}$ is allowed.

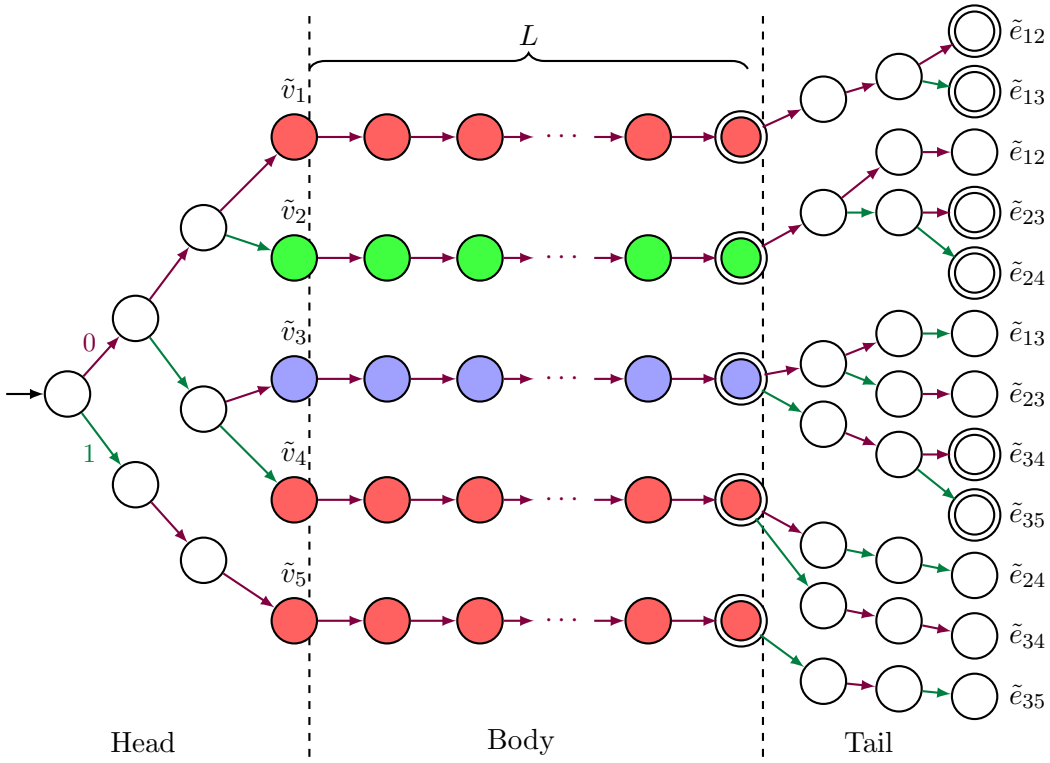


Figure 2: Tree representation of the sample (S_+, S_-) obtained from the graph in Figure 1(*a*). The state marked with $\tilde{v}_i$ is reached by reading $\tilde{v}_i$. The two states marked with $\tilde{e}_{ij}$ are reached by reading $\tilde{v}_i 0^L \tilde{e}_{ij}$ and $\tilde{v}_j 0^L \tilde{e}_{ij}$. The colors of states correspond to the coloring on the graph in Figure 1(*a*).

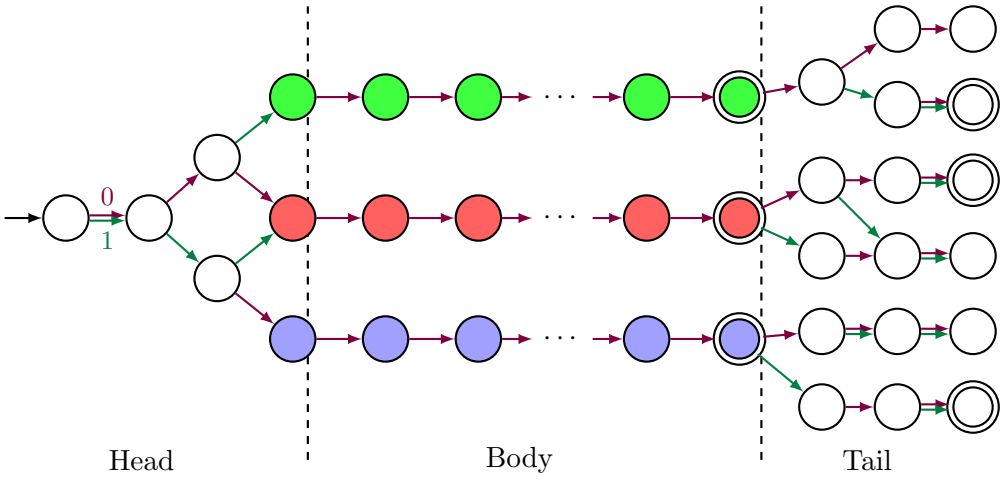


Figure 3: Consistent ADFA with fewer than $(K+1)L = 4L$ states obtained by merging some states of the automaton in Figure 2.

Proof Suppose G admits a K-coloring Φ. We present a consistent acyclic DFA $M = (Q, \Sigma, \delta, F, q_0)$ with fewer than $(K+1)L$ states. By arbitrarily defining transition edges missing from M, one can obtain a (total) DFA with the same number of states. The automaton has K 0-chains of $q_{k,0}, \ldots, q_{k,L} \in Q$ among which only $q_{k,L}$ is accepting for all $k \in [1, K]$. We will traverse those 0-chains when reading the body string 0^L of each example in S. Concerning the head strings $\tilde{v}_i$, the transition function δ is defined so that

$$\delta^*(q_0, \tilde{v}_i) = q_{k,0} \text{ for } k = \Phi(v_i)\,,$$

which requires fewer than $4|V|$ auxiliary states, since this part can be no bigger than the perfect binary tree of height $\lceil \log_2 |V| \rceil$. For the tail strings $\tilde{e}_{ij}$, from the state $q_{k,L}$, the DFA has paths labeled with $\tilde{e}_{ij}$ for $\Phi(v_i) = k$ or $\Phi(v_j) = k$. The terminal state of the path labeled with $\tilde{e}_{ij}$ from $q_{k,L}$ is accepting if and only if $\Phi(v_i) = k$ and $i < j$. Since $e_{ij} \in E$ implies $\Phi(v_i) \neq \Phi(v_j)$, such a construction is possible. At most $2|E|\lceil \log_2 |E| \rceil$ states suffice for the tail part since we have $2|E|$ paths each of which has length $\lceil \log_2 |E| \rceil$. All in all, the DFA M has fewer than $4|V| + KL + 2|E|\lceil \log_2 |E| \rceil \leq m$ states. Note that the term KL counts the number of states in the body part: $q_{k,h}$ for $k \in [1, K]$ and $h \in [1, L]$, whereas the states $q_{k,0}$ are counted in the head part.

Conversely, suppose that (S_+, S_-) admits a consistent DFA with fewer than m states. Let $q_{i,h} = \delta^*(q_0, \tilde{v}_i 0^h)$ for $i \in [1, |V|]$ and $h \in [0, L]$. Since only $\tilde{v}_i 0^L$ is accepted, $q_{i,h} \neq q_{i,h'}$ for any $h < h' \leq L$. On the other hand, it is possible that $q_{i,L} = q_{j,L}$ for distinct i and j. If $q_{i,L} \neq q_{j,L}$, obviously $q_{i,h} \neq q_{j,h'}$ for all $h, h' \leq L$. That is, the 0-chains ending at $q_{i,L}$ and $q_{j,L}$ are disjoint. Since M has fewer than $(K+1)L$ states, it can contain at most K disjoint 0-chains of length L. This implies that the set $\{\, q_{i,L} \mid i \in [1, |V|] \,\}$ has at most K elements. Let us color the vertices of V so that v_i and v_j have the same color if and only if $q_{i,L} = q_{j,L}$. This gives a K-labeling. If $e_{ij} \in E$ with $i < j$, the fact that $\tilde{v}_i 0^L \tilde{e}_{ij} \in S_+$ and $\tilde{v}_j 0^L \tilde{e}_{ij} \in S_-$ implies $q_{i,L} \neq q_{j,L}$. That is, v_i and v_j are colored differently. Therefore, this coloring is a proper K-coloring on the graph. ■

Theorem 2 ConDFA *and* ConADFA *are NP-complete when the input alphabet is binary and the instance sample is prefix-complete.* ConMoore *and* ConMealy *are NP-complete when the input alphabet is binary.*

Proof It does not matter in the above argument whether the empty string is accepted or rejected. Thus, the reduction given above proves the NP-hardness of ConMoore. The same reduction proves the NP-hardness of ConMealy: distinct states are still witnessed by a common suffix yielding different outputs, and a Mealy machine can output + on each transition entering a state that is accepting in the DFA. ■

4.2. Inapproximability

Pitt and Warmuth (1993) showed that polynomial-ratio approximation in polynomial time is hard for MinConDFA unless P = NP. We show that it is still the case when we restrict instance samples to be prefix-complete.

Zuckerman (2007) showed that the chromatic number of a graph is not polynomial-time approximable within a polynomial ratio unless P = NP. Suppose that there is a polynomial-approximation algorithm for MinConDFA with prefix-complete instances. That is, there exists $c > 0$ such that for any instance sample, it is guaranteed that

$$\frac{\hat{m}}{m_*} \leq m_*^c$$

where $\hat{m}$ is the size of the output of the algorithm and m_* is the size of the smallest consistent DFA. Applying the approximation algorithm to the reduction instance in Section 4.1, Lemma 1 implies the graph is $\hat{k}$-colorable for $\hat{k} = \lfloor \hat{m}/L \rfloor$.

Thus, the chromatic number k_* of the graph is approximated by $\hat{k}$ with ratio

$$\frac{\hat{k}}{k_*} = \frac{2\hat{k}L}{2k_*L} \leq \frac{2\hat{k}L}{(k_*+1)L} < \frac{2\hat{m}}{m_*} \leq 2m_*^c$$

by $\hat{k}L \leq \hat{m}$ and $(k_*+1)L > m_*$. Since m_* is polynomially bounded in the graph size, the approximation ratio is bounded by a polynomial. We obtain the inapproximability theorem for MinConDFA.

Theorem 3 *Unless* P = NP*, no polynomial-time algorithm approximates* MinConDFA *or* MinConADFA *within any polynomial ratio on prefix-complete instances over binary alphabets. The same holds for* MinConMoore*, and* MinConMealy *on instances with binary input and output alphabets.*

5. Learning DFAs from prefix-complete samples generated by single binary strings

This section shows the NP-hardness of ConDFA when an instance sample set consists of all and only prefixes of a single string over binary alphabets. Our argument also implies that ConMoore and ConMealy are NP-complete when the instance is just a single pair of strings over binary alphabets. Considering ConADFA under this restriction does not make sense, as the smallest ADFA size must be just the length of the input single string plus one.

Our reduction is a modification of the one we used in Section 4. Without loss of generality, we additionally assume that $\tilde{v}_1 = 0^{\lceil \log_2 |V| \rceil}$. The single string *Str* we use here is obtained by concatenating in any order all the strings in S from Section 4, with many consecutive 0's inserted before each of them. We set this number of 0's to be any integer N such that it is polynomially bounded in $|V|$ and $N > (K+1)L$. We let

$$Str = \prod_{s \in T} s \text{ for } T = 0^N S\,.$$

The positive examples among $\mathrm{Pref}(Str)$ are those corresponding to S_+ and those ending with $s0^i$ for some $s \in T^*$ and $i \in [1, N-1]$, i.e.,

$$D_+ = \mathrm{Pref}(Str) \cap T^*(0^{[1,N-1]} \cup 0^N S_+)$$

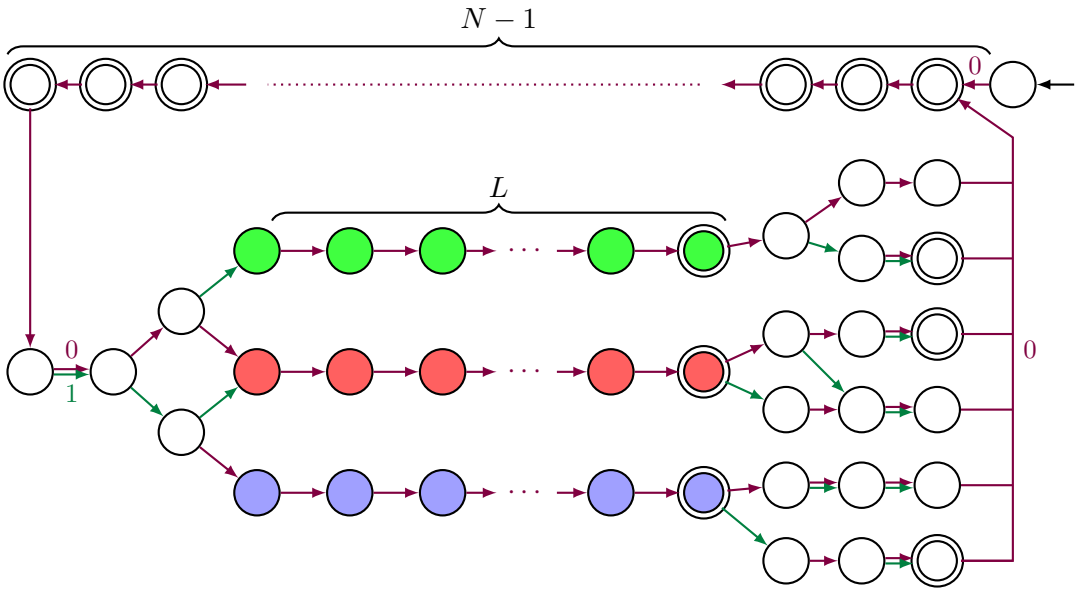


Figure 4: DFA consistent with the sample set obtained from the graph in Figure 1(*a*).

where $0^{[i,j]} = \{\, 0^k \mid k \in [i,j] \,\}$. The negative examples are $D_- = \mathrm{Pref}(Str) - D_+$. The purpose of the long substring 0^N is to force the automaton to return to the same state each time an element of T^*0^N is read. When reading a prefix 0^j of 0^N with $j < N$, we traverse accepting states and then reach a rejecting state when we finish reading 0^N. The substring 0^N repeatedly occurs in Str as a suffix of a negative example in T^*0^N. If a consistent DFA has fewer than $2N$ states, any two 0-chains of length N must share a state; they may differ only in their initial parts, and in particular they end in the same rejecting state. In contrast to the "short" 0-chains of length L for the body strings, the long chain consists mostly of accepting states except the last one. As a result, no short 0-chains can be embedded within the long 0-chain. Figure 4 shows a DFA consistent with the sample set obtained from the graph in Figure 1(*a*).

Lemma 4 *A graph G admits a K-coloring if and only if (D_+, D_-) admits a consistent DFA with fewer than $m = N + (K+1)L$ states.*

Proof Suppose that the graph admits a proper K-coloring. We give a consistent DFA M' by modifying the DFA M given in the proof of Lemma 1. We introduce N new states $q_{0,0}, \ldots, q_{0,N-1}$ that form a 0-chain together with the initial state of M, where $q_{0,0}$ is the new initial state and the 0-chain ends in the initial state of M. In addition, we add a 0-transition from every leaf state q of M to $q_{0,1}$. Here, the leaf states are those reached by reading elements of S in M. This is a consistent DFA with N more states than M.

Conversely, suppose there is a consistent DFA M with fewer than $N + (K+1)L$ states. The presence of the prefix 0^N where 0^N is rejected and 0^j are accepted for all $j \in [1, N-1]$ implies that M must have a 0-chain $q_{0,1}, \ldots, q_{0,N}$ where $\delta^*(q_0, 0^i) = q_{0,i}$ for the initial state q_0 for all $i \in [1, N]$, $q_{0,i}$ are accepting for all $i \in [1, N-1]$, and $q_{0,N}$ is rejecting. Consider the state reached by reading a string of the form $s0^N \in \mathrm{Pref}(Str)$ for some $s \in T^*$. If the

state is not $q_{0,N}$, we must have a 0-chain of length N disjoint from $q_{0,1}, \ldots, q_{0,N}$, which means that the DFA has at least $2N > m$ states, a contradiction. Therefore, each time we read a string in T^*0^N, we must come back to the same state $q_{0,N}$.

Let $q_{i,h} = \delta^*(q_{0,N}, \tilde{v}_i 0^h)$ for $i \in [1, |V|]$ and $h \in [0, L]$. Due to the encoding $\tilde{v}_1 = 0^{\lceil \log_2 |V| \rceil}$, we have a 0-chain of distinct states $q'_h = \delta^*(q_{0,N}, 0^h)$ for $h \in [0, L + \lceil \log_2 |V| \rceil]$, among which $q'_{L+\lceil \log_2 |V| \rceil}$ is the unique accepting state. Note that $q'_0 = q_{0,N}$ and $q'_{\lceil \log_2 |V| \rceil + h} = q_{1,h}$. Since $\delta^*(q'_{h'}, 0^{L-h}) \notin F$ and $\delta^*(q_{i,h}, 0^{L-h}) \in F$, $q_{i,h} \neq q'_{h'}$ for any $h \in [0, L]$ and $h' \in [0, \lceil \log_2 |V| \rceil - 1]$. Let us consider when the long 0-chain $q_{0,j}$ with $j \in [1, N]$ and a short 0-chain $q_{i,h}$ with $h \in [0, L]$ may share the same state. Due to the acceptance and rejection separation, $q_{i,h} = q_{0,j}$ only if either $j = N$ and $h < L$, or $j < N$ and $h = L$, but we have already argued that the former is impossible by $q_{0,N} = q'_0$. So the only possibility is $q_{i,L} = q_{0,j}$ for some j. Let $Q' = \{\, q_{i,L} \mid i \in [1, |V|] \,\}$. By the same argument as the proof of Lemma 1, we have at least $N + |Q'|(L+1) + \lceil \log_2 |V| \rceil - 1 - |Q'|$ states, where the term $\lceil \log_2 |V| \rceil - 1$ comes from the 0-chain q'_h for $h \in [1, \lceil \log_2 |V| \rceil - 1]$, and the additional minus $|Q'|$ comes from the fact that each element of Q' may lie on the long 0-chain. Since we have fewer than m states, we conclude that

$$N + |Q'|L + \lceil \log_2 |V| \rceil - 1 < N + (K+1)L\,,$$

so $|Q'| \leq K$. Therefore, the graph G is K-colorable by the same argument as in the proof of Lemma 1.

■

Theorem 5 ConDFA *is NP-complete when the input alphabet is binary and the instance sample consists of all and only the prefixes of a single string.* ConMoore *and* ConMealy *are NP-complete when the input is a single run over a binary alphabet.*

We note that, unlike Lemma 1, Lemma 4 cannot be extended to show polynomial-ratio inapproximability. In our reduction, any consistent DFA requires at least N states, while the trivial $|V|$-coloring gives a consistent DFA with at most $N + (|V| + 1)L$ states. Since the reduction for the approximation problem must set $N > (|V| + 1)L$ (as no target value K is given in the approximation problem), this trivial consistent DFA has fewer than twice the number of states of a smallest consistent DFA. Note that even if we could use a tighter known upper bound $\hat{k} \leq |V|$ on the chromatic number and set $N = (\hat{k} + 1)L + 1$, there would exist a consistent DFA with at most $N + (\hat{k} + 1)L$ states, which is again less than twice the size of a smallest consistent DFA. This approach does not even rule out a factor-2 approximation.

6. Conclusion and future work

Expanding upon previous work, we have narrowed the class of instances for which finding a machine with the minimum number of states consistent with the sample remains intractable. Using a previous reduction from a graph problem, we proved that ConDFA, ConMoore, and ConMealy are NP-complete when the sample is prefix-complete over a binary alphabet and consists of all prefixes of a single string. Dropping the single-string restriction, we also

proved, assuming P $\neq$ NP, that MinConDFA, MinConADFA, MinConMoore, and MinConMealy cannot be approximated within any polynomial ratio by a polynomial-time algorithm.

The main question left open by our work is whether the inapproximability of Theorem 3 survives the single-string restriction of Section 5. Here the two requirements appear to be in tension, at least under our approach: the long 0^N padding that makes the single-string reduction possible necessarily dominates the automaton's size, swamping the coloring-dependent term and capping at two the inapproximability ratio that this construction can certify. Establishing inapproximability would require a construction in which the coloring-dependent part, rather than a fixed dominating chain, governs the optimum size; whether such a construction exists, or whether the single-string case instead admits a good approximation, remains unclear. The fact that the prefixes of a single string form a path rather than a tree suggests recasting the problem as a constrained partition of $[0, |Str|]$, which may be more amenable to approximation.

More broadly, since our results push the natural restrictions on the input quite far, the complementary task of designing algorithms that solve or approximate the problem efficiently on well-behaved samples seems the more promising long-term direction.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Numbers JP23K11325, JP24H00697, JP24K14827, JP25K14981 (RY) and JP25K14979 (AS).

A part of Radu Cosmin Dumitru's work was done when he was visiting Tohoku University as an exchange student.

References

Dana Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39(3):337–350, 1978. doi: 10.1016/S0019-9958(78)90683-6.

Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. Pre-reduction graph products: Hardnesses of properly learning DFAs and approximating EDP on DAGs. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 444–453, 2014. doi: 10.1109/FOCS.2014.54.

Colin de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27:125–138, 1997. doi: 10.1023/A:1007353007695.

Viliam Geffert, Dominika Pališínová, and Alexander Szabari. State complexity of binary coded regular languages. *Theoretical Computer Science*, 990:114399, 2024. doi: 10.1016/j.tcs.2024.114399.

E Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967. doi: 10.1016/S0019-9958(67)91165-5.

E Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978. doi: 10.1016/S0019-9958(78)90562-4.

Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972. doi: 10.1007/978-1-4684-2001-2_9.

Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the Abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In *Grammatical Inference*, pages 1–12, 1998. doi: 10.1007/BFb0054059.

Jonas Lingg, Mateus de Oliveira Oliveira, and Petra Wolf. Learning from positive and negative examples: New proof for binary alphabets. *Information Processing Letters*, 183: 106427, 2024. doi: 10.1016/j.ipl.2023.106427.

Jose Oncina and Pedro García. Inferring regular languages in polynomial update time. *Pattern Recognition and Image Analysis*, pages 49–61, 1992. doi: 10.1142/9789812797902_0004.

Leonard Pitt and Manfred K. Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of ACM*, 40(1):95–142, 1993. doi: 10.1145/138027.138042.

Shinichi Shimozono, Kouichi Hirata, and Ayumi Shinohara. On the hardness of approximating the minimum consistent acyclic DFA and decision diagram. *Information Processing Letters*, 66(4):165–170, 1998. doi: 10.1016/S0020-0190(98)00065-9.

B. A. Trakhtenbrot and Ya. M. Barzdin. *Finite Automata: Behavior and Synthesis*. North Holland Publishing Company, 1973.

Kaori Ueno, Shinichi Shimozono, Kazuyuki Narisawa, and Ayumi Shinohara. On the hardness of approximating the minimum consistent DFA from prefix samples. In *ICALP 2013 Satellite Workshop on Learning Theory and Complexity*, 2013.

Manuel Vázquez de Parga, Pedro García, and Damián López. Minimal consistent DFA revisited. *Theoretical Computer Science*, 647:43–49, 2016. doi: 10.1016/j.tcs.2016.07.023.

Chenyi Zhang. Minimal consistent DFA from sample strings. *Acta Informatica*, 57:657–670, 2020. doi: 10.1007/s00236-020-00365-8.

David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007. doi: 10.4086/TOC.2007.V003A006.