# Learning from Positive and Negative Examples:
# New Proof for Binary Alphabets

**Jonas Lingg** *Eberhard Karls Universität Tübingen, Germany*          J.LINGG@MAILBOX.ORG

**Mateus de Oliveira Oliveira**[*] *University of Bergen, Norway*          MATEUS.OLIVEIRA@UIB.NO

**Petra Wolf**[†] *Universität Trier, Germany*          WOLFP@INFORMATIK.UNI-TRIER.DE

## Abstract

One of the most fundamental problems in computational learning theory is the problem of learning a finite automaton $A$ consistent with a finite set $P$ of positive examples and with a finite set $N$ of negative examples. By consistency, we mean that $A$ accepts all strings in $P$ and rejects all strings in $N$. It is well known that this problem is NP-complete. In the literature, it is stated that NP-hardness holds even in the case of a binary alphabet. As a standard reference for this theorem, the work of Gold from 1978 is either cited or adapted. But as a crucial detail, the work of Gold actually considered Mealy machines and not deterministic finite state automata (DFAs) as they are considered nowadays. As Mealy automata are equipped with an output function, they can be more compact than DFAs which accept the same language. We show that the adaptations of Gold's construction for Mealy machines stated in the literature have some issues, and provide a correct proof for the fact that the *DFA-consistency problem* for binary alphabets is NP-complete.

**Keywords:** Learning Automata, Computational Complexity, Finite Samples

## 1. Introduction

In the *DFA-consistency problem* (DFA-CON) we are given a pair of disjoint sets of strings $P, N \subseteq \Sigma^*$ and a positive integer $k$. The goal is to determine whether there is a deterministic finite automaton (DFA) $A$ with at most $k$ states that accepts all strings in $P$ and rejects all strings in $N$. Even though Pitt and Warmuth (1993) showed that the problem cannot be approximated within any polynomial factor, DFA-CON has become one of the most central problems in computational learning theory (see Gold (1967); Angluin (1978); Gold (1978); Pitt (1989); Parekh and Honavar (2001)) with applications that span several subfields of artificial intelligence and related areas, such as automated synthesis of controllers Ramadge and Wonham (1987), model checking Groce et al. (2002); Mao et al. (2016) optimization Najim et al. (1990); Yazidi et al. (2020); Bouhmala (2015); Coste and Kerbellec (2006) neural networks Meybodi and Beigy (2002); Hasanzadeh-Mofrad and Rezvanian (2018); Mayr and Yovine (2018); Guo et al. (2019), multi-agent systems Nowé et al. (2005), and others Rezvanian et al. (2018); Najim and Poznyak (2014); De la Higuera (2010).

The DFA-CON problem has been studied for at least five decades, including parameterized analysis by Fernau et al. (2015), and it was stated multiple times that the NP-completeness of the DFA-CON problem also holds in the case of binary alphabets, see Angluin (1978); de la Higuera (2010); Fernau et al. (2015). In Angluin (1978), the work of Gold

(1978) is cited for this fact and in de la Higuera (2010) and Fernau et al. (2015) adaptations of the construction from Gold (1978) are given. The issue is that NP-hardness results for variants of DFA-Con considered in Gold (1978) are actually stated in terms of Mealy machines, and do not translate directly to the modern context of DFAs in the case of binary alphabets. By taking a closer look on the preliminaries, Angluin (1978) is also considering Mealy machines. For the adaptations in de la Higuera (2010) and Fernau et al. (2015), both of them follow the same approach, and both contain inaccuracies that invalidate the proofs. They have in common that they are adaptations of the construction by Gold for the consistency problem of Mealy machines Gold (1978) but as the Mealy machines considered in Gold (1978) (mapping $\Sigma^* \to \Gamma$, $|\Gamma| = 2$) can be more compact (when interpreted as language acceptors) than DFAs recognizing the same language, the difference in number of states causes the adaptations to fail.

We solve this issue by giving a new construction for the claim that the DFA-Con problem is NP-complete when restricted to binary alphabets.

This work is structured as follows. After giving necessary definitions, we present our main result, a new construction for the NP-hardness of DFA-Con restricted to binary alphabets. Then, we discuss in more detail why the reductions in de la Higuera (2010); Fernau et al. (2015) fail and why the construction of Gold does not directly transfer to the setting of DFAs.

We thank an anonymous reviewer of LearnAut 2022 for suggesting a simpler construction than we had initially.

## 2. Preliminaries

For a finite alphabet $\Sigma$, we call $\Sigma^*$ the set of all words over $\Sigma$. For a regular expression $r$ and a fixed number $k$, we denote $k$ concatenations of $r$ with $r^k$. A *deterministic finite automaton* (DFA) is a tuple $A = (Q, \Sigma, \delta, s_0, F)$ where $Q$ is a finite set of states, $\Sigma$ a finite alphabet, $\delta : Q \times \Sigma \to Q$ a total transition function, $s_0$ the initial state and $F \subseteq Q$ the set of final states. We call $A$ a *complete* DFA if we want to highlight that $\delta$ is a *total* function. If $\delta$ is partial, we call $A$ a partial DFA. We generalize $\delta$ to words by $\delta(q, aw) = \delta(\delta(q, a), w)$ for $q \in Q, a \in \Sigma, w \in \Sigma^*$. We further generalize $\delta$ to sets of input letters $\Gamma \subseteq \Sigma$ by $\delta(q, \Gamma) = \bigcup_{\gamma \in \Gamma} \{\delta(q, \gamma)\}$. A DFA $A$ accepts a word $w \in \Sigma^*$ if and only if $\delta(s_0, w) \in F$. We let $\mathcal{L}(A)$ denote the *language of $A$*, i.e., the set of all words accepted by $A$. We denote the empty word with $\epsilon$, $|\epsilon| = 0$. The DFA-Con problem is formally defined as follows.

**Definition 1 (DFA-Con)**
*Input: Finite set of words $P, N \subseteq \Sigma^*$ with $P \cap N = \emptyset$, and integer $k$.*
*Question: Is there a DFA $A = (Q, \Sigma, \delta, s_0, F)$ with $|Q| \leq k$, $P \subseteq \mathcal{L}(A)$, and $\mathcal{L}(A) \cap N = \emptyset$?*

## 3. Main Result

As our main result, we present an adaptation for DFAs of the reduction by Gold.

**Theorem 2** *Restricted to binary alphabets,* DFA-Con *is* NP-complete.

We show that there is a reduction from ALL-POS-NEG 3SAT[1] to DFA-CON mapping each instance of ALL-POS-NEG 3SAT with $n \geq 1$ variables and $m \geq 1$ clauses to an instance of DFA-CON with $k = n + m$, $|\Sigma| = 2$, $|P| = \mathcal{O}(m)$, and $|N| = \mathcal{O}(m^2 + nm)$ in time $\mathcal{O}(n^3 + n^2 m)$.

Let $\phi$ be a Boolean formula in 3CNF with $n \geq 1$ variables

$$V = \{x_0, x_1, \ldots, x_{n-1}\}$$

and $m \geq 1$ clauses $C = \{c_0, c_1, \ldots, c_{m-1}\}$, where each clause contains either only positive literals or only negative literals. Given $\phi$, we let $k = n + m$ and

$$
\begin{aligned}
P = & \{\epsilon, a^k\} \cup \{a^i bb \mid i \in [0, \ldots, m-1], C_i \text{ positive}\} \\
N = & \{a^i \mid 0 < i < k\} \\
& \cup \{a^i bb \mid i \in [0, \ldots, m-1], C_i \text{ negative}\} \\
& \cup \{a^i ba^{k-r} \mid (0 \leq i \leq m-1) \wedge (0 \leq r < k) \wedge (r < m \vee x_{r-m} \notin C_i)\}
\end{aligned}
$$

be the corresponding instance of DFA-CON.

We show that there exists a satisfying variable assignment

$$\beta \colon V \to \{\texttt{false}, \texttt{true}\}$$

if and only if there exists a DFA with at most $k$ states that is consistent with $P$ and $N$.

We start with a remark on the structure of a DFA $A$ with at most $k$ states accepting $\{\epsilon, a^k\}$ and rejecting $\{a^i \mid 0 < i < k\}$. Let $s_i$ be the state of $A$ reached after reading $a^i$. For each two distinct $i$ and $j$ in $\{0, 1, \ldots, k-1\}$, $A$ accepts after reading $a^{k-i}$ from $s_i$, but rejects after reading $a^{k-j}$ from $s_i$. Therefore, the states $s_0, \ldots, s_{k-1}$ are pairwise distinct. This implies that $s_k = s_0$, and that $A$ consists of a loop of $k$ states $s_0, s_1, \ldots, s_{k-1}$ where $s_0$ is the sole initial and the sole accepting state, and for each $i \in \{0, 1, \ldots, k-1\}$, $s_i$ transitions to $s_{i+1 \mod k}$ when reading letter $a$ (see Figure 1).
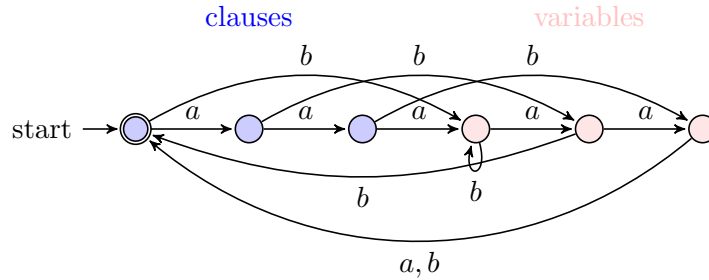


Figure 1: Automaton for $\phi = \neg x_0 \wedge x_1 \wedge x_2$.

The intuition is that the first $m$ states, $s_0, s_1, s_{m-1}$, correspond to the clauses $C_0, C_2, \ldots, C_{m-1}$ and the $n$ last states, $s_m, s_{m+1}, \ldots, s_{m+n-1}$, correspond to the variables $x_0, x_1, \ldots, x_{n-1}$.

1. Given a Boolean formula in conjunctive normal form where all clauses have at most three literals (this form is called 3CNF) and all literals in a clause are either all positive or all negative. Is there a variable assignment satisfying the formula?

We start by showing that if $\phi$ is satisfiable, then there exists a DFA $A$ with $k$ states consistent with $P$ and $N$. Let $\beta\colon V \to \{\texttt{false}, \texttt{true}\}$ be a satisfying variable assignment. We let $A = (\{s_0, s_1, \ldots, s_{k-1}\}, \{a, b\}, \delta, s_0, \{s_0\})$. For each $i \in \{0, 1, \ldots, k-1\}$, we let $\delta(s_i, a) = s_{i+1 \mod k}$. The transitions corresponding to letter $b$ depend on the satisfying assignment $\beta$. For each $i \in \{0, 1, \ldots, m-1\}$, let $x_j$ be a variable in $C_i$ such that its assignment under $\beta$ satisfies $C_i$. Note that since $\beta$ is a satisfying assignment, such variable always exists. We set $\delta(s_i, b) = s_{m+j}$. Intuitively, if the automaton is at the state $s_i$ corresponding to clause $C_i$, then after reading symbol $b$ it transitions to the state $s_{m+j}$ corresponding to variable $x_j$. For each $j \in \{0, 1, \ldots, n-1\}$, if $\beta(x_j) = \texttt{true}$, then we set $\delta(s_{m+j}, b) = s_0$. Otherwise, we set $\delta(s_{m+j}, b) = s_{m+j}$. Intuitively, if the automaton is at the state $s_{m+j}$ corresponding to variable $x_j$, then after reading symbol $b$, it transitions to the accepting state $s_0$ if $x_j$ is set to $\texttt{true}$ and to the state $s_{m+j}$ otherwise (note that $s_{m+j}$ is a rejecting state). See Figure 1 for an example.

Now, we show that $A$ accepts all words in $P$ and rejects all words in $N$. By construction, $A$ accepts $\{\epsilon, a^k\}$ and rejects $\{a^i \mid 0 < i < k\}$. Let $a^i bb \in \{a^i bb \mid i \in [0, \ldots, m-1], C_i \text{ positive}\}$. After reading $a^i$, $A$ transitions to state $s_i$ corresponding to the positive clause $C_i$. By construction, reading $b$ from $s_i$ leads to the state $s_{m+j}$ corresponding to a variable $x_j$ contained in $C_i$ such that $\beta(x_j) = \texttt{true}$. Therefore, $\delta(\delta(s_i, b), b) = s_0$ and $A$ accepts $a^i bb$. Similarly, $A$ rejects all the words in $\{a^i bb \mid i \in [0, \ldots, m-1], C_i \text{ negative}\}$. Finally, $A$ rejects all the words in the set

$$\{a^i ba^{k-r} \mid (0 \le i \le m-1) \wedge (0 \le r < k) \wedge (r < m \vee x_{r-m} \notin C_i)\}. \tag{1}$$

To see this, note that for each $i \in \{0, \ldots, m-1\}$, the state reached by the string $a^i b$ is the state $s_{m+j}$ corresponding to some variable $x_j$ in $C_i$. Therefore, given $r \in \{0, 1, \ldots, k-1\}$, the string $a^i ba^{k-r}$ is accepted if and only if $r = m+j$. Nevertheless, this is never the case, since either $r < m$, or $m \le r < k$ and $x_{r-m} \notin C_i$.

For the converse, suppose that there exists a DFA $A$ with at most $k$ states consistent with $P$ and $N$. As we saw already, $A$ has exactly $k$ states $s_0, s_1, \ldots, s_{k-1}$ that form a loop with the letter $a$, and $s_0$ is both the initial state and the unique accepting state. For each $i \in \{0, 1, \ldots, m-1\}$, the negative samples in (1) ensure that the transition with $b$ from state $s_i$ points to the state $s_{m+j}$ corresponding to a variable $x_j$ contained in $C_i$, since if this were not the case, some string in the set (1) would have been accepted by $A$. Now, we define an assignment $\beta$ of variables depending on the transitions labelled with $b$. For each $j \in \{0, 1, \ldots, n-1\}$, if the transition labeled by $b$ going from state $s_{m+j}$ points to the accepting state $s_0$, then we set $\beta(x_j) = \texttt{true}$. Otherwise, we set $\beta(x_j) = \texttt{false}$. For each $i \in \{0, 1, \ldots, m-1\}$, let $C_i$ be a positive clause in $\phi$. By consistency, $A$ accepts the word $a^i bb$. Note that after reading string $a^i$, $A$ reaches the state $s_i$ corresponding to $C_i$, and then, by reading $b$ the automaton transitions to the state $s_{m+j}$ corresponding to some variable $x_j$ contained in $C_i$. Finally, after reading the last $b$ the automaton transitions to the unique accepting state $s_0$. Therefore, by construction, $\beta(x_j) = \texttt{true}$ and the assignment of $x_j$ satisfies $C_i$. A similar argument works if $C_i$ is a negative clause. Therefore $\beta$ satisfies $\phi$. $\square$

## 4. Counterexamples Regarding DFA-Con with $|\Sigma| = 2$

In the literature, several proofs for the NP-hardness of DFA-Con for fixed alphabet size of $|\Sigma| = 2$ have appeared. We give in the following a counterexample which contradict the proofs in Fernau et al. (2015) and in de la Higuera (2010). We further discuss in this section why the claim does not follow directly from the construction in Gold (1978).
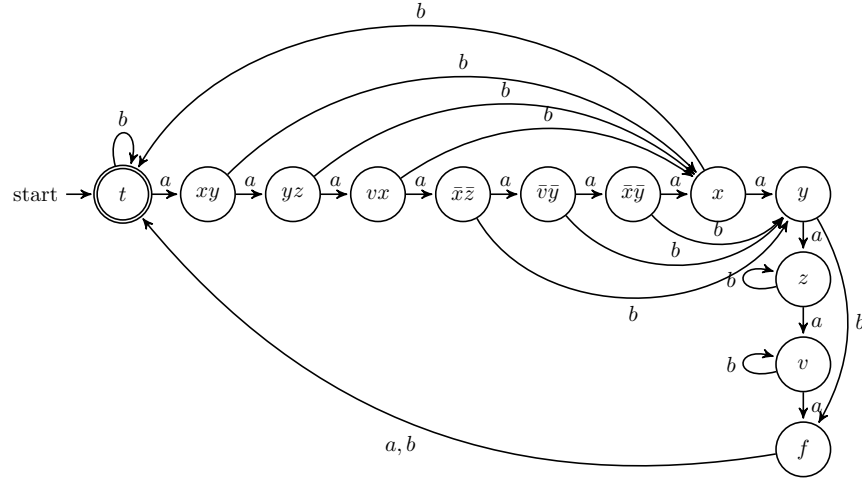
**Counterexample (Fernau et al. (2015))**



Figure 2: Automaton corresponding to $\phi$.

We give a Boolean formula that is unsatisfiable and shows that the construction used in the proof of Lemma 15 in Fernau et al. (2015) allows us to construct an automaton that is consistent with $P$ and $N$. Consider the unsatisfiable formula

$$\phi = (x \vee y) \wedge (y \vee z) \wedge (v \vee x) \wedge (\neg x \vee \neg z) \wedge (\neg v \vee \neg y) \wedge (\neg x \vee \neg y)$$

with $n = 4$ variables and $m = 6$ clauses. We index the clauses from left to right in the formula above, starting with $i = 1$. We construct the DFA-Con instance $P, N \subseteq \{a, b\}^*$, $k = m + n + 2 = 12$ according to the construction presented in the proof of Lemma 15 in Fernau et al. (2015):

$$P = \{\epsilon, b, a^{12}, a^{11}b\} \cup \{a^i bbb \mid 1 \leq i \leq 6\} \cup \{a^i bb \mid 1 \leq i \leq 3\} \cup \{a^i bba \mid 4 \leq i \leq 6\},$$

$$N = \{a^k \mid 1 \leq k \leq 11\} \cup \{a^i b, a^i ba \mid 1 \leq i \leq 6\} \cup \{a^i bb \mid 4 \leq i \leq 6\}.$$

Figure 2 shows a DFA with 12 states over $\{a, b\}$ that is consistent with $P$ and $N$ despite the formula $\phi$ being unsatisfiable. This refutes Lemma 15 in Fernau et al. (2015).

**Counterexample (de la Higuera (2010))**

The set of clauses $C = \{\{\neg x_2, \neg x_3, \neg x_5\}_1, \{\neg x_8\}_2, \{x_1, x_4, x_8\}_3, \{x_1, x_6, x_8\}_4, \{x_6, x_7, x_8\}_5, \{\neg x_4, \neg x_6\}_6, \{\neg x_1, \neg x_6\}_7, \{\neg x_1, \neg x_7\}_8\}$ represents an *unsatisfiable* Boolean formula $\phi$ with

$n = 8$ variables and $m = 8$ where each clause is either purely positive or purely negative. The index imposes an ordering on the clauses. The variable-set $V = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ is also ordered. In accordance with the construction in the proof of Theorem 6.2.1 in de la Higuera (2010), page 120, we define the sets

$$
\begin{aligned}
P =& \{a^8\} \cup \{a^{i-1}ba^8b \mid i \in \{3, 4, 5\}\}, \\
N =& \{a^k, a^{8+k} \mid 1 \leq k < 8\} \cup \{a^{i-1}ba^8b \mid i \in \{1, 2, 6, 7, 8\}\} \\
& \cup \{a^{i-1}ba^{8-j+1} \mid i \in \{3, 4, 5\}, x_j \notin C_i\} \\
& \cup \{a^{i-1}ba^{8-j+1} \mid i \in \{1, 2, 6, 7, 8\}, \neg x_j \notin C_i\}.
\end{aligned}
$$

using the two orderings that are implicit in the proof for Theorem 6.2.1 in the textbook. Claim 1 in the proof of Theorem 6.2.1 claims that every DFA consistent with $P$ and $N$ has at least $n + 1$ states. In contrast, Claim 3 speaks about the existence of an $n$-state DFA. The precise state-bound of the constructed instance in the proof of Theorem 6.2.1 is not explicitly given. We show that the construction with the state-bound given in Claim 1 cannot be correct by giving an $(n + 1)$-state DFA in Figure 3, which is consistent with the sets $P$ and $N$ obtained from an unsatisfiable formula. Also for a bound of exact $n$ states, this proof is not correct, as the example discussed with respect to Gold's construction below in this section is also a counterexample for the construction of Theorem 6.2.1 in de la Higuera (2010). Consider the formula $\varphi = \neg x_1 \wedge x_2 \wedge x_3$ with clause set $\{\{\neg x_1\}, \{x_2\}, \{x_3\}\}$ and variable set $\{x_1, x_2, x_3\}$ implicitly ordered from left to right. Here, $m = 3$ and $n = 3$. This formula is clearly satisfiable and has pure clauses, but there is no DFA with 3 states that can separate the two sets of words of the de la Higuera construction properly. Let's call the three states $q_1$, $q_2$, $q_3$. Since $a^3 \in P$ and $\{a^1, a^2, a^4, a^5\} \subseteq N$ we get for $a$: $\delta(q_1, a) = q_2$, $\delta(q_2, a) = q_3$, $\delta(q_3, a) = q_1$ and $q_1$ must be an accepting state. Now, consider the transition with $b$ for the state $q_2$. Since the second clause $\{x_2\}$ ($i = 2$) is positive we have by the third item of the construction $a^{i-1}ba^nb \in P$, hence $a^1ba^3b \in P$. By the second condition of this item we have for $x_3 \notin \{x_2\}$: $a^{i-1}ba^{n-j+1} \in N$, hence $a^1ba^1 \in N$. From this point we also get for $x_1 \notin \{x_2\}$: $a^1ba^3 \in N$. These three words now lead to a contradiction concerning $\delta(q_2, b)$. We cannot have $\delta(q_2, b) = q_3$ since then $\delta(q_1, aba) = q_1$, which contradicts $aba \in N$. We cannot have $\delta(q_2, b) = q_1$ since then $\delta(q_1, abaaa) = q_1$, which contradicts $abaaa \in N$. Hence, we must have $\delta(q_2, b) = q_2$. But then, $\delta(q_1, a^1ba^3b) = q_2$, which contradicts $abaaab \in P$. Hence, we cannot find a three-state DFA consistent with the construction.

### General Issue

Both of the constructions (Fernau et al. (2015) and de la Higuera (2010)) rely on the construction in the proof of Theorem 2 in Gold (1978) where a reduction from the 'Satisfiability Question' to the problem 'Is there a finite automaton with states reachable by $T$ which agrees with $D$?' is given. Here, $T$ is a finite set of words representing states, i.e., the words in $T$ are elements of the equivalence classes representing states of the sought automaton; moreover no two elements of $T$ are in the same class. The finite set $D$ (set of data) consists of pairs of words over the input alphabet and single output letters. The key point is that in Gold (1978) the term automaton refers to 'Mealy model finite state automaton' and hence, the automata do not include a set of final states, instead they are enhanced with
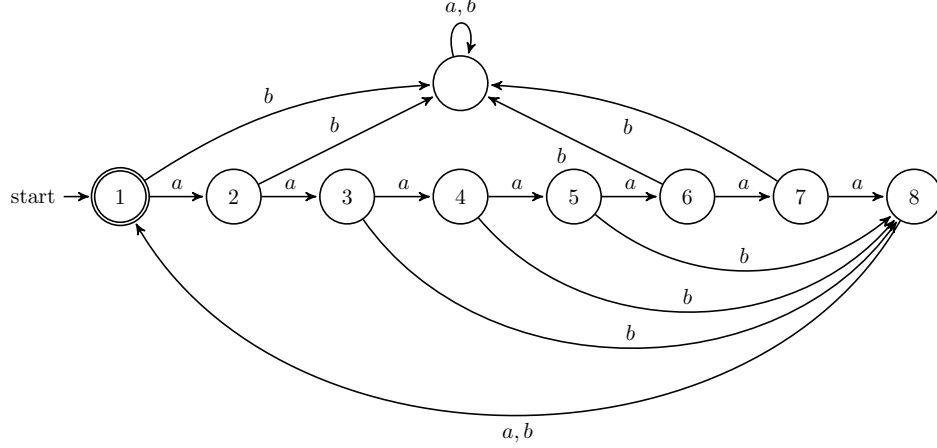
Figure 3: Automaton with $n+1 = 9$ states corresponding to the clause set $\{\{\neg x_2, \neg x_3, \neg x_5\}, \{\neg x_8\}, \{x_1, x_4, x_8\}, \{x_1, x_6, x_8\}, \{x_6, x_7, x_8\}, \{\neg x_4, \neg x_6\}, \{\neg x_1, \neg x_6\}, \{\neg x_1, \neg x_7\}\}$.

an output function, which can be interpreted as indicating the acceptance or rejection of a word. For a binary output alphabet, these two models are equivalent considering the class of recognizable languages, but the number of states needed to recognize a language may differ. For instance, the language $\{0, 1\}^*0$ can be accepted with a one-state Mealy automaton but for DFAs it can only be accepted by a two-state DFA. Coming back to the consistency problem for DFAs, one is tempted to think that a proof for the NP-hardness for DFA-CON is hidden in the corollary following Theorem 2 in Gold (1978) ('$Q_{\min}(D, n)$ is NP-complete for $\mathrm{Card}(U) = \mathrm{Card}(Y) = 2$'; $U$ is the input alphabet and $Y$ the output alphabet and $Q_{\min}(D, n)$ asks, given data $D$ and positive integer $n$, is there a finite automaton with $n$ states which agrees with $D$?) but with the arguments above, this is not the case. For instance, we can give a counterexample to this claim. Consider the satisfiable formula $\varphi_1 = \neg x_1 \wedge x_2 \wedge x_3$ with $n = 3$ and implicit variable order $x_1 < x_2 < x_3$ as well as clause order $\{\neg x_1\} < \{x_2\} < \{x_3\}$. Following the construction in Theorem 2 in Gold (1978), one can verify that there is a three-state Mealy machine recognizing the words in the state characterization matrix correctly, but there is no three-state DFA consistent with the data. Another counterexample is the satisfiable formula $\varphi_2 = x_1 \wedge x_3 \wedge \neg x_2$ with variable order $x_1 < x_3 < x_2$ and clause order $\{x_1\} < \{x_3\} < \{\neg x_2\}$.

# References

Dana Angluin. On the Complexity of Minimum Inference of Regular Sets. *Information and Control*, 39(3):337–350, 1978.

Noureddine Bouhmala. A multilevel learning automata for max-sat. *International Journal of Machine Learning and Cybernetics*, 6(6):911–921, 2015.

François Coste and Goulven Kerbellec. Learning automata on protein sequences. In *JOBIM*, pages 199–210, 2006.

Colin De la Higuera. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, 2010.

Colin de la Higuera. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, 2010.

Henning Fernau, Pinar Heggernes, and Yngve Villanger. A Multi-Parameter Analysis of Hard Problems on Deterministic Finite Automata. *Journal of Computer and System Sciences*, 81(4):747–765, 2015.

E. Mark Gold. Language Identification in the Limit. *Information and Control*, 10(5): 447–474, 1967.

E. Mark Gold. Complexity of Automaton Identification from Given Data. *Information and Control*, 37(3):302–320, 1978.

Alex Groce, Doron A. Peled, and Mihalis Yannakakis. Adaptive model checking. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 8th International Conference, TACAS 2002, Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2002, Proceedings*, volume 2280 of *Lecture Notes in Computer Science*, pages 357–370. Springer, 2002.

Haonan Guo, Shilin Wang, Jianxun Fan, and Shenghong Li. Learning automata based incremental learning method for deep neural networks. *IEEE Access*, 7:41164–41171, 2019.

Mohammad Hasanzadeh-Mofrad and Alireza Rezvanian. Learning automata clustering. *Journal of Computational Science*, 24:379–388, 2018.

Hua Mao, Yingke Chen, Manfred Jaeger, Thomas D Nielsen, Kim G Larsen, and Brian Nielsen. Learning deterministic probabilistic automata from a model checking perspective. *Machine Learning*, 105(2):255–299, 2016.

Franz Mayr and Sergio Yovine. Regular inference on artificial neural networks. In Andreas Holzinger, Peter Kieseberg, A Min Tjoa, and Edgar R. Weippl, editors, *Machine Learning and Knowledge Extraction - Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Proceedings*, volume 11015 of *Lecture Notes in Computer Science*, pages 350–369. Springer, 2018.

Mohammad Reza Meybodi and Hamid Beigy. New learning automata based algorithms for adaptation of backpropagation algorithm parameters. *International Journal of Neural Systems*, 12(01):45–67, 2002.

K Najim, L Pibouleau, and MV Le Lann. Optimization technique based on learning automata. *Journal of Optimization Theory and Applications*, 64(2):331–347, 1990.

Kaddour Najim and Alexander S Poznyak. *Learning Automata: Theory and Applications*. Elsevier, 2014.

Ann Nowé, Katja Verbeeck, and Maarten Peeters. Learning automata as a basis for multi agent reinforcement learning. In Karl Tuyls, Pieter Jan't Hoen, Katja Verbeeck, and Sandip Sen, editors, *Learning and Adaption in Multi-Agent Systems, First International Workshop, LAMAS 2005*, volume 3898 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 2005.

Rajesh Parekh and Vasant Honavar. Learning dfa from simple examples. *Machine Learning*, 44(1):9–35, 2001.

Leonard Pitt. Inductive Inference, DFAs, and Computational Complexity. In Klaus P. Jantke, editor, *Analogical and Inductive Inference, International Workshop AII 1989*, volume 397 of *Lecture Notes in Computer Science*, pages 18–44. Springer, 1989.

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

Peter J Ramadge and W Murray Wonham. Supervisory Control of a Class of Discrete Event Processes. *SIAM journal on control and optimization*, 25(1):206–230, 1987.

Alireza Rezvanian, Ali Mohammad Saghiri, Seyed Mehdi Vahidipour, Mehdi Esnaashari, and Mohammad Reza Meybodi. *Recent Advances in Learning Automata*, volume 754 of *Studies in Computational Intelligence*. Springer, 2018.

Anis Yazidi, Nourredine Bouhmala, and Morten Goodwin. A team of pursuit learning automata for solving deterministic optimization problems. *Applied Intelligence*, 50(9): 2916–2931, 2020.