Deciding minimal distinguishing DFAs is NP-complete

Deciding minimal distinguishing DFAs is NP-complete

Jan Martens
j.j.m.martens@tue.nl

June 7, 2023

Abstract

In this paper, we present a proof of the NP-completeness of computing the smallest Deterministic Finite Automaton (DFA) that distinguishes two given regular languages as DFAs. A distinguishing DFA is an automaton that recognizes a language which is a subset of exactly one of the given languages. We establish the NP-hardness of this decision problem by providing a reduction from the Boolean Satisfiability Problem (SAT) to deciding the existence of a distinguishing automaton of a specific size.

1 Introduction

We consider the problem of automatically explaining the inequivalence of Deterministic Finite Automata (DFAs). In particular, we are interested in short witnesses for the inequivalence. A straightforward approach to explain the inequivalence of two DFAs would be to provide a distinguishing word, i.e. a word that is accepted by one of the automata but not the other.

This method of finding minimal distinguishing words is well understood and decidable in polynomial time [6]. An efficient implementation is given in [8] that has the same runtime complexity as the best known algorithm that decides language equivalence, known as Hopcroft's minimization [3].

In this work we are motivated by smaller witnesses of inequivalence in the form of regular languages. These languages might contain invariants that provide a shorter and more intuitive explanation. For example, consider the DFAs $\mathcal{A}$ and $\mathcal{B}$ shown in Figure 1. The shortest distinguishing word for these DFAs is a^7. Indeed, we confirm $a^7 \in \mathcal{L}(\mathcal{A})$ but $a^7 \notin \mathcal{L}(\mathcal{B})$. A different explanation for the inequivalence of $\mathcal{A}$ and $\mathcal{B}$ could be: every odd length sequence of a's is accepted by $\mathcal{A}$ and not by $\mathcal{B}$.

We call a DFA a *distinguishing* automaton for two DFAs if the language recognized is a subset of exactly one of the two DFAs. In the example from Figure 1, we see that our distinguishing witness with invariant is equivalent to a distinguishing automaton with only two states, i.e. the DFA A_{odd} such

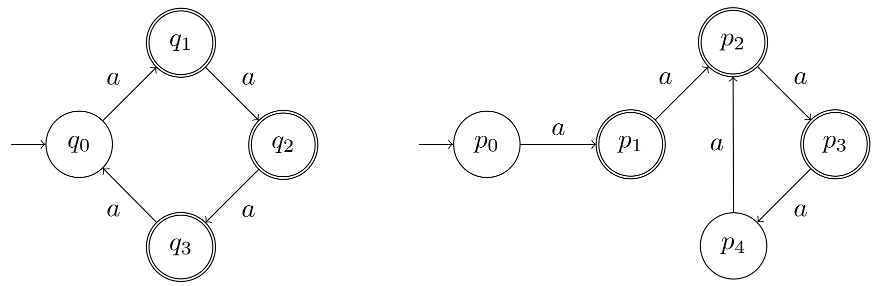

Figure 1: The DFA $\mathcal{A}$ on the left and the DFA $\mathcal{B}$ on the right side.

that $\mathcal{L}(A_{odd}) = \{a^{2i+1} \mid i \in \mathbb{N}\}$. An automaton recognizing only the minimal distinguishing word a^7 would contain at least eight states.

In the setting of model based development it can be key to understand the differences between state based systems. This led us to study the synthesis of *distinguishing* DFAs, and leads naturally to following decision problem.

***k*-DFA-DIST:** Let A_1 and A_2 be DFAs such that $\mathcal{L}(A_1) \neq \mathcal{L}(A_2)$, and $k \in \mathbb{N}$ a number. Decide if there is a DFA A_{dist} with at most k states such that:

$$\mathcal{L}(A_{dist}) \subseteq \mathcal{L}(A_1) \iff \mathcal{L}(A_{dist}) \not\subseteq \mathcal{L}(A_2).$$

The contribution of this work is that we prove the intractability of k-DFA-DIST.

Theorem 1. Deciding k-DFA-DIST is NP-complete.

The reduction from CNF-SAT that proves the NP-completeness is new to our knowledge. We believe this reduction of CNF-SAT formulas to regular languages is an intuitive method of showing DFA problems NP-complete.

There are some decision problems on DFAs that show some similarities, but are different from the work here. For instance the early work of Gold [2] and Pfleeger [7] in which it is shown that learning minimal DFAs from (partial) observations is NP-complete. In the line of this work by Gold, so-called *separating* languages are widely studied in the literature [1, 5]. Here the separating problem is, given languages L_1 and L_2, to find a separating language L_{sep} such that $L_{sep} \subseteq L_1$ and $L_{sep} \cap L_2 = \emptyset$. Although this resembles our distinguishing problem, a direct relation is not trivial.

Another influential work is due to Kozen [4]. This work includes a proof of NP-hardness of deciding whether the intersection of a finite number of DFAs is empty.

2 Notation & Background

For two natural numbers $i, j \in \mathbb{N}$ we write $[i, j] = i, i+1, \ldots, j$ as the closed interval from i to j. Given a finite alphabet Σ, a sequence of elements of Σ

is called a word. We define Σ^i as the set of all words over Σ of length i, and $\Sigma^* = \bigcup_{i\in\mathbb{N}} \Sigma^i$ for all words over Σ. Given words $u, v \in \Sigma^*$, we write $u \cdot v$ and uv for word concatenation. Additionally, given a number $i \in \mathbb{N}$ and a word $u \in \Sigma^*$ we write u^i for the concatenation of i times the word u.

Definition 2. A Deterministic Finite Automata (DFA) $A = (Q, \Sigma, \delta, q_0, F)$ is a five-tuple consisting of:

- Q a finite set of states,
- Σ a finite set of symbols called the alphabet,
- $\delta : Q \times \Sigma \to Q$ the transition function,
- $q_0 \in Q$ the initial state, and
- $F \subseteq Q$ the set of final states.

The transition function δ extends naturally to a transition function for words $\delta^* : Q \times \Sigma^* \to Q$. This is done inductively as follows:

$$\delta^*(q, \varepsilon) = q$$
$$\delta^*(q, aw) = \delta^*(\delta(q, a), w).$$

The language recognized by a DFA $A = (Q, \Sigma, \delta, q_0, F)$, is denoted by $\mathcal{L}(A)$, and consists of all words $w \in \Sigma^*$ such that $\delta^*(q_0, w) \in F$.

The *Myhill-Nerode theorem* is a useful tool to establish the number of states necessary to recognize a language. It is based on the equivalence relation relating words that have the exact same accepting extensions.

Definition 3. Let $x, y \in \Sigma^*$ be words and $L \subseteq \Sigma^*$ a language, then $x \equiv_L y$ if and only if for all $z \in \Sigma^*$ it holds that $xz \in L \iff yz \in L$.

Theorem 4. (Myhill-Nerode [3, Theorem 3.9]) Let $L \subseteq \Sigma^*$ be a language, then L is regular if and only if the relation $\equiv_L$ has a finite number of equivalence classes.

A more specific corollary of the theorem relates the number of equivalence classes of $\equiv_L$ to the smallest number of states a DFA needs in order to recognize L.

Corollary 5. Let L be a regular language over an alphabet Σ, then the smallest DFA A that recognizes L has k states where k is the number of equivalence classes of the relation $\equiv_L$.

3 Reduction

Before we introduce the reduction we define some notation in which we encode truth values of propositions. In the reduction we represent truth assignments as words over the Boolean alphabet $\mathbb{B} = \{\mathtt{0}, \mathtt{1}\}$. Given a set of propositional variables $Prop = \{p_1, \dots, p_k\}$, a truth assignment $\rho : Prop \to \mathbb{B}$ is represented by the word $a_1 \dots a_k \in \mathbb{B}^k$, where $a_i = \rho(p_i)$ for every $i \in [1, k]$. The set $\mathcal{X} = \mathbb{B}^k$ defines all words that represent truth assignments.

Now we are ready to introduce our reduction from CNF-SAT in order to prove Theorem 1. Let $\phi = C_1 \wedge \dots \wedge C_n$ be a CNF formula over the propositional variables $Prop = \{p_1, \dots, p_k\}$, we define two regular languages over the alphabet $\Sigma = \mathbb{B} \cup \{\sharp\}$. The first language $L^-_\phi \subseteq \Sigma^*$ is the finite set of at most n concatenated truth assignments separated by a $\sharp$, i.e.

$$L^-_\phi = \{w_1\sharp \dots w_j\sharp \mid j \in [1, k] \text{ and } w_1, \dots, w_j \in \mathcal{X}\}.$$

The second language $L^+_\phi \subseteq \Sigma^*$ is a superset of L^-_ϕ. In addition to all the word of L^-_ϕ, the language L^+_ϕ contains all words that have as prefix n truth assignments $w_1, \dots, w_n \in \mathcal{X}$ that consecutively satisfy all clauses $C_1, \dots, C_n$, more precisely that is,

$$L^+_\phi = L^-_\phi \cup \{w_1\sharp \cdots w_n\sharp w \mid w \in \Sigma^*, w_i \in \mathcal{X} \text{ and } w_i \text{ satisfies } C_i \text{ for all } i \in [1, n]\}.$$

The languages L^-_ϕ and L^+_ϕ are regular, and hence there are automata that recognize these languages. In particular there are automata recognizing these languages that are polynomial in size. One way of observing this fact is by inspecting the number of Myhill-Nerode equivalence classes of L^+_ϕ and L^-_ϕ.

Lemma 6. Given a CNF formula ϕ, the languages L^+_ϕ and L^-_ϕ are recognizable by an automaton that is polynomial in the size of ϕ.

The next lemma proves the key fact of our reduction. A truth assignment that satisfies a CNF formula ϕ as recurring pattern forms a small distinguishing automaton. Inversely a distinguishing automaton smaller than a certain size necessarily implies the existence of a satisfying truth assignment for ϕ.

Lemma 7. Let $\phi = C_1 \wedge \dots \wedge C_n$ be a CNF formula over k propositional letters $Prop = \{p_1, \dots, p_k\}$. Then ϕ is satisfiable if and only if there is a DFA A_{dist} with at most $k + 2$ states such that $\mathcal{L}(A_{dist}) \subseteq L^+_\phi$ and $\mathcal{L}(A_{dist}) \not\subseteq L^-_\phi$.

Proof. We prove both directions of the implication separately.

(⇒) Assume ϕ is satisfiable, then there is a satisfying truth assignment ρ that is mapped to the word $w_\rho = \rho(p_1) \dots \rho(p_k) \in \mathcal{X}$. We define the language $L_{dist} = \{(w_\rho \cdot \sharp)^i \mid i \in \mathbb{N}\}$, and show that L_{dist} witnesses this implication.

First we show that $L_{dist} \subseteq L^+_\phi$. Assume $i \in \mathbb{N}$, if $i \leqslant n$ then by definition $(w_\rho \cdot \sharp)^i \in L^-_\phi$ and hence also in $(w_\rho \cdot \sharp)^i \in L^+_\phi$. If $i > n$, since ρ is a satisfying assignment, it holds for any $w' \in \Sigma^*$ that $(w_\rho \cdot \sharp)^n w' \in L^+_\phi$, and thus also $(w_\rho \cdot \sharp)^n (w_\rho \cdot \sharp)^{i-n} \in L^+_\phi$. By covering both cases this means $L_{dist} \subseteq L^+_\phi$.

Next, we observe that $(w_\rho \cdot \sharp)^{n+1} \notin L^-_\phi$, and thus $L_{dist} \not\subseteq L^-_\phi$. Hence, since $L_{dist} \subseteq L^+_\phi$ any DFA that recognizes L_{dist} is a distinguishing automaton.

The minimal DFA A_{dist} such that $\mathcal{L}(A_{dist}) = L_{dist}$ contains one loop with $k+1$ states containing all positions of the word $w_\rho \cdot \sharp$ and a sink state to reject all other words. Thus, if ϕ is satisfiable we can construct A_{dist} with $k+2$ states that distinguishes L^+_ϕ and L^-_ϕ, which was to be showed.

($\Leftarrow$) We assume A_{dist} is a DFA with at most $k+2$ states such that for the language accepted $\hat{L} = \mathcal{L}(A_{dist})$ it holds that $\hat{L} \subseteq L^+_\phi$ and $\hat{L} \not\subseteq L^-_\phi$. We show that this means ϕ is satisfiable.

Since $\hat{L} \setminus L^-_\phi \neq \emptyset$ and $\hat{L} \subseteq L^+_\phi$ there is a word $w \in L^+_\phi \setminus L^-_\phi$ accepted by A_{dist}. By definition w is of shape $w = w_1\sharp \dots w_n\sharp w'$ where $w' \in \Sigma^+$ and $w_1, \dots, w_n \in \mathcal{X}$ and for every $i \in [1,k]$ the word w_i represents a satisfying truth assignment for the clause C_i. Next we show that w_1 represents a satisfying truth assignment for ϕ by counting the number of equivalence classes of $\equiv_{\hat{L}}$ for the prefixes of $w_1 \cdot \sharp$, together with the postfix $w_{post} = w_2\sharp \dots w_n\sharp w'$ that witnesses an accepting postfix for $w_1\sharp$.

We define the set U as the set containing all prefixes of $w_1 = a_1 \dots a_k$, i.e.

$$U = \{\varepsilon\} \cup \{a_1 \dots a_j \mid j \in [1,k]\}.$$

If $v, u \in U$ and $v \neq u$ then $v \not\equiv_{\hat{L}} u$, since there is a $\sigma \in \Sigma^*$ such that $v\sigma = w$ and $w \in \hat{L}$ and $u\sigma \notin \hat{L}$. This means there are $|U| = k+1$ distinct classes of $\equiv_{\hat{L}}$. Lastly, since $\sharp z \notin \hat{L}$ for any $z \in \Sigma^*$ we can also conclude that $\sharp \not\equiv_{\hat{L}} u$ for all $u \in U$.

Since we assumed that A_{dist} has at most $k+2$ states, by Corollary 5 there are at most $k+2$ equivalence classes of $\equiv_{\hat{L}}$. Since trivially $w_1\sharp \not\equiv_{\hat{L}} \sharp$, by the pigeonhole principle there is a prefix $u \in U$ such that at $w_1\sharp \equiv_{\hat{L}} u$.

It can not be the case that $u = a_1 \dots a_i$ for some $i \in [1,k]$, since

$$a_1 \dots a_i \cdot a_{i+1} \dots a_k\sharp w_{post} \in \hat{L}$$
$$w_1\sharp \cdot a_{i+1} \dots a_k\sharp w_{post} \notin \hat{L}.$$

By eliminating all alternatives we conclude $u = \varepsilon$. Using this equivalence and since $\varepsilon \cdot w_1\sharp w_{post} \in L_{dist}$ we derive that $w_1\sharp \cdot w_1\sharp w_{post} \in L_{dist}$. In particular, this means that $(w_1\sharp)^n \cdot w_{post} \in L_{dist}$. By definition of L^+_ϕ this means that the truth assignment w_1 satisfies all clauses $C_1, \dots, C_n$ and hence it is a satisfying assignment for ϕ. This witnesses that ϕ is a satisfying formula. □

This lemma allows us to prove Theorem 1.

Proof of Theorem 1. Membership of NP follows naturally. For two DFAs A_1 and A_2 we can, in polynomial time, check if $\mathcal{L}(A_1) \subseteq \mathcal{L}(A_2)$. This can be done by computing the emptiness of $\mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$. Moreover, either A_1 or A_2 itself necessarily already is a distinguishing automaton, so the minimal distinguishing DFA is definitely polynomial in size.

NP-hardness is a direct consequence of Lemma 7 and of the fact that $L^-_\phi \subseteq L^+_\phi$, so the language of any distinguishing automaton is a subset of L^+_ϕ and not vice-versa. □

Acknowledgements: The author thanks Tim Willemse for raising the question of distinguishing transition-systems with invariants. Thanks also to Jan Friso Groote and Anna Stramaglia for providing helpful suggestions on this document.

References

[1] Erik D Demaine, Sarah Eisenstat, Jeffrey Shallit, and David A Wilson. Remarks on separating words. In *Descriptional Complexity of Formal Systems: 13th International Workshop, DCFS 2011, Gießen/Limburg, Germany, July 25-27, 2011. Proceedings 13*, pages 147–157. Springer, 2011.

[2] E Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

[3] John Hopcroft. An n log n algorithm for minimizing states in a finite automaton. In Zvi Kohavi and Azaria Paz, editors, *Theory of Machines and Computations*, pages 189–196. Academic Press, 1971.

[4] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 254–266. IEEE, 1977.

[5] Orna Kupferman, Nir Lavee, and Salomon Sickert. Certifying dfa bounds for recognition and separation. *Innovations in Systems and Software Engineering*, 18(3):405–416, 2022.

[6] Tyler Moore. Gedanken-experiments on sequential machines. In C. E. Shannon and J. McCarthy, editors, *Automata Studies, Annals of Mathematical Studies, no. 34*. Citeseer, 1956.

[7] Charles P Pfleeger. State reduction in incompletely specified finite-state machines. *IEEE Transactions on Computers*, 100(12):1099–1102, 1973.

[8] Rick Smetsers, Joshua Moerman, and David N Jansen. Minimal separating sequences for all pairs of states. In Adrian-Horia Dediu, Jan Janoušek, Carlos Martín-Vide, and Bianca Truthe, editors, *Language and Automata Theory and Applications (LATA 2016)*, pages 181–193. Springer, 2016.