A theoretical analysis of optimal and heuristic methods for DFA learning

Bachelor's Degree Thesis

Horia Radu

Supervisors: Sicco Verwer, Simon Dieck

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Horia Radu
Final project course: CSE3000 Research Project
Thesis committee: Sicco Verwer, Simon Dieck, Soham Chakraborty

Abstract. Deterministic finite automata (DFA) are interpretable models used for classification and prediction tasks based on sequence data. They often act as surrogate models for software systems. Plenty of methods exist for the purpose of DFA learning. Examples include optimal algorithms such as SAT-based encoding and various heuristic methods as the likes of the BlueFringe framework of the EDSM algorithm. By definition, optimal algorithms can guarantee a minimal DFA consistent with the training data, but this does not exclude the possibility of heuristics also finding the optimal solution. However, it is generally believed that optimal methods could always require strictly less data to learn such a minimal model than their counterparts. In our research, we provide mathematical proofs and counter-examples that show the above statement to be false. We further demonstrate that, unless formally defined, there exist numerous languages and settings where heuristics outperform optimal methods on data efficiency benchmarks. Finally, we prove that optimal methods are equal to the BlueFringe framework in terms of optimistic learning efficiency.

Keywords: DFA learning · optimal methods · heuristics · theoretical analysis

1 Introduction

Irish Franciscan philosopher John Punch, in his 1639 commentary on the works of Duns Scotus, said: "Entities are not to be multiplied without necessity" (Non sunt multiplicanda entia sine necessitate). Commonly known as Occam's Razor, this principle advocates for the simplest explanation for phenomena.

We find a natural application of this concept in the field of automata learning, where the objective is to infer a deterministic finite automaton (DFA) that precisely captures a target language from sequence data. Often, they act as surrogate models for systems. They are trained to closely replicate a system's behaviour for an interpretable analysis in various fields such as computational linguistics, bioinformatics, speech processing, and verification [7].

Normally, this learning process relies on heuristics to construct a DFA consistent with an input dataset. Such methods include evidence-driven state merging (EDSM) [4] and Alergia [2], which iteratively merge the states of the DFA to arrive at simpler models. In contrast, exact methods for identifying models of minimal size exist, for instance, using SAT solvers [7] or iterative oracle querying [1].

The pursuit of a minimal DFA aligns with the previously stated Occam's razor, ensuring interpretability. This raises an important question: beyond ease of understandability, could exact methods also be more data-efficient? This research investigates the hypothesis that minimal DFAs learned through exact methods require less training data than their heuristic counterparts, potentially improving performance in terms of learning efficiency. Prior work on characteristic samples [3] has shown that there exists a certain minimal sample of words sufficient for both methods to learn a DFA consistent with a target language. Furthermore, previous research also includes an efficient algorithm for computing characteristic samples alongside an upper bound for the size [6] [1]. However, it remains unclear whether this sample is always smaller when exact algorithms are used. Thus, the core research question of this paper is:

Do optimal methods for DFA learning require less data than heuristics to produce correct minimal models?

Our goal is to arrive at a fully theoretical proof that holds for all cases. Therefore, we are looking to mathematically show that either optimal methods require less data every time or find a

counter-example for this statement. Previous work by Nerode et al. [5] has provided the necessary theoretical framework to attempt such proofs in the form of the Myhill-Nerode theorem. We split our main question into the following sub-questions:

1. Do optimal methods always require less data than a heuristic?
2. Do heuristics ever require less data than optimal methods?
3. Do optimal methods ever require less data than a heuristic?
4. Do optimal methods and heuristics always require the same amount of data?

Our contribution can be summarised as follows:

- A counter-example that shows how heuristics can perform as well as optimal methods (Section 3.1);
- A counter-example that urges us to strictly define our heuristic, as otherwise optimal methods could almost always be outperformed (Section 3.1);
- A proof that state-merging heuristics could always get lucky and outperform optimal methods (Section 3.2);
- Proof that shows that we can always build a dataset of the same size as the minimal characteristic sample of the optimal method, such that BlueFringe can provide the same result. Otherwise stated as the characteristic sample is method independent. (Section 3.3).

The rest of the paper is divided as follows: In section 2 we give the formal definition of the problem and the used notations. Then, we answer our research questions in section 3, providing the necessary proofs. Next, in section 4, we discuss the ethical considerations of our research. We conclude this paper in section 5 by discussing the implications of our research within the broader context of the field and recommend future research.

2 Preliminaries

This section serves the purpose of formally defining our problem. We begin by explaining our most used terms, followed by giving the mathematical form of our research questions in section 2.1. Next in this section we explain two extra concepts that will be used later in the paper. Most definitions and notations are the same as those introduced by Verwer et al. [7]. We finalise the preliminaries in section 2.2, where we briefly introduce the Myhill-Nerode theorem alongside some basic group theory notions that will be used in the later proofs. We assume the reader to be familiar with the theory of languages and automata.

2.1 Definitions

We start by defining frequently used notations and formally defining our goals. A deterministic finite automaton (DFA) is a finite-state machine made up of states and labelled transition edges. It accepts a word if there exists a sequence of labels along a path from the designated start state to a

final (accepting) state, matching the input. This makes DFAs suitable for recognising any regular language [7]. We make use of the following notations:

- $L(\mathcal{A})$ the language of DFA $\mathcal{A}$;
- $S = S_+ \cup S_-$ the finite input sample for DFA learning, where S_+ are the positive strings and S_- the negative ones;
- $\mathcal{S}_o$ the set of all characteristic samples for the optimal method, and analogously. $\mathcal{S}_h$ for the heuristic. The characteristic sample is condensed in section 1, but further explained by Gold et al. [3];
- $\mathcal{S}_o^{min} = \arg\min_{\mathcal{S} \in \mathcal{S}_o} |\mathcal{S}|$ the minimal (smallest in number of words) characteristic sample for the optimal method, with the heuristic version defined respectively;
- REG_Σ the set of all regular languages over an alphabet Σ;
- $|\mathcal{A}|$ the size of a DFA $\mathcal{A}$, where we measure the number of states it contains;
- ϵ the empty string.

With these notations in mind, we present the following formal definition of our research questions. The aim of this paper is to mathematically (dis)prove the following statements:

$$\forall \Sigma : \forall L \in REG_\Sigma : |\mathcal{S}_o^{min}| < |\mathcal{S}_h^{min}| \tag{1}$$

Do optimal methods always require less data than a heuristic?

$$\exists \Sigma : \exists L \in REG_\Sigma : |\mathcal{S}_o^{min}| > |\mathcal{S}_h^{min}| \tag{2}$$

Do heuristics ever require less data than optimal methods?

$$\exists \Sigma : \exists L \in REG_\Sigma : |\mathcal{S}_o^{min}| < |\mathcal{S}_h^{min}| \tag{3}$$

Do optimal methods ever require less data than a heuristic?

$$\forall \Sigma : \forall L \in REG_\Sigma : |\mathcal{S}_o^{min}| = |\mathcal{S}_h^{min}| \tag{4}$$

Do optimal methods and heuristics always require the same amount of data?

Extra

For later use, we also need to introduce the concepts of DFA characteristic sample and state-merging heuristic algorithm, along with an example of such a method.

The ultimate goal of DFA identification is to construct the smallest DFA $\mathcal{A}$ consistent with S such that $S_+ \subseteq L(\mathcal{A})$ and $S_- \subseteq \Sigma^* \setminus L(\mathcal{A})$. In this context, the DFA characteristic sample of DFA $\mathcal{B}$ is a dataset $\mathcal{S}$ such that the minimal consistent DFA with $\mathcal{S}$ is DFA $\mathcal{B}$. This is different from the original definition of the characteristic sample given by Gold et al. [3] because it concerns the DFA and not the regular language recognised.

State-merging heuristics begin by building a tree-shaped DFA from the sample, known as an Augmented Prefix Tree Acceptor (APTA), see figure 1 by Verwer et al. [7] for an example. An APTA ensures that two strings reach the same state if and only if they share a common prefix

leading to that state, hence the term "prefix tree." The APTA is "augmented" because it includes unlabeled states if no input string ends there. Afterwards, the algorithm starts merging the states of the APTA, minimising the size of the model while preserving its accuracy. Merging two states q_0 and q_1 creates a new state q_2 which inherits all incoming and outgoing transitions of both original states. This merge is allowed only if q_0 and q_1 are consistent, i.e. they both have the same label [7].

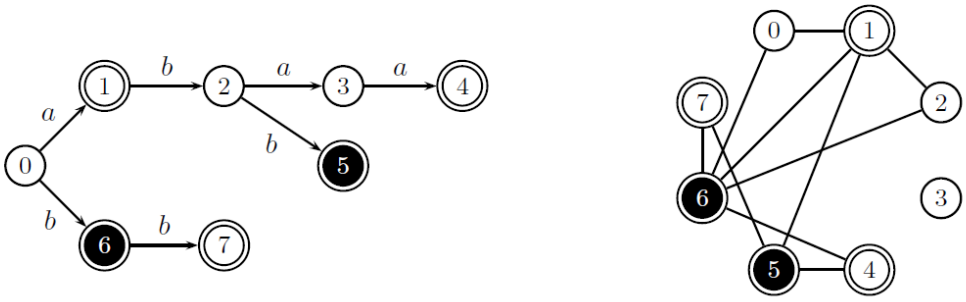

Fig. 1: An augmented prefix tree acceptor for $S = S_+ = \{a, abaa, bb\} \cup S_- = \{abb, b\}$ (left) and the corresponding consistency graph (right). Some vertices in the consistency graph are not directly inconsistent, but inconsistent due to determinization. For instance, state 2 and 6 are inconsistent because the strings abb and bb will end in the same state if these states are merged. Also state 1 and 2 are inconsistent because the strings a and abb will end in the same state if these states are merged [7].

The most successful state-merging heuristic algorithm for DFA identification to date is Evidence-Driven State Merging (EDSM) [7]. EDSM is a greedy algorithm that chooses which merges to perform based on a heuristic guided by evidence from the input data. Since it considers $|V|_2$ merge possibilities at each iteration and V can be large, this is a computationally expensive method [7]. Furthermore, EDSM can guarantee the smallest DFA only if its best-first search has explored all smaller solutions [7]. To address this, EDSM is typically used with the BlueFringe framework, see figure 2 by Verwer et al. [7], which reduces the number of merge candidates without sacrificing accuracy on the training dataset, thus greatly improving the performance [4]. This framework keeps the following at each step of its execution:

- a set of red states (core states already in the final DFA);
- a blue fringe (candidate states for merging coloured blue);
- white states (unexplored).

Merges are only performed between red and blue states. If a blue state cannot be merged with any red state, i.e. no consistent merge is possible, it is promoted to red and all of its children are coloured blue. Within this framework, EDSM is a polynomial-time greedy algorithm that quickly converges to a **local** optimum [7].

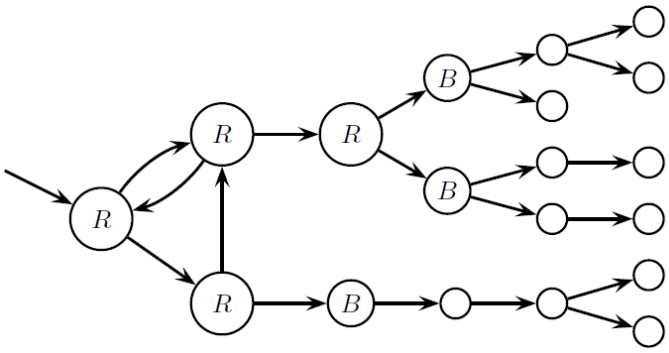

Fig. 2: The BlueFringe framework. The red states (labelled R) are the identified parts of the automaton. The blue states (labelled B) are the current candidates for merging. The uncolored states are pieces of the APTA [7].

2.2 Myhill-Nerode Theorem

We end the preliminaries with a brief introduction to group theory, specific to the Myhill-Nerode theorem, alongside some notations that will be used in some of the latter proofs. The theorem states that given a regular language L over an alphabet Σ and two words $w_1, w_2 \in L$, then a word $w_3 \in \Sigma^*$ s.t. $w_1 w_3 \in L \iff w_2 w_3 \notin L$ is called a distinguishing extension [5]. If no such w_3 exists for the pair of w_1, w_2, then the two words are called indistinguishable and are part of the same equivalence class. All such classes together form a free monoid under Σ^* [5]. A monoid is an abstract group of items closed and associative over an operation that has a neutral element. In our case, we restrict this definition to the free monoid, defining the operation as string concatenation where the empty string is the neutral element. Closure and associativity are trivial to prove. As such, we define the following notations:

- $a \equiv_L b$ the equivalence of string a and b over the regular language L and alphabet Σ. Mathmatically this can be also stated as for all $r \in \Sigma^*$ we have $ar \in L \iff br \in L$
- $[a]_L = \{w | w \in \Sigma^* \wedge w \equiv_L a\}$ the equivalence class of string a over the language L and alphabet Σ consisting of all strings indistinguishable from a.
- $[a]_L \cdot [b]_L = [a \cdot b]_L$ the closure property of equivalence classes under concatenation [5].

3 Answers and Proofs

The goal of this chapter is to provide answers to the stated research questions along with the necessary mathematical proofs. We begin in section 3.1 by presenting the counter-examples we found that deny the first research question by proving the second. Following the negation of our initial hypothesis, we explore the newly arisen challenges in section 3.2 and refine the scope of our research. Finally, we answer our last questions in section 3.3.

3.1 Counter-Examples

This subsection contains two counter-examples that contradict our initial hypothesis, that optimal methods always require strictly less data than heuristics. We first offer an example where a heuristic

does as well as the optimal method, followed by showing that heuristics could always outperform optimal methods.

Heuristic Performs as well as Optimal Method The first example that we present is one in which heuristic methods require the same amount of data as optimal methods. This means that there exists at least one regular language for which the latter does not need strictly less data to learn than the former, negating the statement in equation 1.

Lemma 1. *If all input data points given to an optimal DFA learning method are accepting or rejecting, then the output DFA is a single-state DFA.*

Proof. We know by definition that the optimal method finds the minimal DFA consistent with the given input. Moreover, a DFA cannot have fewer than one state, as it would not exist anymore.

Consider a single-state DFA that accepts every word in the extended alphabet of its language, see figure 3 for an example. Since it accepts every word in the extended alphabet, that also includes all the words given as input, that are labelled as accepting strings in the hypothesis.

Since the considered DFA is both consistent with our input data and also minimal, as there could be no smaller DFA in existence, then we have shown that an optimal method would output a single-state DFA given any all accepting input dataset. This proof holds for an all rejecting input dataset scenario as well, with its respective changes.

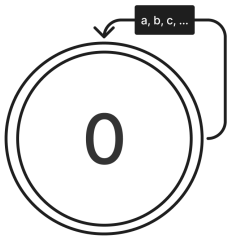

Fig. 3: A single-state DFA that accepts every word in its extended alphabet Σ^*. State 0 is both a start and an accepting state. We define a self-loop for every character of the alphabet Σ, making it impossible to arrive at a rejecting state.

Corollary 1. *At least two differently labelled input data points are needed when learning any regular language $L \neq \Sigma^*$ and $L \neq \emptyset$ with optimal methods.*

Proof. Following from lemma 1, if all input data points have the same label, the result when running the optimal algorithm will be a single-state DFA. Further, our language needs to be consistent with at least one positive and one negative word. Since both cannot end in the same state, but the outputted model is consistent with both, the model must have at least two states. It results that at least two distinctly labelled input data points are needed for the task of learning any such regular language.

This counter-example revolves around the language $L = \{a\}$ for all Σ where $a \in \Sigma$, a trivial finite language that only accepts the single string a and rejects everything else. The minimal consistent DFA with this language is a two-state DFA, see figure 4. We assume that undefined transitions lead to a rejecting sink state and that we use EDSM as our heuristic algorithm. Under these premises, both the optimal and heuristic methods only require two input data points to output the DFA in figure 4. The previous statement follows from corollary 1 which says that at least two data points are needed to learn this language using optimal methods and we can easily verify that the sample $S_+ = \{a\} \cup S_- = \{\epsilon\}$ is enough for both methods. Now, we have disproven equation 1, as we have shown that there exists a Σ such that there exists a language $L \in REG_\Sigma$ for which $|\mathcal{S}_o^{min}| \geq |\mathcal{S}_h^{min}|$ holds. Hence, the following equation is a possible reformulation:

$$\forall \Sigma : \forall L \in REG_\Sigma : |\mathcal{S}_o^{min}| \leq |\mathcal{S}_h^{min}| \tag{5}$$

Do heuristics always require at least as much data as an optimal method?

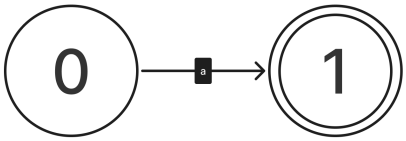

Fig. 4: A minimal consistent DFA for the language $L = \{a\}$. State 0 is the initial state and non-accepting, whereas state 1 is the accepting state. Moving from state 0 to state 1 is possible through reading character a. All missing transitions lead to a rejecting sink state.

Heuristic Outperforms Optimal Method The second counter-example that we provide is one that disproves the recently provided equation 5 that replaces the strict inequality with its more relaxed version. We show how not having a formal definition of heuristic methods can allow the creation of a heuristic that can outperform optimal methods on specific tasks.

For this counter-example, we will consider the following regular language $L = (abc)^+$ for all Σ where $\{a, b, c\} \subseteq \Sigma$, in which we accept any word that is composed of repeating sequences of the string abc, empty string excluded. Moreover, consider the following heuristic algorithm:

1. Start with the minimal consistent DFA for the language $L = (abc)^+$
2. Look at the next input data point; if none are left, halt and output the DFA from step 1. If the data point is consistent with the language L, go to step 2; else move to step 3.
3. Run EDSM from the beginning of the input and output its result.

This algorithm is capable of finding a DFA consistent with any input dataset, as it reverts to running EDSM in the worst case, which has been proven to accomplish this [4]. If our task is to learn the regular language $L = (abc)^+$, then the heuristic would require fewer data to do so when compared to the optimal method. Consider what happens when we give a single input data point to both methods, specifically the accepting word abc. Since it is consistent with the target language, the heuristic will output the minimal DFA for L, our desired result. Further, from corollary 1 and the fact that

($L \neq \Sigma^*$ and $L \neq \emptyset$ and $L(\mathcal{A}) = L$) implies $|\mathcal{A}| \geq 2$, it results that we need at least two input data points to learn our language through the optimal method. Hence, this specific heuristic requires less data to learn language L.

Since we had no restrictions on the definition of a heuristic method, this is a valid counter-example that disproves equation 5, as we have shown equation 2 to be true.

3.2 Problem Refinement

We use this subsection to show how the previous counter-examples pose new challenges to our research and urge us to limit its scope. We begin by exploring the issues with properly defining the heuristic. Moreover, we propose a new refined goal of our research to circumvent the previous issues by redefining the characteristic sample.

Heuristic Definition We will attempt to formally define what a heuristic method means, showcasing the difficulties of this task along the way. As shown in the second counter-example in section 3.1, having no such thing can lead to the creation of any arbitrary heuristic to serve a specific goal.

Lemma 2. *Plain EDSM can always perform at least as well as optimal methods in terms of data efficiency for learning a particular language L in an optimistic scenario, formally, the negation of equation 3.*

Proof. This follows from the definition of the SAT-based algorithm in Verwer et al. [7], where the basic idea is to find the minimal amount of colours needed to solve the graph colouring embedding of the DFA learning problem, where this number will represent the size of the optimal output. As stated by Nerode et al. [5], the size of the minimal DFA consistent with a language L is equal to the number of equivalence classes, as defined in section 2.2 of the language L. The SAT-based algorithm will have a node for every state in the APTA, and colouring two nodes the same colour can be seen as "merging" the two states. Formally, that is simply assigning an already existing equivalence class given the characteristics of language L. Additionally, we know that EDSM merges the states of the APTA in a heuristic fashion.

We can say that the APTA states considered for merging are the nodes $N = n_1, n_2, ..n_k$ for an input sample S where $k \geq |S|$ since we build the augmented tree. The optimal "merges" that the SAT algorithm does can be therefore noted as the set of pairs $M_{OPT} = (n_{1i}, n_{1j}), (n_{2i}, n_{2j}), ..(n_{ki}, n_{kj})$. A pair (n_{mi}, n_{mj}) means colouring the m_{th} APTA state, given any arbitrary order, in the manner dictated by the optimal solution, will assign the strings ending in states n_i and n_j in the same equivalence class. We emphasise the fact that these are not actual merges happening iteratively when running the optimal method, but a mere analogy. Additionally, define $M_H = \{$chain $= \{(n_{1i}, n_{1j}), (n_{2i}, n_{2j}), ..(n_{pi}, n_{pj}) | 1 \leq i, j \leq k$ and $1 \leq p \leq k-1\}|$ chain is a valid order of merges given $S\}$ as all the valid possible chain of merges that the EDSM algorithm can follow by considering all possiblities at every iteration. The meaning of a chain being a valid order of merges given S is that there exists a language L consistent with S for which all merges are only done between APTA states representing strings in the same equivalence class under L.

However, not all elements of M_H are viable for EDSM considering an arbitrary dataset, as nodes are chosen for merging based on the amount of evidence in the input, i.e. the number of times a word appears. Using minimal characteristic samples ensures that each input string is unique, but the distance to the root of the APTA is used as a tie-breaker. Thus, we can order M_{OPT} on this

metric and ensure that each equivalent "merge" is considered by EDSM. Moreover, because M_{OPT} is a valid chain of merges on the APTA consistent with S and therefore $M_{OPT} \in M_H$, then in an optimistic scenario, EDSM will follow an order of merges leading to an optimal solution. Since all optimal methods provide the same solution through the definition of optimality, referring to any such algorithm was sufficient to prove our statement.

When referring to heuristic methods, for the rest of this paper, we will consider EDSM, as briefly described in section 2.1. Yet, for the basic implementation of EDSM as described in Lang et al. [4], we can easily prove that it will always be at least as good as the optimal method for data efficiency in optimistic cases and disprove our third research question, see lemma 2. To avoid this trivial scenario, we further restrict our definition of a heuristic by only considering the BlueFringe framework of the EDSM algorithm, summarised in section 2.1. This limits the possibility of node merges at any point and introduces the likelihood of not being able to follow the same optimal merges.

Proposition 1. *If, when running the same dataset on both an optimal method and a state-merging heuristic, we arrive at two differently sized outputs, then the DFA resulting from the heuristic is bigger.*

Proof. Consider any input dataset S, for which we define DFA $\mathcal{A}$ as the output of the optimal method on S, and DFA $\mathcal{B}$ as the output of the state-merging heuristic. By hypothesis $|\mathcal{A}| \neq |\mathcal{B}|$. For the sake of contradiction that $|\mathcal{B}| < |\mathcal{A}|$. But we know that $\mathcal{B}$ is consistent with S, and that the optimal algorithm is guaranteed to output the minimal DFA consistent with the input, and so $|\mathcal{A}| \leq |\mathcal{B}|$. We arrive at a contradiction and prove our initial proposition.

Proposition 2. *The size of the resulting DFA from an optimal method is monotonic with the changes in the input.*

Proof. By definition of the optimal method, we know that if we have a dataset S which results in a DFA $\mathcal{A}$ upon solving, then DFA A is consistent with any subset of S. Therefore, the minimal consistent DFA of any subset of S cannot be larger than A. The other side of the inequality is proved conversely.

Lemma 3. *State-merging heuristics could outperform optimal methods in data efficiency by making non-optimal merges in multitudinous cases, formally stated as for all Σ there exist $L_1, L_2, ...L_k \in REG_\Sigma$ where $k >> 1$ and $L_1 \neq L_2 \neq ... \neq L_k$ s.t. the statement $|\mathcal{S}_o^{min}| > |\mathcal{S}_h^{min}|$ holds.*

Proof. Consider any input dataset S, for which we define DFA $\mathcal{A}$ as the output of the optimal method on S, and DFA $\mathcal{B}$ as the output of the state-merging heuristic. Further suppose $|\mathcal{A}| \neq |\mathcal{B}|$. From proposition 1 it follows that $|\mathcal{B}| > |\mathcal{A}|$. Further, by definition of both algorithms, we know that the two DFAs are the minimal consistent DFAs with the language that they recognise and therefore $L(\mathcal{A}) \neq L(\mathcal{B})$. Make a final assumptions that $\mathcal{S}_o^{min} = S$ for $L(\mathcal{A})$. Now, proposition 2 holds for this scenario because DFA B is more complex than DFA A and is also consistent with S, which is the minimal dataset that preserves all complexities of DFA A.

From the fact that $\mathcal{B}$ is consistent with S, proposition 2 and that $|\mathcal{B}| > |\mathcal{A}|$ it also follows that we cannot learn DFA $\mathcal{B}$ and implicitly $L(\mathcal{B})$ optimally without increasing the size of our input dataset. Hence, the state-merging heuristic performed better than the optimal algorithm for the task of learning $L(\mathcal{B})$ by making non-optimal merges. It is trivial to see how this applies to many languages, as it often happens that heuristics provide larger outputs than their counterparts [7].

Finally, we need not restrict our definition of EDSM to the BlueFringe framework to observe the following problem, as it applies to any state-merging heuristic method. It is possible that such an algorithm can outperform the optimal counterpart in many cases from a specific perspective, see lemma 3. As such, we have shown multiple challenges in defining a heuristic for the scope of our research, as all explored ones resulted in scenarios where optimal methods lagged in performance.

Characteristic Sample Redefinition Next, we will change our characteristic sample definition to avoid the previously described problem in lemma 3. The definition that will be referred to for the rest of this paper is the DFA characteristic sample introduced in section 2.1.

By using this definition, we restrict the scope of our research and need not worry anymore about heuristics outperforming optimal algorithms. This is because if there exists a smaller characteristic sample such that the heuristic finds a minimal consistent DFA, then this is also a valid input-output for the optimal method. This is formally stated as equation 5, which holds for this definition.

Thus, the final question that we aim to answer within the restricted scope presented in this subsection is equation 4. Or in natural language, can we build a dataset, for every regular language and any alphabet, of the same size as the minimal characteristic sample of the optimal method such that when solving it using the BlueFringe framework of the EDSM algorithm, it is possible to only make optimal merges between states of the APTA?

3.3 Final Answer

Finally, we answer our last question by showing that BlueFringe cannot be blocked from arriving at the optimal solution.

Lemma 4. *If two nodes n_1 and n_2 are coloured red and white respectively, at iteration i during the execution of BlueFringe on dataset S, then for any subset $Sub \subset S$ such that $n_1, n_2 \in N$ as per the definition in lemma 2, the two nodes will be coloured identically after the same merges still valid from the $i-1$ previous iterations of the execution on S.*

Proof. We use the definition of BlueFringe from section 2.1 and the notation of nodes from lemma 2. The premise of this lemma implies that n_2 is coloured grey and therefore too far down the search tree to be considered for merging with n_1. Consider removing data points from S such that the two nodes are still present in the APTA. The definition of the APTA forces us to keep all the intermediary nodes from the root to any of our states, even if removed from the dataset. Therefore, by following the same previous merge, when still valid, the relation of colours in the APTA would be the same between the nodes n_1 and n_2 as all intermediary nodes persisted. This can be seen in figures 5a and 5b, where even though the word represented by state 2 is missing from figure 5b, states 1 and 3 keep their colours within the framework.

Lemma 5. *There exists a minimal characteristic sample for the optimal method for which we cannot build a sample of the same size such that BlueFringe will find the same solutions, although it can find other optimal DFAs.*

Proof. Suppose that we have an $\mathcal{S}_{min\_o}$ for DFA $\mathcal{A}$ where for all $w \in S_+$ it holds that $|w| < 2 \iff |w| \leq 3$. Further, a word w_2 with $|w_2| = 2$ and $w_2 \in S_-$ exists. By the definition of the APTA, there must be a node for the word w_3 with $|w_3| = 3$ as well, for which the

(a) Figure of an APTA where states 1,2,3 are all accepting and colored red, blue and white respectively. All three states represent an accepted word in the input S. Transitions are irrelevant.

(b) Figure of an APTA where states 1,2,3 are all colored red, blue and white, respectively. States 1 and 3 represent an accepted word in the input Sub, while state 2 is unlabelled.

final label is unknown. We also suppose that the transition from the state w_2 to w_3 is labelled by a character that is not used for any other transition leading to a non-sink state. A final assumption has to be made: $w_4 \in S_+, |w_4| = 4, [w_3] \equiv_{L(\mathcal{A})} [w_4]_L$, and therefore the minimal consistent DFA with S will accept w_3.

Given the way in which we constructed our $\mathcal{S}_{min\_o}$ and all previous assumptions, we cannot guarantee that the output of the BlueFringe framework on this input, noted DFA $\mathcal{B}$, will recognise a language that also accepts w_3. That is because the state of the APTA representing w_3 is not labelled, see figure 6 for an example of such. Furthermore, the state for w_2 is rejecting, so given the greedy nature of the algorithm, it is even more likely that across all regular languages that fit our assumptions, w_3 will not be accepted by DFA $\mathcal{B}$.

We continue this proof by showing that we cannot build a dataset of the same size such that BlueFringe can find the optimal solution. Lemma 4 shows that removing any data points from our input cannot possibly validate optimal merges in earlier iterations. Furthermore, we cannot remove the unlabelled state from the APTA as the character that represents the transition from the states representing w_2 and w_3 is unique for accepting words in that edge and can be essential for $L(\mathcal{A})$. Hence, there exist minimal characteristic samples for learning a language with optimal methods, such that no dataset of the same size can be built for BlueFringe to reach the same solution.

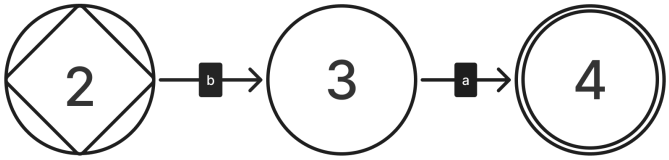

Fig. 6: A section of the APTA from executing BlueFringe on the input S as defined in lemma 5. All missing transitions of the visible states lead to a rejecting sink. State 2 is rejecting, and reading b will transition to state 3, which is currently unlabelled. From there, it is possible to move to the accepting state 4 with an a. States 2,3 and 4 are colored red, blue and white and represent words w_2, w_3 and w_4, respectively.

Theorem 1. *The characteristic sample is method-independent. Otherwise stated as for every characteristic sample for the optimal method, we can build an equally large sample for BlueFringe such that the output is optimal, and formally as equation 4.*

Proof. Given a minimal characteristic sample for the optimal method in the alphabet Σ, $\mathcal{S}_{min\_o}$ along with its output DFA $\mathcal{A}$ with $\mathcal{S}_{min\_h}$ and DFA $\mathbf{B}$ defined respectively for BlueFringe, we assume $\mathcal{A} \neq \mathbf{B}$ and that B is consistent with $\mathcal{S}_{min\_o}$. We will show that $|\mathcal{A}| < |\mathcal{B}|$ implies $|\mathcal{S}_o^{min}| < |\mathcal{S}_h^{min}|$.

For the sake of contradiction, we will assume that $|\mathcal{A}| < |\mathcal{B}|$ and $|\mathcal{S}_o^{min}| = |\mathcal{S}_h^{min}|$. We do not care about the case of $|\mathcal{S}_o^{min}| > |\mathcal{S}_h^{min}|$ as shown in section 3.2. As per our initial statement, B is consistent with $\mathcal{S}_{min\_o}$, which means that the APTA for the heuristic characteristic sample is also consistent with $\mathcal{S}_{min\_o}$. The fact that $|\mathcal{A}| < |\mathcal{B}|$ means that when executing BlueFringe on $\mathcal{S}_{min\_h}$, a merge was made in the APTA that prevented at least two other merges from being viable. This follows from that there exists at least one sample equal in size to the optimal characteristic sample such that the heuristic could have arrived at a DFA equal in size to A, namely $\mathcal{S}_{min\_o}$ itself. From now on, we refer to this input as $\mathcal{S}$. This can be mathmatically written as there exist words $w_1, w_2, w_3, w_4 \in \Sigma*$ such that $w_1 \equiv_{L(\mathcal{B})} w_2$, $w_1 \not\equiv_{L(\mathcal{B})} w_3$, $w_2 \not\equiv_{L(\mathcal{B})} w_4$ and $w_1 \equiv_{L(\mathcal{A})} w_3$, $w_2 \equiv_{L(\mathcal{A})} w_4$, $w_1 \not\equiv_{L(\mathcal{A})} w_2$, but also $w_3 \not\equiv_{L(\mathcal{B})} w_4$. If $w_3 \equiv_{L(\mathcal{B})} w_4$, then the two states could be merged, and thus arrive at the size of DFA $\mathcal{A}$.

Now we will show that the above mathematical formulation is impossible. Because both $\mathcal{A}$ and $\mathcal{B}$ are consistent with $\mathcal{S}$, we know that the following merges are possible between states representing the words:

1. w_1 and w_2 from $w_1 \equiv_{L(\mathcal{B})} w_2$
2. w_2 and w_4 from $w_2 \equiv_{L(\mathcal{A})} w_4$
3. w_1 and w_3 from $w_1 \equiv_{L(\mathcal{A})} w_3$

From here, the problem splits into two cases: either all states representing words w_1, w_2, w_3, w_4 are already labelled in the APTA, or there are some non-terminal nodes. For the former, it is now proven that merging the states of w_3 and w_4 is also possible because all words must be part of the same class for the above merges to be possible, resulting in a contradiction. For the latter, we must refer to lemma 5, where it is stated that merges with unlabelled states can result in a different language than the exact one given by the optimal method due to the greedy nature of the framework. In this case, being that merge 1. is assumed to be executed first and blocking everything else, we can arbitrarily choose w_1 as being labelled positive and w_2 as unlabelled at first, but then being assigned the same class. From the possibilities of merges 2. and 3. we can either assume an already accepting label to w_3 and w_4 or that the greedy algorithm will assign the labels when considering those merges. If they were to be assigned different labels due to merging with another unconsidered state, then a merge would have still happened in the APTA, and the size of the final DFA would have been reduced either way. The only way for any of these two splits of the problem to hold for all assumptions is by removing at least a data point from $\mathcal{S}$ and making merge 2. or 3. not mandatory for $L(\mathcal{A})$.

Henceforth, we have proven that the characteristic sample is independent of the method used, by arriving at a contradiction when trying to show that is possible to have both $|\mathcal{A}| < |\mathcal{B}|$ and $|\mathcal{S}_o^{min}| = |\mathcal{S}_h^{min}|$ alongside the previous definitions in this paper. As the only restriction we have on B is that it is consistent with $\mathcal{S}_o^{min}$, it means that it is not possible to block BlueFringe from making an optimal merge without adding at least an extra data point to its input, as by the statement

proven by this contradiction. This fact can be used in an inductive proof to rule out the case where $|\mathcal{A}| = |\mathcal{B}|$ and $|\mathcal{S}_o^{min}| < |\mathcal{S}_h^{min}|$ starting from a single data point for which $|\mathcal{A}| = |\mathcal{B}| = 1$ and $|\mathcal{S}_o^{min}| = |\mathcal{S}_h^{min}| = 1$ and showing that it is impossible to increase the dataset for both methods without also increasing the size of the DFA for the optimal method.

Following theorem 1, in terms of data efficiency on our new characteristic sample definition from section 3.2 and BlueFringe explanation from section 2.1, the optimal method is as good as the heuristic framework. This fact shows that this characteristic sample is independent of the learning method. Henceforth, we have answered all of our research questions.

4 Responsible Research

In this section, we discuss the ethical implications and reproducibility of our work. The analysis will argue that both considerations are answered by the theoretical nature of our research.

We begin by stating the purely mathematical scope of this paper. Throughout our work, we limited ourselves to solely providing formal statements that follow mathematically from one another. Moreover, all these statements are falsifiable and their truth value can be verified by tracing them back to basic mathematical axioms.

Therefore, no practical experiments had to be done, which greatly reduced the number of ethical aspects that we had to consider. The environmental impact was of little concern for our work because no large models requiring vast computational resources were trained. Furthermore, we consider our work to be as unbiased as the field of theoretical mathematics is. However, we did have to look at the misuse potential of our work in the field of DFA learning, which revolves around the training of predictive models using large amounts of data. We argue that, because we provide no new existing methods for this task but merely theoretically compare existing ones, it is highly unlikely that our work to be used in unethical ways.

Finally, reiterating the fact that we provide no empirical results throughout our work, our work is 100% reproducible and verifiable. This follows from the definition of our fully theoretical scope, where all statements and proofs can be traced and (in)validated from founding mathematical statements.

5 Conclusions and Future Work

We provide formal proofs and mathematical counter-examples that show that, unless formally defined, heuristics can almost always outperform optimal methods on data efficiency benchmarks for DFA learning with no restrictive definitions. Furthermore, we demonstrate how all the heuristic and goal definitions make it impossible to state that one method is strictly better than the other. Finally, we show that, under a limited scope, optimal methods are equal to the BlueFringe framework of the EDSM algorithm, further proving that the characteristic sample is method-independent.

These results imply that the initial statement **Do optimal methods for DFA learning require less data than heuristics to produce correct minimal models?** is false under none or too few restrictions of the field. It is further proven that it is not possible to theoretically quantify the difference in data efficiency between heuristic and optimal methods for every possible scenario. As such, the impact of our research on the bigger picture [1] of DFA learning is that we have shown

[1] https://www.youtube.com/watch?v=WpgNV4Nvetg

that very little more than initial assumptions and definitions can be inferred purely mathematically for a relatively broad context. This fact is consistent with Occam's Razor, stated at the beginning of the paper, as the simplest answer to our question would have been and is, that there is no rule that applies universally.

We end this paper by proposing future research in the context of DFA learning and theoretical analysis of the learning efficiency of heuristic and optimal methods on data efficiency. The next steps of this research are to compare the expected / average data-related benchmarks of the two paradigms. This is more likely to find results in line with the initial hypothesis and available empirical results in the literature showing that more often than not, optimal methods outperform heuristics on this criterion.

Acknowledgments. This study was part of a Bachelor's Degree Thesis and received no direct funding.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Angluin, D.: Learning regular sets from queries and counterexamples. Information and computation **75**(2), 87–106 (1987)
2. Carrasco, R.C., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: International Colloquium on Grammatical Inference. pp. 139–152. Springer (1994)
3. Gold, E.M.: Language identification in the limit. Information and Control **10**(5), 447–474 (1967). https://doi.org/https://doi.org/10.1016/S0019-9958(67)91165-5, https://www.sciencedirect.com/science/article/pii/S0019995867911655
4. Lang, K.J., Pearlmutter, B.A., Price, R.A.: Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In: International Colloquium on Grammatical Inference. pp. 1–12. Springer (1998)
5. Nerode, A., Sauer, B.P.: Fundamental Concepts in the Theory of Systems. ASTIA Document, Wright Air Development Center, Air Research and Development Command, United States Air Force (1957), https://books.google.nl/books?id=QjZwISLU4rAC
6. Smetsers, R., Moerman, J., Jansen, D.N.: Minimal separating sequences for all pairs of states. In: Language and Automata Theory and Applications: 10th International Conference, LATA 2016, Prague, Czech Republic, March 14-18, 2016, Proceedings 10. pp. 181–193. Springer (2016)
7. Verwer, S., Heule, M.J.: Exact DFA identification using SAT solvers. In: Grammatical Inference: Theoretical Results and Applications: 10th International Colloquium, ICGI 2010, Valencia, Spain, September 13-16, 2010. Proceedings 10. pp. 66–79. Springer (2010)