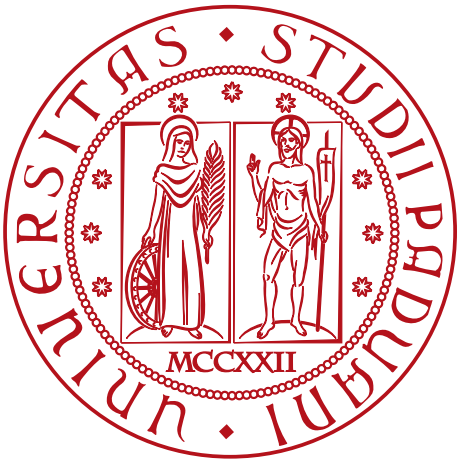
Università degli studi di Padova

Department of Mathematics "Tullio Levi-Civita"

Master of Science in Computer Science



Alternating finite automata through the lens of quasiorders: canonical representation and Angluin-style learning

Master Thesis

Supervisor
Prof. Francesco Ranzato

Co-supervisor
Prof. Pierre Ganty

Candidate
Elia Scandaletti

Academic Year 2023-2024

Elia Scandaletti: *Alternating finite automata through the lens of quasiorders: canonical representation and Angluin-style learning*, Master Thesis © September 2024

"If I have seen further, it is by standing on the shoulders of giants."
— Isaac Newton

Acknowledgements

First of all, I'd like to thank my supervisor, Prof. Francesco Ranzato, not only for the support and precious advice he patiently gave me in the writing of this thesis but also for suggesting to me to do this research abroad. I also want to wholeheartedly express my gratitude to Dr. Pierre Ganty for turning the possibility of doing my thesis at IMDEA into reality and for having always supported me during my stay. Finally, I have to thank all IMDEA's people for having shown me how friendly, stimulating and fun an international research environment can be and for making me feel at home since the first day.

Passando all'italiano, vorrei esprimere la mia gratitudine a coloro che mi sono più vicini.

Vorrei ringraziare i miei amici dell'università, di Croce Rossa e di Ollolanda, senza i quali gli ultimi anni sarebbero stati molto più noiosi e impegnativi e non sarei maturato fino ad essere la persona che sono. In particolare, grazie ad Emma per essere per me un punto di riferimento, sempre pronta ad ascoltarmi e sempre capace di trovare il modo giusto di risollevarmi.

Infine, ultimi ma senza dubbio i più importanti, un grazie di cuore ai miei genitori, le persone che mi supportano da ventiquattro anni, coloro che mi sostengono in ogni mia scelta e mi spingono a cercare la mia strada, anche quando questo significa fare sacrifici o passare notti insonni perché sono distante da casa.

Padova, September 2024 *Elia Scandaletti*

Abstract

The property of possessing a canonical representation is fundamental for any class of automata. While deterministic, non-deterministic, and universal automata have well-established canonical representations, the question remains open for alternating automata. In this study, we propose a definition of canonicity for this class of automata and elucidate several of its properties. Furthermore, we present an algorithm for learning the canonical alternating automata for a given language. This algorithm is a novel adaptation of Angluin's renowned L^* algorithm, originally designed for learning canonical deterministic automata. Our work contributes to the theoretical understanding of alternating automata and provides a practical approach to their canonical representation, potentially opening new avenues for research in automata theory and formal language learning.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1.

Introduction

Regular languages can be defined through various approaches. One prevalent method involves characterizing a regular language as the set of words accepted by finite state automata. Several types of finite automata exist, namely Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NFA), Universal Finite Automata (UFA), and Alternating Finite Automata (AFA), primarily differing in their transition systems between states [LS08; RS97]. While the former types exhibit simpler transitions, the latter are characterized by a reduced number of states.

Specifically, given Q as the set of states and Σ as the alphabet, the transition system of a DFA is defined as $Q \times \Sigma \to Q$, for NFA and UFA as $Q \times \Sigma \to \wp(Q)$, and for AFA as $Q \times \Sigma \to \wp(\wp(Q))$. Notably, NFAs and UFAs can be exponentially more compact than their corresponding DFA, while AFAs can achieve double-exponential compactness [CKS81]. Despite their inherent non-determinism, AFAs can be efficiently represented in deterministic, real-world computers due to their backward determinism as they exhibit deterministic behavior when letters are processed from right to left [SWY00]. An additional advantageous property of AFAs is the computational efficiency of automata operations such as union, intersection, difference, and concatenation in terms of the required number of states [Jir23].

A desirable attribute of DFAs is the uniqueness of the minimal DFA recognizing a given language, up to isomorphism. This property provides a natural definition for canonicity and proves valuable in language comparison scenarios. While this property does not extend to NFAs, it holds for residual NFAs [DLT02]. A residual automaton is defined as an automaton such that all of its states accept a left quotient (or residual) of the language. It has been demonstrated that the minimal residual NFA for a language is unique. It is worth noting that DFAs are always residual. This property is also applicable to UFAs [AEF15]. Consequently, the canonical form for NFAs (resp. UFAs) can be defined as the minimal residual NFA (resp. UFA). However, this approach is not applicable to AFAs, as multiple distinct minimal automata may exist [AEF15].

Figure 1 aims to highlight that minimality can be employed for canonicity only in the case of deterministic automata and residual non-deterministic or universal automata.

Another significant feature shared by DFAs, residual NFAs, and residual UFAs is their favorable learning properties. This implies the existence of algorithms capable of learning an automaton for a given language. Specifically, a learning algorithm is composed of a learner, a teacher and an oracle. The learner can query the teacher about the membership of a word in the language and when sufficient information is gathered, it can construct an automaton and consult the oracle for verification. If the guess is incorrect, the oracle provides a counterexample, prompting the learner to reinitiate the inquiry process with the teacher.

In 1987, D. Angluin introduced the L^* algorithm for learning DFAs [Ang87; Ber+05]. Subsequently, B. Bollig, P. Habermehl, C. Kern, and M. Leucker proposed the NL^* algorithm for learning residual NFAs [Bol+09]. In 2015, D. Angluin, S. Eisenstat, and D. Fisman presented

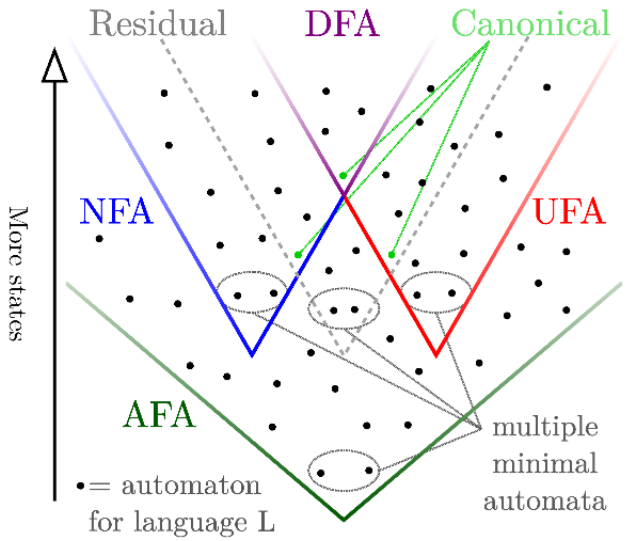


Figure 1: Intuitive representation of canonicity for deterministic, non-deterministic and universal automata. Each dot represents a possible automaton for a given language L.

a template algorithm applicable to learning residual UFAs (UL^*) and AFAs (AL^*), in addition to DFAs and residual NFAs. Finally, S. Berndt, M. Liśkiewicz, M. Lutter, and R. Reischuk formulated the AL^{**} algorithm for learning residual AFAs [Ber+17].

Regular languages can also be defined algebraically by treating them as sets of words with specific properties. In 1957, A. Nerode and J. Myhill proved that a language is regular if and only if it has a finite number of quotients [NM57]. This property can be naturally exploited to define a finite-index equivalence on words. Later, A. de Luca and S. Varricchio expanded on this property, demonstrating that a language is regular if and only if it is closed with respect to two well-quasiorders that are left-monotone and right-monotone [LV94].

These properties were subsequently leveraged by Ganty et al. to provide a framework for deterministic automata construction based on equivalences [GGV19]. A year later, they proposed a similar framework based on quasiorders instead of equivalences [GGV20]. This approach enabled the introduction of non-determinism into the framework.

1.1. Our contributions

Our work originates from a conjecture proposed in [GGV20]. P. Ganty, E. Gutiérrez, and P. Valero investigated the problem of constructing a non-deterministic automaton using quasiorders on words. They demonstrated the possibility of constructing the canonical NFA using special principals –termed L-prime– as states, establishing a bijection between such principals and the states of the automaton. They conjectured that an alternative definition of primality could yield the same result. We have disproved this conjecture through a counterexample and utilized the insights gained as the foundation for the remainder of our work.

In the second part of our research, we introduce a new class of alternating automata called principal automata. We also provide a construction for this class that utilizes $\mathrm{cl}_{\preccurlyeq^r}$-prime principals as states. The definition of primality we employ is the alternative definition proposed in [GGV20],

with the exception of one edge case. We then use this notion to define a canonical automaton as a special case of principal automaton.

Finally, we present a learning algorithm for the canonical alternating automaton of a given language. This algorithm, like other learning algorithms, employs two oracles. The first oracle verifies whether a word belongs to the language, while the second oracle confirms if an automaton correctly recognizes the language and, if not, provides a counterexample.

1.2. Structure of the Thesis

In Chapter 2, we formally discuss the concepts necessary for understanding the remainder of the thesis. In Chapter 3, we disprove the conjecture from [GGV20]. In Chapter 4, we present and define the class of principal automata and, based on this, provide the notion of canonical AFA. In Chapter 5, we present a method to effectively compute a quasiorder based on a table. In Chapter 6, we define a method to effectively construct principal automata, given a computable quasiorder. In Chapter 7, we combine the quasiorder computation with the automata construction from the two previous chapters, resulting in a learning algorithm for canonical AFAs.

For the sake of clarity, when representing an automaton, we will provide a simplified representation, omitting unreachable states and redundant edges from the figures.

Chapter 2.

Background

This chapter presents essential preliminary knowledge requisite for comprehending the subsequent chapters.

2.1. Formal Languages

A formal language is defined as a potentially infinite set of words whose characters belong to a finite set, termed an alphabet. In this dissertation, we exclusively consider words of finite length. Conventionally, Σ denotes an alphabet, while Σ^* refers to the set of all possible finite words over that alphabet. The symbol ε is employed to denote the word of zero length.

The class of regular languages encompasses those languages recognizable by finite state automata, which will be discussed in detail later. For a language L and a word u, we define the left quotient as $u^{-1}L \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid uw \in L\}$ and the right quotient as $Lu^{-1} \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid wu \in L\}$.

2.2. Algebraic Foundations

In subsequent chapters, we will explore relations between words. A relation between words can be conceptualized as a subset of $\Sigma^* \times \Sigma^*$. Consequently, for two relations $\mathcal{R}_1$ and $\mathcal{R}_2$, we can express $\mathcal{R}_1 \subseteq \mathcal{R}_2$, signifying $\forall u, v \in \Sigma^*.\ u\mathcal{R}_1 v \Rightarrow u\mathcal{R}_2 v$. In this context, we say that $\mathcal{R}_1$ is finer than $\mathcal{R}_2$, and conversely, $\mathcal{R}_2$ is coarser than $\mathcal{R}_1$. A relation $\mathcal{R}$ is designated as left (resp. right) if and only if $\forall u, v \in \Sigma^*.\ \forall a \in \Sigma.\ u\mathcal{R}v \Rightarrow au\mathcal{R}av$ (resp. $\forall u, v \in \Sigma^*.\ \forall a \in \Sigma.\ u\mathcal{R}v \Rightarrow ua\mathcal{R}va$).

A quasiorder is defined as a relation $\preccurlyeq$ that is both reflexive and transitive, i.e., $\forall u \in \Sigma^*.\ u \preccurlyeq u$ and $\forall u, v, w \in \Sigma^*.\ u \preccurlyeq v \wedge v \preccurlyeq w \Rightarrow u \preccurlyeq w$. For a quasiorder $\preccurlyeq$, we denote $u \prec v$ as $u \preccurlyeq v \wedge v \not\preccurlyeq u$. An equivalence $\sim$ is a quasiorder that is also symmetric, i.e., $\forall u, v \in \Sigma^*.\ u \preccurlyeq v \Leftrightarrow v \preccurlyeq u$. From a quasiorder, we can induce an equivalence $\sim \stackrel{\text{def}}{=} \preccurlyeq \cap \preccurlyeq^{-1}$. A finite index quasiorder is one for which the induced equivalence has a finite number of equivalence classes. We define a quasiorder $\preccurlyeq$ as L-preserving if and only if $\forall u, v \in \Sigma^*.\ u \preccurlyeq v \wedge u \in L \Rightarrow v \in L$.

A fundamental equivalence on words, the Nerode's equivalence, was introduced by A. Nerode and J. Myhill [NM57]. It is defined as $u \sim_L v \stackrel{\text{def}}{\Longleftrightarrow} u^{-1}L = v^{-1}L$ for any $u, v \in \Sigma^*$. Nerode demonstrated that a language is regular if and only if its corresponding Nerode's equivalence has finite index. Subsequently, A. de Luca and S. Varricchio [LV94] introduced the Nerode's left quasiorder on words, a relaxed variant of the Nerode's equivalence. It is formally defined as $u \preccurlyeq_L^l v \stackrel{\text{def}}{\Longleftrightarrow} Lu^{-1} \subseteq Lv^{-1}$ for any $u, v \in \Sigma^*$. The right counterpart is defined as $u \preccurlyeq_L^r v \stackrel{\text{def}}{\Longleftrightarrow} u^{-1}L \subseteq v^{-1}L$ for any $u, v \in \Sigma^*$. A. de Luca and S. Varricchio also proved that Nerode's left (resp. right) quasiorder is the coarsest L-preserving left (resp. right) quasiorder.

We adopt the notion of closure of a set of words with respect to a quasiorder, as introduced by A. de Luca and S. Varricchio [LV94]. Given a quasiorder $\preccurlyeq$ and a set $S \subseteq \Sigma^*$, we define $\mathrm{cl}_{\preccurlyeq}(S) = \{w \in \Sigma^* \mid \exists u \in S.\ u \preccurlyeq w\}$. A closure is termed principal if and only if $\mathrm{cl}_{\preccurlyeq}(S) = \mathrm{cl}_{\preccurlyeq}(\{u\})$ for some $u \in \Sigma^*$. In such cases, we simply write $\mathrm{cl}_{\preccurlyeq}(u)$.

2.3. Automata Theory

Finite state automata encompass several types distinguished by their transition types: deterministic automata (DFA), non-deterministic automata (NFA), universal automata (UFA), and alternating automata (AFA). All these variants are capable of recognizing precisely the set of regular languages.

A deterministic automaton processes a given word $w = a_1 a_2 ... a_l$ by transitioning from state to state according to its transition function. The automaton accepts the word w from a state q if and only if the state q' reached after reading a_1 accepts the suffix $a_2 ... a_l$.

Definition 1 (Deterministic finite state automata): A deterministic finite state automaton is defined as a tuple $A = (\Sigma, Q, q_0, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states and $\delta : Q \times \Sigma \to Q$ is the transition function.

We denote by $W^D_{q,F}$ the set of words accepted by any state $q \in Q$:

$$W^D_{q,F} = \left\{ax \mid a \in \Sigma, x \in W^D_{\delta(q,a),F}\right\} \cup \varepsilon_q$$

where

$$\varepsilon_q = \begin{cases} \{\varepsilon\} & \text{if } q \in F \\ \emptyset & \text{if } q \notin F \end{cases}.$$

The language of automaton A is denoted as $\mathcal{L}(A) = W^D_{q_0,F}$.

For non-deterministic automata and universal automata, the transition function maps from a state to a set of states. An NFA accepts the word w from state q if and only if at least one of the states $q_1, ..., q_k$ accepts the suffix $a_2 ... a_l$. Conversely, a UFA accepts the word w from state q if and only if all of the states $q_1, ..., q_k$ accept the suffix $a_2 ... a_l$.

Definition 2 (Non-deterministic finite state automata): A non-deterministic finite state automaton is defined as a tuple $A = (\Sigma, Q, I, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states and $\delta : Q \times \Sigma \to \wp(Q)$ is the transition function.

We denote by $W^N_{q,F}$ the set of words accepted by a state $q \in Q$:

$$W^N_{q,F} = \left\{ax \mid a \in \Sigma, \exists q' \in \delta(q,a).\ x \in W^N_{q',F}\right\} \cup \varepsilon_q$$

where

$$\varepsilon_q = \begin{cases} \{\varepsilon\} & \text{if } q \in F \\ \emptyset & \text{if } q \notin F \end{cases}.$$

The language of automaton A is denoted as $\mathcal{L}(A) = \bigcup_{q \in I} W^N_{q,F}$.

Definition 3 (Universal finite state automata): A universal finite state automaton is defined as a tuple $A = (\Sigma, Q, I, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states and $\delta : Q \times \Sigma \to \wp(Q)$ is the transition function.

We denote by $W^U_{q,F}$ the set of words accepted by a state $q \in Q$:

$$W^U_{q,F} = \left\{ax \mid a \in \Sigma, \forall q' \in \delta(q,a).\ x \in W^U_{q',F}\right\} \cup \varepsilon_q$$

where

$$\varepsilon_q = \begin{cases} \{\varepsilon\} \text{ if } q \in F \\ \emptyset \quad \text{if } q \notin F \end{cases}.$$

The language of automaton A is denoted as $\mathcal{L}(A) = \bigcup_{q \in I} W^U_{q,F}$.

Alternating automata represent a generalization of both NFA and UFA. Their transition relation is a function from a state to a positive boolean combination of states. A positive boolean combination of states is either a single state or a disjunction or conjunction of boolean combinations of states. For instance, if a state q after reading a_1 reaches $q_1 \vee (q_2 \wedge q_3)$, q accepts w if and only if q_1 accepts $a_2...a_l$ or both q_2 and q_3 accept $a_2...a_l$. Let Q be the set of states, we denote the set of boolean combinations of states as $\mathcal{B}(Q)$. Throughout this work, we assume, without loss of generality, that boolean combinations are in disjunctive normal form.

Definition 4 (Alternating finite state automata): An alternating finite state automaton is defined as a tuple $A = (\Sigma, Q, I, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $I \in \mathcal{B}(Q)$ is a positive boolean combination of initial states, $F \subseteq Q$ is the set of final states and $\delta : Q \times \Sigma \to Q$ is the transition function.

We denote by $W^A_{B,F}$ the set of words accepted by a boolean combination $B \in \mathcal{B}(Q)$:

$$W^A_{q,F} = \left\{ax \mid a \in \Sigma, x \in W^A_{\delta(q,a),F}\right\} \cup \varepsilon_q$$
$$W^A_{B_1 \wedge B_2,F} = W^A_{B_1,F} \cap W^A_{B_2,F}$$
$$W^A_{B_1 \vee B_2,F} = W^A_{B_1,F} \cup W^A_{B_2,F}$$

where

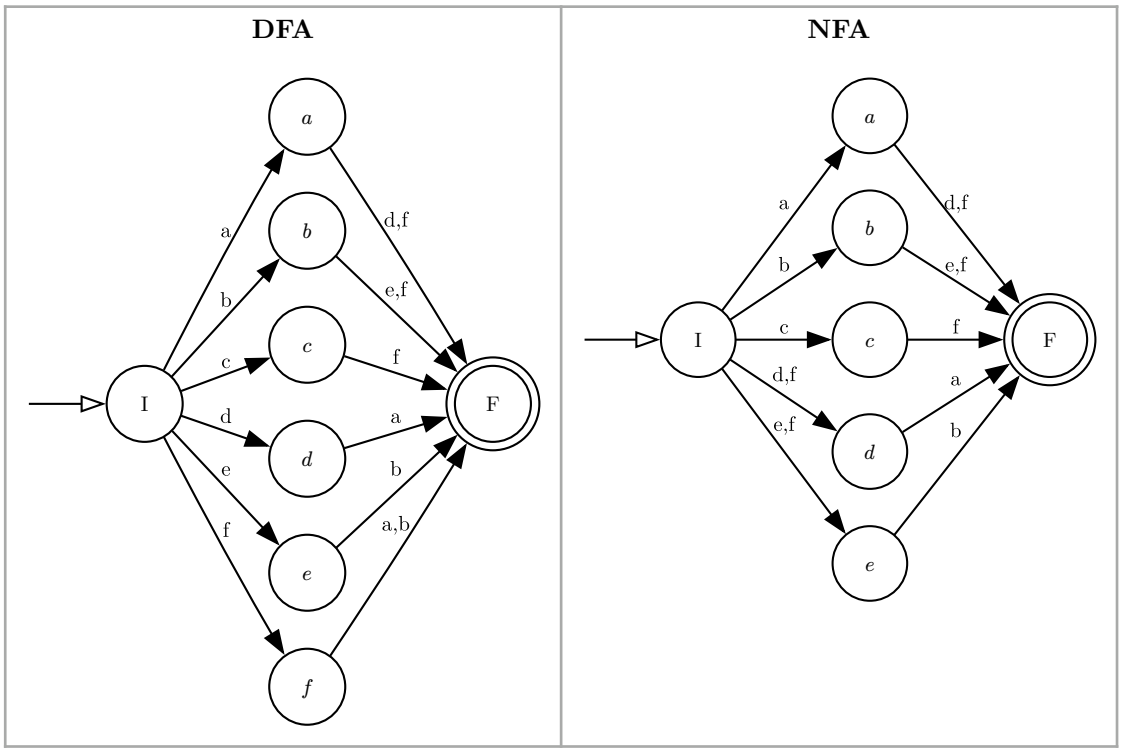
$$\varepsilon_q = \begin{cases} \{\varepsilon\} \text{ if } q \in F \\ \emptyset \quad \text{if } q \notin F \end{cases}.$$

The language of automaton A is denoted as $\mathcal{L}(A) = W^A_{I,F}$.

It is important to note that in Definition 4, we employ positive boolean combinations, as in D. Angluin, S. Eisenstat, and D. Fisman [AEF15]. This approach differs from the definition provided in K. Salomaa, X. Wu, and S. Yu [SWY00], which also permits the use of negation.

In the remainder of this work, we will use $W_{B,F}$ instead of $W^A_{B,F}$ for the sake of conciseness and clarity, as there is no ambiguity in this context.

Example 1: Table 1 illustrates diverse automata types for the language $L = \{ad, af, be, bf, cf, da, eb, fa, fb\}$. In the non-deterministic finite automaton (NFA), state f is substituted by two distinct edges, demonstrating the NFA's capacity for multiple transitions. Conversely, in the universal finite automaton (UFA), state c is replaced by two edges, highlighting the UFA's ability to represent universal quantification over transitions. Alternating finite automata (AFA) exhibit superior expressive power, enabling the replacement of both aforementioned states. Specifically, state c is substituted by a universal edge, while state f is replaced by an existential edge, showcasing the AFA's ability to combine both universal and existential quantification in its state transitions.



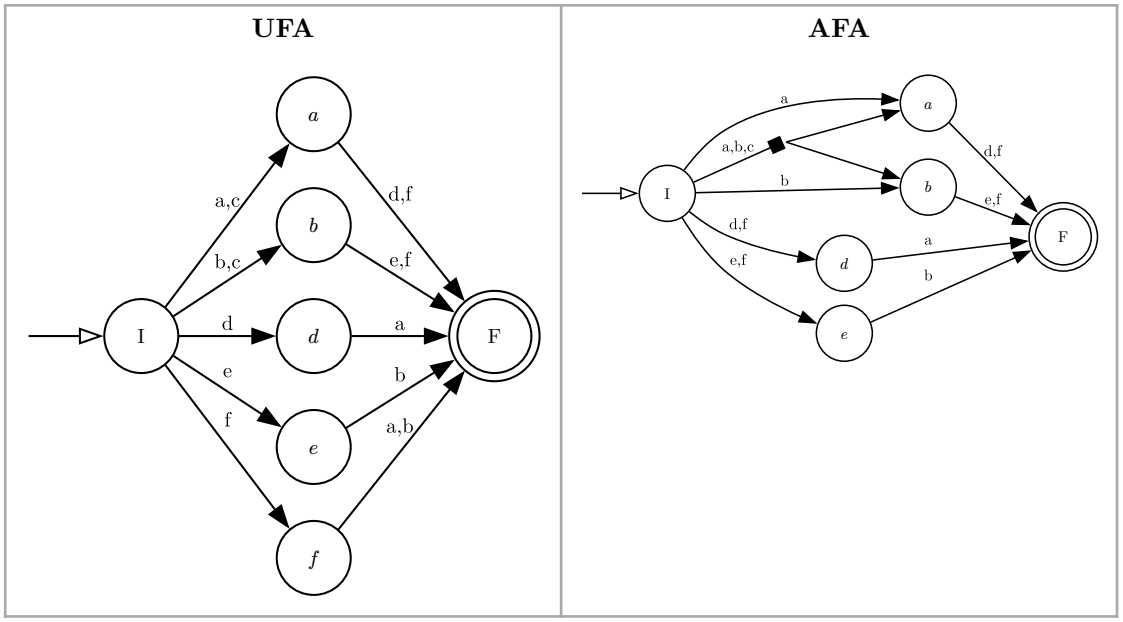


Table 1: Automata of different types recognizing the same language $L = \{ad, af, be, bf, cf, da, eb, fa, fb\}$.

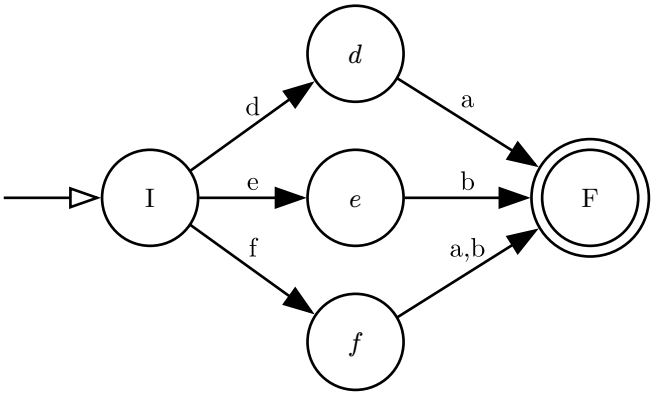
Chapter 3.

Counterexample to a conjecture on NFAs by Ganty et al.

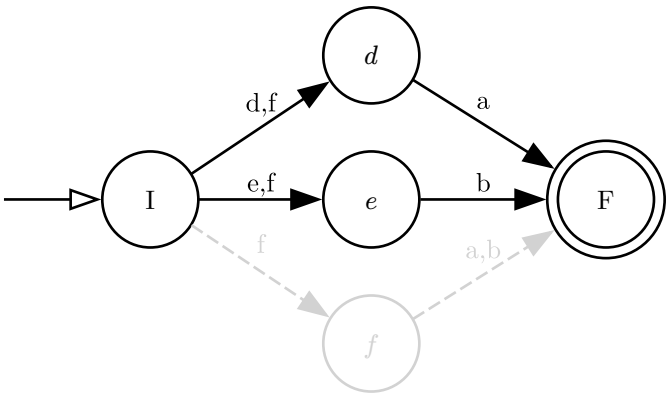
The construction of deterministic finite automata (DFA) is intrinsically linked to the application of congruences, a well-established principle in automata theory. In 2019, P. Ganty, E. Gutiérrez, and P. Valero [GGV19] proposed a novel framework for deterministic automata construction utilizing congruences. Their approach employed principals induced by congruences as states for automata construction. Subsequently, in [GGV20], they extended this concept by substituting congruences with quasiorders. This modification allowed principals to overlap, thereby introducing non-determinism into their construction framework. Specifically, their framework yielded residual non-deterministic automata.

The introduction of non-determinism in the construction framework rendered some principals superfluous. As demonstrated in Example 2, certain states can be eliminated without altering the language accepted by the automaton, although in some instances, new transitions may be necessary to preserve the accepted language.

Example 2: Consider the language $L = \{da, eb, fa, fb\}$ and the following automaton recognizing it:



In this instance, the state f can be eliminated as its language is the union of the languages of states d and e. Its incoming edge should be "divided" among the other two states, resulting in the following automaton:



This property necessitated the differentiation between necessary and superfluous states. Necessary states are termed L-prime, while superfluous states are designated as L-composite.

Definition 5 (L-composite [GGV20, Def. 2]): Let L be a regular language and $\preccurlyeq^r$ be a right quasiorder on Σ^*. For $u \in \Sigma^*$, the principal $\mathrm{cl}_{\preccurlyeq^r}(u)$ is L-composite if and only if

$$u^{-1}L = \bigcup_{x \prec^r u} x^{-1}L.$$

A principal that is not L-composite is defined as L-prime.

The following definition delineates their proposed automata construction method.

Definition 6 (Automata construction $\mathrm{H}^r(\preccurlyeq^r, L)$ [GGV20, Def. 3]): Let $\preccurlyeq^r$ be a right finite index quasiorder and $L \subseteq \Sigma^*$ be a language. Define $\mathrm{H}^r(\preccurlyeq^r, L) = (Q, \Sigma, \delta, I, F)$ as a NFA where:

- $Q = \{\mathrm{cl}_{\preccurlyeq^r}(u) \mid u \in \Sigma^* \wedge \mathrm{cl}_{\preccurlyeq^r}(u) \text{ is } L\text{-prime}\}$;
- $I = \{\mathrm{cl}_{\preccurlyeq^r}(u) \in Q \mid \varepsilon \in \mathrm{cl}_{\preccurlyeq^r}(u)\}$;
- $F = \{\mathrm{cl}_{\preccurlyeq^r}(u) \in Q \mid u \in L\}$;
- $\delta(\mathrm{cl}_{\preccurlyeq^r}(u), a) = \{\mathrm{cl}_{\preccurlyeq^r}(v) \in Q \mid \mathrm{cl}_{\preccurlyeq^r}(u)a \subseteq \mathrm{cl}_{\preccurlyeq^r}(v)\}$ for all $\mathrm{cl}_{\preccurlyeq^r}(u) \in Q, a \in \Sigma$.

It is important to note that in Definition 5, the L-composite property of a principal depends on the corresponding left quotient. Consequently, the entire construction relies not only on principals but also on quotients.

In an effort to eliminate quotients from the construction, the authors sought an alternative definition of composability that depended solely on principals. They proved the following lemma, which states that for the right Nerode's quasiorder $\preccurlyeq^r_L$, composite principals can be described as intersections of prime principals.

Lemma 7 ([GGV20, Lemma 35]): Let $N = (Q, \Sigma, \delta, I, F)$ be an NFA with $\mathcal{L}(N) = L$. Then for any $u \in \Sigma^*$

$$u^{-1}L = \bigcup_{x \in \Sigma^*, x \prec_L^r u} x^{-1}L \implies \mathrm{cl}_{\preccurlyeq_L^r}(u) = \bigcap_{x \in \Sigma^*, x \prec_L^r u} \mathrm{cl}_{\preccurlyeq_L^r}(x).$$

Simultaneously, they conjectured that this implication is, in fact, an equivalence. In other words, they conjectured that the right side of the implication serves as an alternative and equivalent definition of composability.

Example 3: A counterexample to this conjecture is the language $L = \{ad, be, cd, ce, cf\}$.

It can be readily verified that

$$\exists u. \quad \mathrm{cl}_{\preccurlyeq_L^r}(u) = \bigcap_{x \prec u} \mathrm{cl}_{\preccurlyeq_L^r}(x) \quad \wedge \quad u^{-1}L \neq \bigcup_{x \prec u} x^{-1}L$$

for $u = c$.

In Table 2, the discrepancy between the two definitions of primality is evident in the row corresponding to c. This row demonstrates that $\mathrm{cl}_{\preccurlyeq_L^r}(u)$ is L-prime but composite according to the alternative definition.

u	$u^{-1}L$	$\mathrm{cl}_{\preccurlyeq_L^r}(u)$	$u^{-1}L \stackrel{?}{=} \bigcup_{x \prec u} x^{-1}L$	$\mathrm{cl}_{\preccurlyeq_L^r}(u) \stackrel{?}{=} \bigcap_{x \prec u} \mathrm{cl}_{\preccurlyeq_L^r}(x)$
ε	L	$\{\varepsilon\}$	$L \neq \emptyset$	$\{\varepsilon\} \neq \Sigma^*$
a	$\{d\}$	$\{a, c\}$	$\{d\} \neq \emptyset$	$\{a, c\} \neq \Sigma^*$
b	$\{e\}$	$\{b, c\}$	$\{e\} \neq \emptyset$	$\{b, c\} \neq \Sigma^*$
c	$\{d, e, f\}$	$\{c\}$	$\{d, e, f\} \neq \{d\} \cup \{e\}$	$\{c\} = \{a, c\} \cap \{b, c\}$
$u \in L$	$\{\varepsilon\}$	L	$\{\varepsilon\} \neq \emptyset$	$L \neq \Sigma^*$
other u	$\emptyset$	Σ^*	$\emptyset = \emptyset$	$\Sigma^* = \Sigma^*$

Table 2: Each row corresponds to different values of u. Each row contains the residual of u, the principal of u, and verifies if $\mathrm{cl}_{\preccurlyeq_L^r}(u)$ is composite according to L-primality and the alternative definition. The row corresponding to c highlights the discrepancy in the definitions.

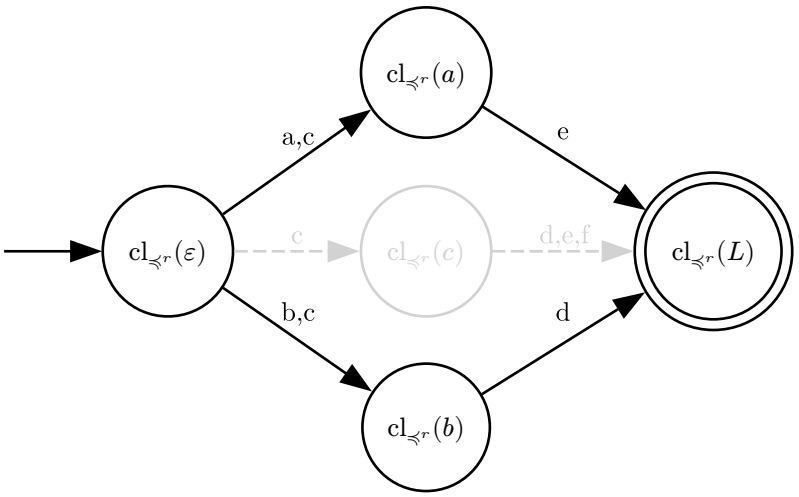


Figure 2: Automaton derived by applying construction H^r to the language $\{ad, be, cd, ce, cf\}$ using the alternative definition of primality. The gray dashed node and edges are excluded from the automaton, as $\{c\}$ is L-prime but not prime according to the alternative definition.

As illustrated in Figure 2, it is not feasible to construct an NFA with only four states that accepts L. To achieve this, it would be necessary to modify the automaton by incorporating a transition consuming f from *both* $\mathrm{cl}_{\preccurlyeq^r}(a)$ and $\mathrm{cl}_{\preccurlyeq^r}(b)$ to $\mathrm{cl}_{\preccurlyeq^r}(L)$. While this is not possible in an NFA, it is similar to a universal transition in AFAs, albeit in a "reversed" manner. The language of the AFA in Figure 3 is L^r. The automaton is constructed by reversing the automaton in Figure 2 and adding a transition labeled f from $\mathrm{cl}_{\preccurlyeq^r}(L)$ to $\mathrm{cl}_{\preccurlyeq^r}(a) \wedge \mathrm{cl}_{\preccurlyeq^r}(b)$.

This observation serves as the foundation for the subsequent part of our research.

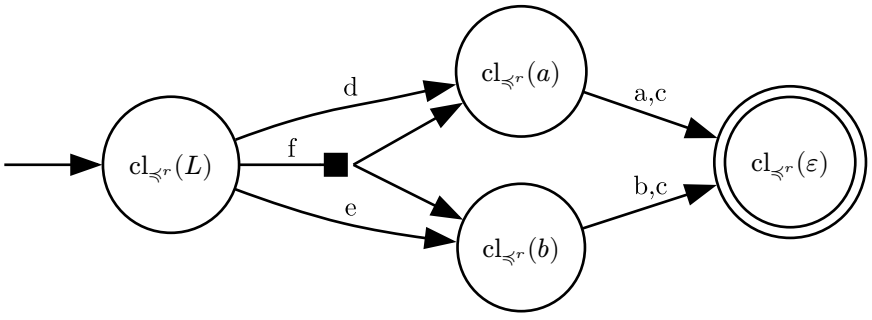


Figure 3: Automaton obtained by transforming the automaton in Figure 2 in order to recognize L^r. The added transition forms the basis for the next part of our research.

Chapter 4.

Canonical Alternating Automata

In this chapter we first define the class of principal automata. We will also provide a way to build such automata which we will later use to define the canonical automaton for a language.

4.1. Principal AFA

Definition 8 (Principal Automata): Let $A = (\Sigma, Q, I, F, \delta)$ be an AFA, we say that A is principal if and only if exists a left finite index quasiorder such that

$$\forall q \in Q.\ \exists u \in \Sigma^*.\ W_{q,F} = \mathrm{cl}_{\preccurlyeq^l}(u)$$

Definition 9 (Composite principal): Let $\preccurlyeq$ be a quasiorder on Σ^*. Given $u \in \Sigma^*$, let $\mathrm{Sup}(u) = \{\mathrm{cl}_{\preccurlyeq}(z) \mid z \in \Sigma^*, \mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(z)\}$, then we say that the principal $\mathrm{cl}_{\preccurlyeq}(u)$ is $\mathrm{cl}_{\preccurlyeq^l}$-composite if and only if

$$\mathrm{Sup}(u) \neq \emptyset \wedge \mathrm{cl}_{\preccurlyeq}(u) = \bigcap \mathrm{Sup}(u).$$

A principal is $\mathrm{cl}_{\preccurlyeq^l}$-prime if and only if it is not $\mathrm{cl}_{\preccurlyeq^l}$-composite.

The idea of composability is similar to the one depicted in Definition 5. The main difference on an intuitive level is that now we are discussing principal automata, not residual automata and therefore we are looking for principals which are the combination of distinct principals, instead of quotients. Since it's not possible for a principal to be the union of distinct smaller principals, we only consider the case in which a principal is the intersection of distinct larger principals.

When we find a composite closure, we can remove the corresponding state and replace it with an universal transition to all the states composing the one that has been removed.

Definition 10 (Automata construction $\mathrm{A}(L, \preccurlyeq^l)$): Let $\preccurlyeq^l$ be a left finite index quasiorder and let $L \subseteq \Sigma^*$ be a regular language. We define $\mathrm{A}(L, \preccurlyeq^l) = (\Sigma, Q, I, F, \delta)$ where:

$$Q = \{\mathrm{cl}_{\preccurlyeq^l}(u) \mid u \in \Sigma^*, \mathrm{cl}_{\preccurlyeq^l}(u) \text{ is } \mathrm{cl}_{\preccurlyeq^l}\text{-composite}\};$$

$$I = \left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u) \in Q \\ u \in L}} \mathrm{cl}_{\preccurlyeq^l}(u) \right) \vee \left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u) \notin Q \\ u \in L}} \bigwedge_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z) \in Q \\ z \prec^l u}} \mathrm{cl}_{\preccurlyeq^l}(z) \right);$$

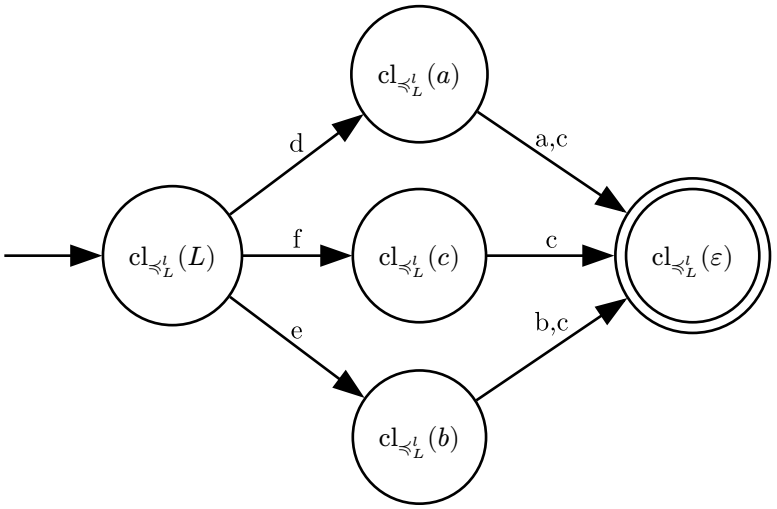
$$F = \{\mathrm{cl}_{\preccurlyeq^l}(u) \in Q \mid u \preccurlyeq^l \varepsilon\}.$$

Before defining the transition system, we have to define a support set. Namely, for any $u \in \Sigma^*$ and $a \in \Sigma$ we define $Q_u^a = \{\mathrm{cl}_{\preccurlyeq^l}(v) \mid v \in \Sigma^*, a\mathrm{cl}_{\preccurlyeq^l}(v) \subseteq \mathrm{cl}_{\preccurlyeq^l}(u)\}$. We can now define $\delta : Q \times \Sigma \to \mathcal{B}(Q)$ as

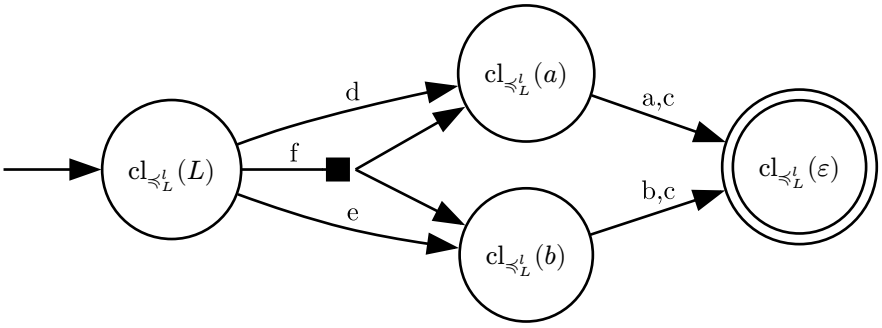
$$\delta(\mathrm{cl}_{\preccurlyeq^l}(u), a) = \left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(u) \in Q_u^a \cap Q} \mathrm{cl}_{\preccurlyeq^l}(u) \right) \vee \left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(u) \in Q_u^a \setminus Q} \; \bigwedge_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z) \in Q \\ z \prec^l u}} \mathrm{cl}_{\preccurlyeq^l}(z) \right).$$

Example 4: In this example we compare a NFA whose states are principals with the AFA $\mathrm{A}(L, \preccurlyeq_L^l)$. We will use $L = \{da, eb, dc, ec, fc\}$, with $\preccurlyeq_L^l$ being the Nerode's left quasiorder [LV94].

The following is a principal NFA



Note that $\mathrm{cl}_{\preccurlyeq_L^l}(c)$ is $\mathrm{cl}_{\preccurlyeq^l}$-composite, therefore we can replace it with an universal transition.



This automaton actually correspond to $\mathrm{A}(L, \preccurlyeq_L^l)$.

The most important characteristics of this construction, which we will use later, is that it always produces a principal automaton (Theorem 12) and that the automaton $\mathrm{A}\big(L, \preccurlyeq^l\big)$ correctly recognize L if and only if $\preccurlyeq^l$ is L-preserving (Theorem 14).

Lemma 11 states that, under some conditions, when considering the intersection of principals we can consider only prime principals. This lemma is important as it will greatly simplify many proofs.

Lemma 11: Given a regular language L and a finite index quasiorder $\preccurlyeq$, then for any $u \in \Sigma^*$

$$\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(z) \in Q^P} \mathrm{cl}_\preccurlyeq(z) = \bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(y)} \mathrm{cl}_\preccurlyeq(y)$$

where $Q^P = \big\{\mathrm{cl}_{\preccurlyeq^l}(z) \mid z \in \Sigma^*, \mathrm{cl}_{\preccurlyeq^l}(z) \text{ is } \mathrm{cl}_{\preccurlyeq^l}\text{-prime}\big\}$.

Proof of Lemma 11: We proceed by induction on the number of closures which are proper supersets of $\mathrm{cl}_\preccurlyeq(u)$. This is a well founded recursion because the number of closures is finite.

Base case: Assume $\nexists \mathrm{cl}_\preccurlyeq(v).\, \mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(v)$. Then the lemma is trivially true, since $\Sigma^* = \Sigma^*$.

Inductive step: We assume the lemma holds for all closures which are proper supersets of $\mathrm{cl}_\preccurlyeq(u)$. Let $Q^C = \big\{\mathrm{cl}_{\preccurlyeq^l}(z) \mid z \in \Sigma^*, \mathrm{cl}_{\preccurlyeq^l}(z) \text{ is } \mathrm{cl}_{\preccurlyeq^l}\text{-composite}\big\}$.

Observe that

$$\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(z)} \mathrm{cl}_\preccurlyeq(z) = \left(\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(z) \in Q^P} \mathrm{cl}_\preccurlyeq(z)\right) \cap \left(\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(y) \in Q^C} \mathrm{cl}_\preccurlyeq(y)\right).$$

Therefore, it is sufficient to prove that

$$\left(\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(z) \in Q^P} \mathrm{cl}_\preccurlyeq(z)\right) \subseteq \left(\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(y) \in Q^C} \mathrm{cl}_\preccurlyeq(y)\right).$$

For all $\mathrm{cl}_\preccurlyeq(y) \in Q^C$, we know that $\mathrm{cl}_\preccurlyeq(y) = \bigcap_{\mathrm{cl}_\preccurlyeq(y) \subset \mathrm{cl}_\preccurlyeq(x)} \mathrm{cl}_\preccurlyeq(x)$. Therefore, by inductive hypothesis, $\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(y) \in Q^C \Rightarrow \mathrm{cl}_\preccurlyeq(y) = \bigcap_{\mathrm{cl}_\preccurlyeq(y) \subset \mathrm{cl}_\preccurlyeq(x) \in Q^P} \mathrm{cl}_\preccurlyeq(x)$.

We can thus rewrite it as

$$\left(\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(z) \in Q^P} \mathrm{cl}_\preccurlyeq(z)\right) \subseteq \left(\bigcap_{\mathrm{cl}_\preccurlyeq(u) \subset \mathrm{cl}_\preccurlyeq(y) \in Q^C} \bigcap_{\mathrm{cl}_\preccurlyeq(y) \subset \mathrm{cl}_\preccurlyeq(x) \in Q^P} \mathrm{cl}_\preccurlyeq(x)\right),$$

which is equivalent to

$$\left(\bigcap_{\mathrm{cl}_\preccurlyeq(z) \in Q_1} \mathrm{cl}_\preccurlyeq(z)\right) \subseteq \left(\bigcap_{\mathrm{cl}_\preccurlyeq(x) \in Q_2} \mathrm{cl}_\preccurlyeq(x)\right),$$

where $Q_1 = \{\mathrm{cl}_{\preccurlyeq}(z) \in Q^P \mid \mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(z)\}$ and $Q_2 = \{\mathrm{cl}_{\preccurlyeq}(x) \in Q^P \mid \mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(x) \wedge \exists \mathrm{cl}_{\preccurlyeq}(y) \in Q^C.\ \mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(y) \subset \mathrm{cl}_{\preccurlyeq}(x)\}$.

This last equation holds since $Q_2 \subseteq Q_1$.

■

Theorem 12 ($\mathrm{A}(L, \preccurlyeq^l)$ is principal): Given a regular language L and a left finite index quasiorder $\preccurlyeq^l$, then $\mathrm{A}(L, \preccurlyeq^l) = (\Sigma, Q, I, F, \delta)$ is principal. In particular,

$$\forall \mathrm{cl}_{\preccurlyeq^l}(u) \in Q.\ W_{\mathrm{cl}_{\preccurlyeq^l}(u),F} = \mathrm{cl}_{\preccurlyeq^l}(u).$$

Proof of Theorem 12: For simplicity, in the proof we will use the notation cl instead of $\mathrm{cl}_{\preccurlyeq^l}$.

Let $\mathrm{A}(L, \preccurlyeq^l) = (\Sigma, Q, I, F, \delta)$.

We first prove that $w \in \mathrm{cl}(u) \Rightarrow w \in W_{\mathrm{cl}(u),F}$ by induction on the length of w, for any $\mathrm{cl}(u) \in P$.

Base case: Assume $w = \varepsilon$. Then, $\varepsilon \in \mathrm{cl}(u) \Rightarrow u \preccurlyeq^l \varepsilon \Rightarrow \mathrm{cl}(u) \in F \Rightarrow \varepsilon \in W_{\mathrm{cl}(u),F}$.

Inductive step: Assume $w = ax, a \in \Sigma, x \in \Sigma^*$ and assume, by inductive hypothesis that $x \in \mathrm{cl}(z) \Rightarrow x \in W_{\mathrm{cl}(z),F}$ for all $\mathrm{cl}(z) \in Q$.

$$\begin{aligned} ax \in \mathrm{cl}(u) &\Rightarrow [\text{By def of cl}] \\ \mathrm{cl}(ax) \subseteq \mathrm{cl}(u) &\Rightarrow [\because a\ \mathrm{cl}(x) \subseteq \mathrm{cl}(ax)\ [\text{GGV20, Lemma 1}]] \\ a\ \mathrm{cl}(x) \subseteq \mathrm{cl}(u) &\Rightarrow \\ \mathrm{cl}(x) \in Q^a_u. & \end{aligned}$$

We now have two cases:

$\mathrm{cl}(x)$ *prime*:

$$\begin{aligned} \mathrm{cl}(x) \in Q^a_u \cap Q &\Rightarrow \\ \mathrm{cl}(x) \subseteq \bigcup_{\mathrm{cl}(v) \in (Q^a_u \cap Q)} \mathrm{cl}(v) &\Rightarrow \\ x \in \bigcup_{\mathrm{cl}(v) \in (Q^a_u \cap Q)} \mathrm{cl}(v) &\Rightarrow \\ ax \in a \bigcup_{\mathrm{cl}(v) \in (Q^a_u \cap Q)} \mathrm{cl}(v) &\Rightarrow \\ ax \in W_{\mathrm{cl}(u),F}. & \end{aligned}$$

$\mathrm{cl}(x)$ *composite*:

$$\begin{aligned} \mathrm{cl}(x) \in Q^a_u \setminus Q &\Rightarrow \\ \mathrm{cl}(x) = \bigcap_{\mathrm{cl}(x) \subset\ \mathrm{cl}(z)} \mathrm{cl}(z) &\Rightarrow [\text{By Lemma 11}] \end{aligned}$$

$$\begin{aligned}
\mathrm{cl}(x) = \bigcap_{\mathrm{cl}(x)\subset\ \mathrm{cl}(z)\in Q} \mathrm{cl}(z) &\Rightarrow \text{[By the inductive hypotesis]}\\
\mathrm{cl}(x) \subseteq \bigcap_{\mathrm{cl}(x)\subset\ \mathrm{cl}(z)\in Q} W_{\mathrm{cl}(z),F} &\Rightarrow\\
x \in \bigcap_{\mathrm{cl}(x)\subset\ \mathrm{cl}(z)\in Q} W_{\mathrm{cl}(z),F} &\Rightarrow\\
ax \in a \bigcap_{\mathrm{cl}(x)\subset\ \mathrm{cl}(z)\in Q} W_{\mathrm{cl}(z),F} &\Rightarrow \text{[Because } \mathrm{cl}(x) \in Q^a_u \setminus Q\text{]}\\
ax \in W_{\mathrm{cl}(u),F}.&
\end{aligned}$$

In both cases, we get $ax \in W_{\mathrm{cl}(u),F}$.

Then we prove that $w \in W_{\mathrm{cl}(u),F} \Rightarrow w \in \mathrm{cl}(u)$ by induction on the length of w, for $\mathrm{cl}(u) \in Q$.

Base case: Assume $w = \varepsilon$. Then, $\varepsilon \in W_{\mathrm{cl}(u),F} \Rightarrow \mathrm{cl}(u) \in F \Rightarrow u \preccurlyeq^l \varepsilon \Rightarrow \varepsilon \in \mathrm{cl}(u)$.

Inductive step: Assume $w = ax, a \in \Sigma, x \in \Sigma^*$ and assume, by inductive hypothesis that $x \in W_{\mathrm{cl}(y),F} \Rightarrow x \in \mathrm{cl}(y)$ for all $\mathrm{cl}(y) \in Q$.

$$\begin{aligned}
ax \in W_{\mathrm{cl}(u),F} &\Rightarrow\\
\exists\ \mathrm{cl}(v) \in Q^a_u \cap Q.\ x \in W_{\mathrm{cl}(v),F} \vee&\\
\exists\ \mathrm{cl}(v) \in Q^a_u \setminus Q.\ \forall\ \mathrm{cl}(z) \in Q, \mathrm{cl}(v) \subset \mathrm{cl}(z).\ x \in W_{\mathrm{cl}(z),F} &\Rightarrow\\
\exists\ \mathrm{cl}(v) \in Q^a_u.\ \begin{cases} x \in W_{\mathrm{cl}(v),F} & \text{if } \mathrm{cl}(v) \in Q\\ \forall\ \mathrm{cl}(z) \in Q, \mathrm{cl}(v) \subset \mathrm{cl}(z).\ x \in W_{\mathrm{cl}(z),F} & \text{if } \mathrm{cl}(v) \notin Q \end{cases} &\Rightarrow\\
\text{[By inductive hypotesis]}&\\
\exists\ \mathrm{cl}(v) \in Q^a_u.\ \begin{cases} x \in \mathrm{cl}(v) & \text{if } \mathrm{cl}(v) \in Q\\ \forall\ \mathrm{cl}(z) \in Q, \mathrm{cl}(v) \subset \mathrm{cl}(z).\ x \in \mathrm{cl}(z) & \text{if } \mathrm{cl}(v) \notin Q \end{cases} &\Rightarrow\\
\exists\ \mathrm{cl}(v) \in Q^a_u.\ \begin{cases} x \in \mathrm{cl}(v) & \text{if } \mathrm{cl}(v) \in Q\\ x \in \bigcap_{\mathrm{cl}(v)\subset\ \mathrm{cl}(z)\in Q} \mathrm{cl}(z) & \text{if } \mathrm{cl}(v) \notin Q \end{cases} &\Rightarrow\\
\text{[By Lemma 11]}&\\
\exists\ \mathrm{cl}(v) \in Q^a_u.\ \begin{cases} x \in \mathrm{cl}(v) & \text{if } \mathrm{cl}(v) \in Q\\ x \in \bigcap_{\mathrm{cl}(v)\subset\ \mathrm{cl}(z)\in Q} \mathrm{cl}(z) & \text{if } \mathrm{cl}(v) \notin Q \end{cases} &\Rightarrow\\
\text{[By Definition 9]}&\\
\exists\ \mathrm{cl}(v) \in Q^a_u.\ \begin{cases} x \in \mathrm{cl}(v) & \text{if } \mathrm{cl}(v) \in Q\\ x \in \mathrm{cl}(v) & \text{if } \mathrm{cl}(v) \notin Q \end{cases} &\Rightarrow\\
\exists\ \mathrm{cl}(v) \in Q^a_u.\ x \in \mathrm{cl}(v) &\Rightarrow\\
\text{[By definition of } Q^a_u\text{]}&\\
ax \in \mathrm{cl}(u).&
\end{aligned}$$

■

Lemma 13: Given a regular language L and a left finite index quasiorder $\preccurlyeq^l$, then

$$\mathcal{L}\big(\mathrm{A}\big(L, \preccurlyeq^l\big)\big) = \mathrm{cl}_{\preccurlyeq^l}(L).$$

Proof of Lemma 13:

$$\begin{aligned}
\mathcal{L}\big(\mathrm{A}\big(L, \preccurlyeq^l\big)\big) &= \text{[By Definition 4]}\\
W_{I,F} &= \text{[By Definition 10]}\\
\left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q\\ u\in L}} W_{u,F}\right) \cup \left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q\\ u\in L}} \bigcap_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z)\in Q\\ z\prec^l u}} W_{z,F}\right) &= \text{[By Theorem 12]}\\
\left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q\\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \cup \left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q\\ u\in L}} \bigcap_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z)\in Q\\ z\prec^l u}} \mathrm{cl}_{\preccurlyeq^l}(z)\right) &= \text{[By Lemma 11]}\\
\left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q\\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \cup \left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q\\ u\in L}} \bigcap_{z\prec^l u} \mathrm{cl}_{\preccurlyeq^l}(z)\right) &= \text{[By Definition 9]}\\
\left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q\\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \cup \left(\bigcup_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q\\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) &=\\
\bigcup_{u\in L} \mathrm{cl}_{\preccurlyeq^l}(u) &=\\
\mathrm{cl}_{\preccurlyeq^l}(L).
\end{aligned}$$

■

Theorem 14 (Correctness of $\mathrm{A}\big(L, \preccurlyeq^l\big)$): Given a regular language L and a left finite index quasiorder $\preccurlyeq^l$, then

$$\mathcal{L}\big(\mathrm{A}\big(L, \preccurlyeq^l\big)\big) = L \quad \Leftrightarrow \quad \preccurlyeq^l \text{ is } L\text{-preserving}.$$

Proof of Theorem 14: Let $\mathrm{A}\big(L, \preccurlyeq^l\big) = (\Sigma, Q, I, F, \delta)$.

We easily prove that if $\preccurlyeq^l$ is L-preserving, then $\mathcal{L}\big(\mathrm{A}\big(L, \preccurlyeq^l\big)\big) = L$ thanks to Lemma 13.

Then we prove that if $\preccurlyeq^l$ is not L-preserving, then $\mathcal{L}\big(A\big(L, \preccurlyeq^l\big)\big) \neq L$.

$$\begin{aligned}
&\preccurlyeq^l \text{ not } L\text{-preserving} \Leftrightarrow\\
&\exists u, v \in \Sigma^*.\ u \in L \wedge v \in \mathrm{cl}_{\preccurlyeq^l}(u) \wedge v \notin L \Rightarrow\\
&\exists v \in \Sigma^*.\ v \in \mathcal{L}\big(\mathrm{A}\big(L, \preccurlyeq^l\big)\big) \wedge v \notin L \Leftrightarrow
\end{aligned}$$

$$\exists v \in \Sigma^*.\ v \in \mathrm{cl}_{\preccurlyeq^l}(L) \wedge v \notin L \Leftrightarrow \text{[From Lemma 13]}$$
$$\exists v \in \Sigma^*.\ v \in \mathcal{L}\big(\mathrm{A}\big(L, \preccurlyeq^l\big)\big) \wedge v \notin L.$$

■

Observe that $\mathrm{A}\big(L, \preccurlyeq^l\big)$ is an alternating automata construction. It may not be an UFA nor an NFA. This claim is supported by Figure 4, since $\delta(L, c) = \{\varepsilon\} \vee (\{d, f\} \wedge \{e, f\})$.

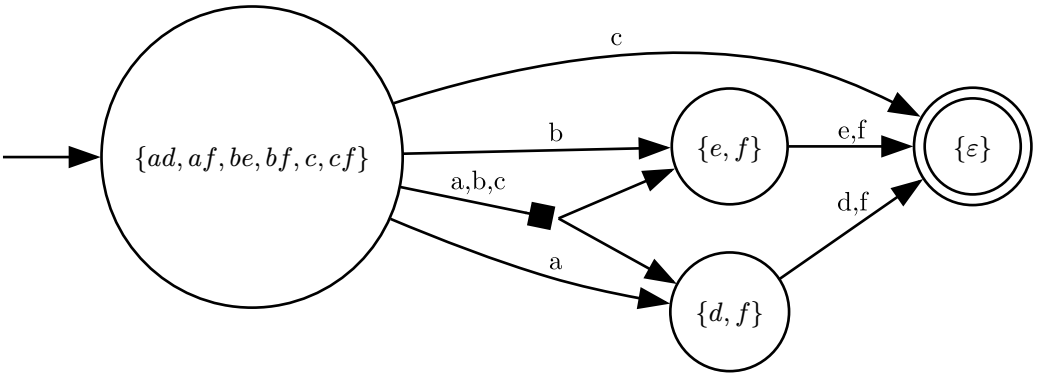


Figure 4: The automaton $\mathrm{A}\big(L, \preccurlyeq^l_L\big)$ for language $L = \{ad, af, be, bf, c, cf\}$ and Nerode's left quasiorder, as described in Definition 10.

4.2. Canonical AFA

Now that we have defined the class of principal automata, we can define the canonical automaton for a given language.

Definition 15 (Canonical AFA): Given a regular language L, we define the canonical automaton for such language as $\mathrm{A}\big(L, \preccurlyeq^l_L\big)$, where $\preccurlyeq^l_L$ is the Nerode's quasiorder for L.

Theorem 16 (Correctness of Canonical AFA): The canonical automaton for a language L recognize the language L.

Proof of Theorem 16: It follows immediately from Theorem 14 along with the fact that the Nerode quasiorder is L-preserving. ■

The canonical automaton is indeed unique, up to isomorphism, for any language by construction. We conjecture that it is also the minimal principal automaton in the number of states, even though we have not been able to prove it. We think so because the Nerode's quasiorder is the coarsest L-preserving quasiorder [LV94]. Therefore, this quasiorder induces the minimum number of principals.

The canonical AFA of a given language may not be a residual automaton. This claim is supported by Figure 5, since in this case $\mathrm{cl}_{\preccurlyeq^l_L}(c) = \{c\}$ which is not a quotient of L.

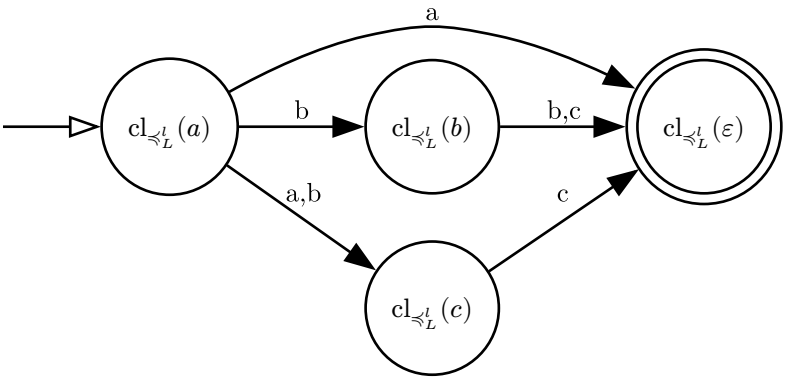


Figure 5: Canonical AFA for language $L = \{a, ac, bb, bc\}$, as described in Definition 15.

Chapter 5.

Effective Quasiorder

In this chapter we first introduce an effective method to use membership queries to compute a left finite index quasiorder through the use of inclusion tables. We will also show some properties of this quasiorder.

5.1. Inclusion Table

In order to define the morphism-based quasiorder, we need to introduce inclusion tables.

Definition 17 (Inclusion table): An inclusion table is a tuple $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$, where L is a regular language and $\mathcal{R}$ and $\mathcal{C}$ are finite sets of words. From a table we can define two functions:

$$\begin{aligned} &\text{row} : \Sigma^* \to \mathcal{C} \to \{+, -\} && \text{col} : \Sigma^* \to \mathcal{R} \to \{+, -\} \\ &\text{row}(w) = c \mapsto wc \in L && \text{col}(w) = r \mapsto rw \in L \end{aligned}$$

Note that the first argument of both row and col can be any word. This allows us to "extend" the table beyond its boundaries in each direction.

Definition 18 (Rows closed on the right): Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$, we say that its rows are closed on the right if and only if

$$\varepsilon \in \mathcal{R} \wedge \forall r \in \mathcal{R}.\ \forall a \in \Sigma.\ \exists r' \in \mathcal{R}.\ \text{row}(ra) = \text{row}(r').$$

A counterexample would be ε if $\varepsilon \notin \mathcal{R}$, or ra such that $r \in \mathcal{R}$, $a \in \Sigma$ and $\nexists r' \in \mathcal{R}.\ \text{row}(ra) = \text{row}(r')$ otherwise.

Definition 19 (Columns closed on the left): Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$, we say that its columns are closed on the left if and only if

$$\varepsilon \in \mathcal{C} \wedge \forall c \in \mathcal{C}.\ \forall a \in \Sigma.\ \exists c' \in \mathcal{C}.\ \text{col}(ac) = \text{col}(c').$$

A counterexample would be ε if $\varepsilon \notin \mathcal{C}$, or ac such that $c \in \mathcal{C}$, $a \in \Sigma$ and $\nexists c' \in \mathcal{C}.\ \text{col}(ac) = \text{col}(c')$ otherwise.

Example 5: The table below has columns closed on the left and rows closed on the right. This means that the row (resp. column) of any word in $\mathcal{R}\Sigma$ (resp. $\Sigma\mathcal{C}$) is a row (resp. column) of the table. For example, $\text{row}(de) = [-----] = \text{row}(aa)$ and $\text{col}(fb) = [+----] = \text{col}(da)$.

L		$\mathcal{C}$				
		ε	a	b	c	da
$\mathcal{R}$	ε	$-$	$-$	$-$	$-$	$+$
	aa	$-$	$-$	$-$	$-$	$-$
	da	$+$	$-$	$-$	$-$	$-$
	d	$-$	$+$	$-$	$-$	$-$
	e	$-$	$-$	$+$	$-$	$-$
	f	$-$	$+$	$+$	$-$	$-$

Table 3: Inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$, with $L = \{ad, af, be, bf, cf, da, eb, fa, fb\}$, $\mathcal{R} = \{\varepsilon, aa, da, d, e, f\}$ and $\mathcal{C} = \{\varepsilon, a, b, c, da\}$.

5.2. Effective Finite Index Quasiorder

In Example 5 we highlighted how when a table has rows and columns closed respectively on the right and left, we can extend its rows and columns by one character and obtain a row or column of the table. By repeating this process, we can map any word to a row or a column of the table.

Definition 20 (Row morphism): Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ whose rows are closed on the right, we can define a function $\tau : \mathcal{R} \times \Sigma \to \mathcal{R}$. We define $\tau(ra) = r'$ such that $r' \in \mathcal{R}$ and $\text{row}(ra) = \text{row}(r')$. We know by Definition 18 that such r' exists. In case of multiple possible values of r', we can just pick the smallest in lexicographical order[1].

We can extend τ to Σ^* by recursively applying it. For clarity, we define a different morphism $\psi : \Sigma^* \to \mathcal{R}$ to avoid overloading τ. We define $\psi(r) = r$ if $r \in \mathcal{R}$, $\psi(wa) = \tau(\psi(w)a)$, otherwise.

Definition 21 (Column morphism): Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ whose columns are closed on the left, we can define a function $\rho : \Sigma \times \mathcal{C} \to \mathcal{C}$. We define $\rho(ac) = c'$ such that $c' \in \mathcal{C}$ and $\text{row}(ac) = \text{row}(c')$. We know by Definition 19 that such c' exists. In case of multiple possible values of c', we can just pick the smallest in lexicographical order[1].

We can extend ρ to Σ^* by recursively applying it. For clarity, we define a different morphism $\varphi : \Sigma^* \to \mathcal{C}$ to avoid overloading ρ. We define $\varphi(c) = c$ if $c \in \mathcal{C}$, $\varphi(aw) = \rho(a\varphi(w))$, otherwise.

Note that both ψ and ψ are idempotent, which means that for any $w \in \Sigma^*$ $\psi(\psi(w)) = \psi(w)$ and $\varphi(\varphi(w)) = \varphi(w)$.

Observe that extending the table using ψ or φ is not the same as extending the table by adding rows or columns. Consider Table 3 as an example, in

[1] Any other criteria is fine, as long as the function is well-defined

this case $\varphi(be) = \rho(b\varphi(e)) = \rho(b\rho(e\varphi(\varepsilon))) = \rho(b\rho(e\varepsilon)) = \rho(bc) = c$, but at the same time $\mathrm{col}(be) = [+-----] \neq \mathrm{col}(\varphi(be)) = \mathrm{col}(c) = [------]$.

Definition 22 (Morphism-based quasiorder): Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ whose columns are closed on the left, we can define for any $u, v \in \Sigma^*$

$$u \preccurlyeq^l_\varphi v \overset{\text{def}}{\iff} \forall r \in \mathcal{R}.\, r\varphi(u) \in L \Rightarrow r\varphi(v) \in L$$

Example 6: The table in this example represents the relation $\preccurlyeq^l_\varphi$ extracted from table $T = (L, \mathcal{R}, \mathcal{C})$ in Example 5, where $L = \{ad, af, be, bf, cf, da, eb, fa, fb\}$, $\mathcal{R} = \{\varepsilon, aa, da, d, e, f\}$ and $\mathcal{C} = \{\varepsilon, a, b, c, da\}$.

$\preccurlyeq^l_\varphi$	ε	a	b	c	da
ε	+	−	−	−	−
a	−	+	−	−	−
b	−	−	+	−	−
c	+	+	+	+	+
da	−	−	−	−	+

Note that in general it is not true that $\preccurlyeq^l_\varphi$ is coarser or finer than $\preccurlyeq^l_L$. In this example, we have $c \preccurlyeq^l_\varphi be$ while $c \not\preccurlyeq^l_L be$. Also, $da \preccurlyeq^l_L be$ while $da \not\preccurlyeq^l_\varphi be$.

A fundamental characteristic of $\preccurlyeq^l_\varphi$ is being a left finite index quasiorder. This will enable the possibility of using $\preccurlyeq^l_\varphi$ for automata construction. This property is proven in Theorem 23. Lemma 24 and Theorem 25 prove another fundamental property of $\preccurlyeq^l_\varphi$ – if $\preccurlyeq^l_\varphi$ is L-preserving, then it is equal to the Nerode's left quasiorder for L.

Theorem 23 ($\preccurlyeq^l_\varphi$ is a left finite index quasiorder): Let $\mathcal{T}$ be an inclusion table with columns closed on the left, $\preccurlyeq^l_\varphi$ is a finite index quasiorder. Furthermore, if $\mathcal{T}$'s rows are closed on the right, then $\preccurlyeq^l_\varphi$ is left.

Proof of Theorem 23: Proving that $\preccurlyeq^l_\varphi$ is a quasiorder is straightforward, as both reflexivity and transitivity of $\preccurlyeq^l_\varphi$ come from the reflexivity and transitivity of the implication.

Indeed $\preccurlyeq^l_\varphi$ is also a finite index quasiorder, since for any word $w \in \Sigma^*$ we have $w \sim^l_\varphi \varphi(w)$ and $\varphi(w)$ can have at most $|\mathcal{C}|$ distinct values.

Finally, we prove that if $\mathcal{T}$'s columns are closed on the right, $\preccurlyeq^l_\varphi$ is left. Assume that $u \preccurlyeq^l_\varphi v$ for some $u, v \in \Sigma^*$, we prove that $au \preccurlyeq^l_\varphi av$ for any $a \in \Sigma$.

$$u \preccurlyeq^l_\varphi v \Leftrightarrow$$
$$\forall r \in \mathcal{R}.\, r\varphi(u) \in L \Rightarrow r\varphi(v) \in L \Rightarrow$$

$$\begin{aligned}
&\forall r \in \mathcal{R}.\, \tau(ra)\varphi(u) \in L \Rightarrow \tau(ra)\varphi(v) \in L \Leftrightarrow \\
&\text{[By definition of } \tau\text{, since the rows are closed on the right]} \\
&\forall r \in \mathcal{R}.\, ra\varphi(u) \in L \Rightarrow ra\varphi(v) \in L \Leftrightarrow \\
&\text{[By definition of } \rho\text{, since the columns are closed on the left]} \\
&\forall r \in \mathcal{R}.\, r\rho(a\varphi(u)) \in L \Rightarrow r\rho(a\varphi(v)) \in L \Leftrightarrow \\
&\forall r \in \mathcal{R}.\, r\varphi(au) \in L \Rightarrow r\varphi(av) \in L \Leftrightarrow \\
&au \preccurlyeq^l_\varphi av.
\end{aligned}$$

■

Lemma 24: Let $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ be an inclusion table. If $\mathcal{T}$'s columns are closed on the left and $\mathcal{T}$'s rows are both closed on the right and prefix-closed, assuming that $\preccurlyeq^l_\varphi$ is L-preserving, then

$$\preccurlyeq^l_L \quad \subseteq \quad \preccurlyeq^l_\varphi .$$

Proof of Lemma 24: Observe that since $\preccurlyeq^l_\varphi$ is L-preserving and $\forall u \in \Sigma^*.\, u \sim^l_\varphi \varphi(u)$, then $u \in L \Leftrightarrow \varphi(u) \in L$. Also observe that $\preccurlyeq^l_\varphi$ is a left quasiorder by Theorem 23.

We want to show that for any $u, v \in \Sigma^*$, if $\forall w \in \Sigma^*.\, wu \in L \Rightarrow wv \in L$ then $\forall r \in \mathcal{R}.\, r\varphi(u) \in L \Rightarrow r\varphi(v) \in L$. We proceed by induction on the length of r.

Base case: Assume $r = \varepsilon$.

$$\begin{aligned}
&\varphi(u) \in L \Leftrightarrow \\
&\quad u \in L \Rightarrow \text{[Since } u \preccurlyeq^l_L v\text{]} \\
&\quad v \in L \Leftrightarrow \\
&\varphi(v) \in L.
\end{aligned}$$

Inductive step: Assume $r = xa$. Since x is shorter than r and must be in $\mathcal{R}$, since it is prefix-closed, we can assume inductively that $\forall u, v \in \Sigma^*.\, u \preccurlyeq^l_L v \Rightarrow (x\varphi(u) \in L \Rightarrow x\varphi(v) \in L)$.

$$\begin{aligned}
xa\varphi(u) \in L &\Leftrightarrow \text{[Becasue } \mathcal{C} \text{ is closed on the left]} \\
x\rho(a\varphi(u)) \in L &\Leftrightarrow \text{[By Definition 20]} \\
x\varphi(au) \in L &\Rightarrow \text{[By inductive hypotesis, since } au \preccurlyeq^l_L av\text{]} \\
x\varphi(av) \in L &\Leftrightarrow \text{[By Definition 20]} \\
x\rho(a\varphi(v)) &\Leftrightarrow \text{[Becasue } \mathcal{C} \text{ is closed on the left]} \\
xa\varphi(v) \in L.&
\end{aligned}$$

■

Theorem 25 ($\preccurlyeq^l_\varphi$ is equivalent to $\preccurlyeq^l_L$): Let $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ be an inclusion table. If $\mathcal{T}$'s columns are closed on the left and $\mathcal{T}$'s rows are both closed on the right and prefix-closed, assuming that $\preccurlyeq^l_\varphi$ is L-preserving, then

$$\forall u, v \in \Sigma^*.\ u \preccurlyeq^l_L v \Leftrightarrow u \preccurlyeq^l_\varphi v.$$

Proof of Theorem 25: The proof follows immediately from Lemma 24 and the fact that Nerode's left quasiorder is the coarsest left L-preserving quasiorder [LV94]. ■

Chapter 6.

Effective Automata Construction

In this chapter we introduce an abstraction to represent principals of finite index quasiorders. We also show how to use this abstraction to compute a principal automaton. Finally, we prove some properties of automata built in this way.

6.1. Principal Abstraction

Definition 26 (Base of $\sim^l$): Let $\sim^l$ be a left equivalence on words. We say that a finite set P is a base of $\sim^l$ if and only if

$$\varepsilon \in P \wedge \forall p \in P.\ \forall a \in \Sigma.\ \exists p'.\ ap \sim^l p'.$$

We can thus define a function $\mu : \Sigma P \setminus P \to P$ such that $\mu(ap) \sim^l ap$ for any $a \in \Sigma$, $p \in P$.

Lemma 27: Given a base of $\sim^l$ P, define the function $\theta : \Sigma^* \to P$ as $\theta(p) = p$ if $p \in P$, $\theta(aw) = \mu(a\theta(w))$ otherwise, then

$$\forall w \in \Sigma^*.\ w \sim^l \theta(w).$$

Proof of Lemma 27: Let's prove that $\forall w \in \Sigma^*.\ w \sim^l \theta(w)$ by induction.

Base case: Assume $w = \varepsilon$.

$$\begin{aligned} P \text{ is } \sim^l \text{-closed} &\Rightarrow \\ \varepsilon \in P &\Rightarrow \\ \theta(\varepsilon) = \varepsilon &\Rightarrow \\ \theta(\varepsilon) \sim^l \varepsilon. & \end{aligned}$$

Inductive case: Assume $w = au$ and, by inductive hypothesis, $\theta(u) \sim^l u$. If $w \in P$, then $\theta(w) = w \Rightarrow \theta(w) \sim^l w$. Otherwise, if $w \notin P$:

$$\begin{aligned} \theta(au) &= \\ \mu(a\theta(u)) &\sim^l \quad \text{[By Definition 26]} \\ a\theta(u) &\sim^l \quad \text{[Since } \theta(u) \sim^l u \text{ and } \sim^l \text{ is a left equivalence]} \\ au & \end{aligned}$$

■

Example 7: Consider the quasiorder defined in Example 6. We will refer to it as $\preccurlyeq^l_\varphi$ and denote $\sim^l_\varphi = \preccurlyeq^l_\varphi \cap \left(\preccurlyeq^l_\varphi\right)^{-1}$. A base of $\sim^l_\varphi$ is $\{\varepsilon, a, b, c, da\}$. This is true because for any word $w \in \Sigma^*$, by definition of $\preccurlyeq^l_\varphi$, $w \sim^l_\varphi \varphi(w)$ and, by definition of φ, $\varphi(w) \in \{\varepsilon, a, b, c, da\}$.

Observe that the cardinality of a base of a quasiorder is an upper bound for the number of equivalence classes of the quasiorder. Furthermore, every equivalence class has at least one member in the base.

Definition 28 (Principal abstraction): Given a left quasiorder $\preccurlyeq^l$, let P be a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$.

For any $p \in P$, we define $[\![p]\!] : P \to \{+, -\}$ as $[\![p]\!] = p' \mapsto p \preccurlyeq^l p'$.

Indeed, from the property of θ follows that $\forall aw \in \Sigma^* \setminus P.\ [\![aw]\!] = p' \mapsto aw \preccurlyeq^l p'$.

We define $[\![u]\!] \sqsubseteq [\![v]\!] \overset{\text{def}}{\iff} \forall p \in P.\ [\![u]\!]p \Rightarrow [\![v]\!]p$.

We define $[\![u]\!] \sqcap [\![v]\!] : P \to \{+, -\}$ as $([\![u]\!] \sqcap [\![v]\!])p = [\![u]\!]p \wedge [\![v]\!]p$.

Let $\mathrm{Sup}(u) = \{v \mid v \in P, [\![u]\!] \sqsubset [\![v]\!]\}$, we say that $[\![u]\!]$ is $[\![\cdot]\!]$-composite if and only if

$$\mathrm{Sup}(u) \neq \emptyset \wedge [\![u]\!] = \bigsqcap_{v \in \mathrm{Sup}(u)} [\![v]\!].$$

A principal is $[\![\cdot]\!]$-prime if and only if it is not $[\![\cdot]\!]$-composite.

Now that we have an abstraction for principals, we need to prove its soundness. Lemma 29, Corollary 30, Lemma 31 and Lemma 32 prove it.

Lemma 29: Given a left quasiorder $\preccurlyeq^l$, let P be a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$, then

$$\forall u, v \in \Sigma^*.\ \mathrm{cl}_{\preccurlyeq^l}(u) \subseteq \mathrm{cl}_{\preccurlyeq^l}(v) \Leftrightarrow [\![u]\!] \sqsubseteq [\![v]\!].$$

Proof of Lemma 29:

$$\begin{aligned}
&\mathrm{cl}_{\preccurlyeq^l}(u) \subseteq \mathrm{cl}_{\preccurlyeq^l}(v) \Leftrightarrow \\
&\forall w \in \Sigma^*.\ u \preccurlyeq^l w \Rightarrow v \preccurlyeq^l w \Leftrightarrow \text{[By Lemma 27]} \\
&\forall w \in \Sigma^*.\ u \preccurlyeq^l \theta(w) \Rightarrow v \preccurlyeq^l \theta(w) \Leftrightarrow [\Rightarrow \text{ holds becausee } \forall p \in P.\ \theta(p) = p] \\
&\forall p \in P.\ u \preccurlyeq^l p \Rightarrow v \preccurlyeq^l p \Leftrightarrow \\
&[\![u]\!] \sqsubseteq [\![v]\!].
\end{aligned}$$

■

Corollary 30: Given a left finite index quasiorder $\preccurlyeq^l$, let P be a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$, then

$$\forall u, v \in \Sigma^*.\ u \sim^l v \Leftrightarrow \mathrm{cl}_{\preccurlyeq^l}(u) = \mathrm{cl}_{\preccurlyeq^l}(v) \Leftrightarrow [\![u]\!] = [\![v]\!].$$

Proof of Corollary 30: The proof follows immediately from Lemma 29. ■

Lemma 31: Given a left finite index quasiorder $\preccurlyeq^l$, let P be a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$, then

$$\forall u, v, w \in \Sigma^*.\ \mathrm{cl}_{\preccurlyeq^l}(u) = \mathrm{cl}_{\preccurlyeq^l}(v) \cap \mathrm{cl}_{\preccurlyeq^l}(w) \Leftrightarrow [\![u]\!] = [\![v]\!] \sqcap [\![w]\!].$$

Proof of Lemma 31:

$$\begin{gathered}
\mathrm{cl}_{\preccurlyeq^l}(u) = \mathrm{cl}_{\preccurlyeq^l}(v) \cap \mathrm{cl}_{\preccurlyeq^l}(w) \Leftrightarrow \\
\forall x \in \Sigma^*.\ u \preccurlyeq^l x \Leftrightarrow v \preccurlyeq^l x \wedge w \preccurlyeq^l x \Leftrightarrow \text{[By Lemma 27]} \\
\forall x \in \Sigma^*.\ u \preccurlyeq^l \theta(x) \Leftrightarrow v \preccurlyeq^l \theta(x) \wedge w \preccurlyeq^l \theta(x) \Leftrightarrow [\Rightarrow \text{holds because } \forall p \in P.\ \theta(p) = p] \\
\forall p \in P.\ u \preccurlyeq^l p \Leftrightarrow v \preccurlyeq^l p \wedge w \preccurlyeq^l p \Leftrightarrow \\
[\![u]\!] = [\![v]\!] \sqcap [\![w]\!].
\end{gathered}$$

■

Lemma 32: Given a left quasiorder $\preccurlyeq^l$, let P be a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$, then

$$\forall u \in \Sigma^*.\ \mathrm{cl}_{\preccurlyeq^l}(u) \text{ is } \mathrm{cl}_{\preccurlyeq^l}\text{-prime} \Leftrightarrow [\![u]\!] \text{ is } [\![\cdot]\!]\text{-prime}.$$

Proof of Lemma 32: Obverse that the statement is equivalent to $\mathrm{cl}_{\preccurlyeq^l}(u)$ is $\mathrm{cl}_{\preccurlyeq^l}$-composite $\Leftrightarrow$ $[\![u]\!]$ is $[\![\cdot]\!]$-composite, for any $u \in \Sigma^*$.

$$\begin{gathered}
\mathrm{cl}_{\preccurlyeq^l}(u) \text{ is } \mathrm{cl}_{\preccurlyeq^l}\text{-composite} \Leftrightarrow \\
\left(\exists x \in \Sigma^*.\ \mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(x)\right) \wedge \mathrm{cl}_{\preccurlyeq^l}(u) = \bigcap_{\substack{x \in \Sigma^* \\ \mathrm{cl}_{\preccurlyeq^l}(x) \supset \mathrm{cl}_{\preccurlyeq^l}(u)}} \mathrm{cl}_{\preccurlyeq^l}(x) \Leftrightarrow \text{[Since } P \text{ is } \sim^l\text{-closed]} \\
\left(\exists x \in P.\ \mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(x)\right) \wedge \mathrm{cl}_{\preccurlyeq^l}(u) = \bigcap_{\substack{x \in P \\ \mathrm{cl}_{\preccurlyeq^l}(u) \supset \mathrm{cl}_{\preccurlyeq^l}(x)}} \mathrm{cl}_{\preccurlyeq^l}(x) \Leftrightarrow \text{[By Lemma 29]} \\
\left(\exists x \in P.\ [\![u]\!] \sqsubset [\![x]\!]\right) \wedge \mathrm{cl}_{\preccurlyeq^l}(u) = \bigcap_{\substack{x \in P \\ [\![u]\!] \sqsupset [\![x]\!]}} \mathrm{cl}_{\preccurlyeq^l}(x) \Leftrightarrow \text{[By Lemma 31]} \\
\left(\exists x \in P.\ [\![u]\!] \sqsubset [\![x]\!]\right) \wedge [\![u]\!] = \bigsqcap_{\substack{x \in P \\ [\![u]\!] \sqsupset [\![x]\!]}} [\![x]\!] \Leftrightarrow \\
[\![u]\!] \text{ is } [\![\cdot]\!]\text{-composite}
\end{gathered}$$

■

Lemma 33, which retrace Lemma 11, is fundamental as it shows that, under some conditions, when considering the intersection of principal abstractions we can consider only prime principal abstractions. Furthermore, it will greatly simplify many proofs.

Lemma 33: Given a left quasiorder $\preccurlyeq^l$, let P be a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$, then for any $u \in P$

$$\bigsqcap_{[\![u]\!] \sqsubset [\![v]\!] \in Q} [\![v]\!] = \bigsqcap_{[\![u]\!] \sqsubset [\![v]\!]} [\![v]\!]$$

where $Q = \{[\![u]\!] \mid u \in P, [\![u]\!] \text{ prime}\}$.

Proof of Lemma 33:

$$\bigsqcap_{[\![u]\!] \sqsubset [\![v]\!] \in Q} [\![v]\!] = \bigsqcap_{[\![u]\!] \sqsubset [\![v]\!]} [\![v]\!] \Leftrightarrow$$

$$\bigsqcap_{\substack{[\![u]\!] \sqsubset [\![v]\!] \\ [\![v]\!] \text{ prime}}} [\![v]\!] = \bigsqcap_{[\![u]\!] \sqsubset [\![v]\!]} [\![v]\!] \Leftrightarrow \text{[By Lemma 29]}$$

$$\bigsqcap_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(v) \\ [\![v]\!] \text{ prime}}} [\![v]\!] = \bigsqcap_{\mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(v)} [\![v]\!] \Leftrightarrow \text{[By Lemma 32]}$$

$$\bigsqcap_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(v) \\ \mathrm{cl}_{\preccurlyeq^l}(v) \text{ prime}}} [\![v]\!] = \bigsqcap_{\mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(v)} [\![v]\!] \Leftrightarrow \text{[By Lemma 31]}$$

$$\bigcap_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(v) \\ \mathrm{cl}_{\preccurlyeq^l}(v) \text{ prime}}} \mathrm{cl}_{\preccurlyeq^l}(v) = \bigcap_{\mathrm{cl}_{\preccurlyeq^l}(u) \subset \mathrm{cl}_{\preccurlyeq^l}(v)} \mathrm{cl}_{\preccurlyeq^l}(v)$$

which holds by Lemma 11. ∎

6.2. Effective Automata Construction

Finally, we define an automata construction resembling Definition 10. The main difference is that in this case, we use abstractions instead of principals and the set of states does not depend on any word in Σ^* but only on a finite set of words.

Definition 34 (AFA construction $\mathrm{T}\left(L, \preccurlyeq^l, P\right)$): Given a language $L \subseteq \Sigma^*$, a left finite index quasiorder $\preccurlyeq^l$ and a base of $\sim^l$ P, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$. We define the AFA $\mathrm{T}\left(L, \preccurlyeq^l, P\right) = (\Sigma, Q, I, F, \delta)$ where

$$Q = \{[\![u]\!] \mid u \in P, [\![u]\!] \text{ is } [\![\cdot]\!]\text{-prime}\};$$

$$I = \left(\bigvee_{\substack{[\![u]\!] \in Q \\ u \in L}} [\![u]\!] \right) \vee \left(\bigvee_{\substack{[\![u]\!] \notin Q \\ u \in L}} \bigwedge_{\substack{[\![z]\!] \in Q \\ z \prec^l u}} [\![z]\!] \right);$$

$$F = \{[\![u]\!] \in Q \mid u \preccurlyeq^l \varepsilon\}.$$

Let $P_u^a = \{[\![v]\!] \mid v \in P, \forall p \in P.\ v \preccurlyeq^l p \Rightarrow u \preccurlyeq^l ap\}$. For any $[\![u]\!] \in Q$, $a \in \Sigma$

$$\delta([\![u]\!], a) = \left(\bigvee_{[\![v]\!] \in P_u^a \cap Q} [\![v]\!] \right) \vee \left(\bigvee_{[\![v]\!] \in P_u^a \setminus Q} \bigwedge_{\substack{[\![z]\!] \in Q \\ z \prec^l v}} [\![z]\!] \right).$$

The automaton is well-defined because $\preccurlyeq^l$ is a finite index quasiorder, therefore$P$ is finite, which means that also Q must be finite.

Example 8: Consider the language $L = \{ad, af, be, bf, cf, da, eb, fa, fb\}$ and the quasiorder $\preccurlyeq^l_\varphi$ defined in Example 6. Denote $\sim^l_\varphi = \preccurlyeq^l_\varphi \cap \left(\preccurlyeq^l_\varphi\right)^{-1}$. From Example 7 we know that $P = \{\varepsilon, a, b, c, da\}$ is a base of $\sim^l_\varphi$, thus we can construct the automata $\mathrm{T}\left(L, \preccurlyeq^l_\varphi, P\right)$. We therefore have $Q = \{[\![\varepsilon]\!], [\![a]\!], [\![b]\!], [\![c]\!], [\![da]\!]\}$, $I = [\![da]\!]$ and $F = \{[\![\varepsilon]\!], [\![c]\!]\}$. The full automaton is depicted in Figure 6.

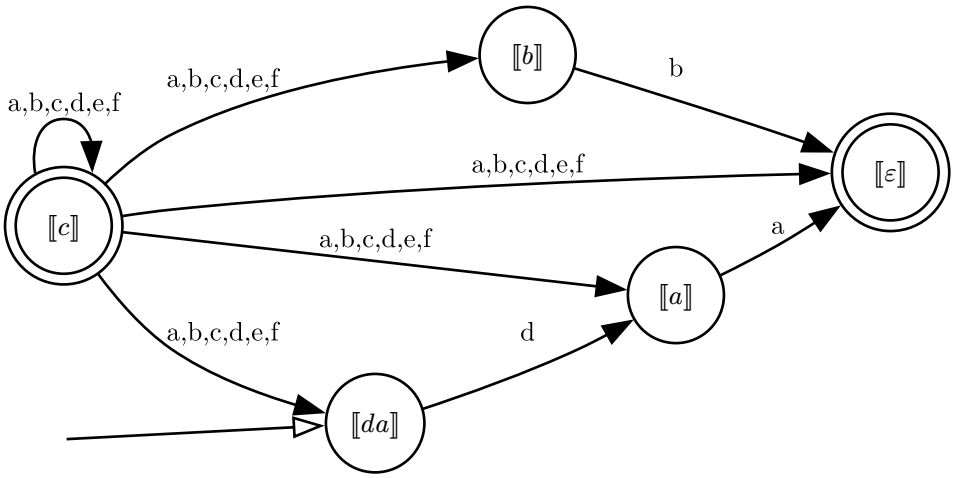


Figure 6: Automaton $\mathrm{T}\left(L, \preccurlyeq^l_\varphi, P\right)$

As we did for automata construction A (Theorem 12), in Theorem 35 we prove that $\mathrm{T}\left(L, \preccurlyeq^l, P\right)$ is principal. Similarly to Theorem 14, in Theorem 37 we prove that $T\left(L, \preccurlyeq^l, P\right)$ recognize the correct language if and only if $\preccurlyeq^l$ is L-preserving. The similarity between automata constructions A and T culminate in Theorem 38, in which we prove that $\mathrm{T}\left(L, \preccurlyeq^l, P\right)$ and $\mathrm{A}\left(L, \preccurlyeq^l\right)$ are isomorphic if $\preccurlyeq^l$ is L-preserving.

Theorem 35 ($\mathrm{T}\left(L, \preccurlyeq^l, P\right)$ is principal): Given a language $L \subseteq \Sigma^*$, a left finite index quasiorder $\preccurlyeq^l$ and a base of $\sim^l$ P, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$. Let $\mathrm{T}\left(L, \preccurlyeq^l, P\right) = (\Sigma, Q, I, F, \delta)$, then

$$\forall [\![u]\!] \in Q.\, W_{[\![u]\!],F} = \mathrm{cl}_{\preccurlyeq^l}(u).$$

Proof of Theorem 35: We prove that $\forall w \in \Sigma^*.\, w \in W_{[\![u]\!],F} \Leftrightarrow w \in \mathrm{cl}_{\preccurlyeq^l}(u)$ by induction on the length of w.

Base case: Assume $w = \varepsilon$, then for any $[\![u]\!] \in Q$

$$\begin{aligned} \varepsilon \in W_{[\![u]\!],F} &\Leftrightarrow \\ [\![u]\!] \in F &\Leftrightarrow \\ u \preccurlyeq^l \varepsilon &\Leftrightarrow \\ \varepsilon \in \mathrm{cl}_{\preccurlyeq^l}(u). & \end{aligned}$$

Inductive case: Assume $w = ax$, with $a \in \Sigma$ and $x \in \Sigma^*$. By inductive hypothesis assume that $\forall [\![v]\!] \in Q.\, W_{[\![v]\!],F} = \mathrm{cl}_{\preccurlyeq^l}(v)$, then for any $[\![u]\!] \in Q$

$$\begin{aligned} ax \in W_{[\![u]\!],F} &\Leftrightarrow \\ x \in W_{\delta([\![u]\!],a),F} &\Leftrightarrow \\ x \in \bigcup_{[\![v]\!] \in P_u^a \cap Q} W_{[\![v]\!],F} \cup \bigcup_{[\![v]\!] \in P_u^a \setminus Q} \bigcap_{\substack{[\![z]\!] \in Q \\ [\![v]\!] \sqsubset [\![z]\!]}} W_{[\![z]\!],F} &\Leftrightarrow \\ \text{[By inductive hypotesis]} & \\ x \in \bigcup_{[\![v]\!] \in P_u^a \cap Q} \mathrm{cl}_{\preccurlyeq^l}(v) \cup \bigcup_{[\![v]\!] \in P_u^a \setminus Q} \bigcap_{\substack{[\![z]\!] \in Q \\ [\![v]\!] \sqsubset [\![z]\!]}} \mathrm{cl}_{\preccurlyeq^l}(z) &\Leftrightarrow \\ \exists [\![v]\!] \in P_u^a.\, \begin{cases} v \preccurlyeq^l x & \text{if } [\![v]\!] \in Q \\ \forall [\![z]\!] \in Q.\, z \prec^l v \Rightarrow z \preccurlyeq^l x & \text{if } [\![v]\!] \notin Q \end{cases} &\Leftrightarrow \\ \text{[By Definition 9]} & \\ \exists [\![v]\!] \in P_u^a.\, \begin{cases} v \preccurlyeq^l x & \text{if } [\![v]\!] \in Q \\ \forall [\![z]\!] \in Q.\, z \prec^l v \Rightarrow z \preccurlyeq^l x & \text{if } [\![v]\!] \notin Q \end{cases} &\Leftrightarrow \\ \text{[By Lemma 33]} & \\ \exists [\![v]\!] \in P_u^a.\, \begin{cases} v \preccurlyeq^l x \text{ if } [\![v]\!] \in Q \\ v \preccurlyeq^l x \text{ if } [\![v]\!] \notin Q \end{cases} &\Leftrightarrow \\ \exists [\![v]\!] \in P_u^a.\, v \preccurlyeq^l x &\Leftrightarrow \\ \exists v \in P.\, \big(\forall p \in P.\, v \preccurlyeq^l p \Rightarrow u \preccurlyeq^l ap\big) \wedge v \preccurlyeq^l x &\Leftrightarrow \\ \text{[The proof of this passage is given immediatly below]} & \\ u \preccurlyeq^l ax & \end{aligned}$$

To conclude the proof of the inductive case we show that

$$\exists v \in P.\, \big(\forall p \in P.\, v \preccurlyeq^l p \Rightarrow u \preccurlyeq^l ap\big) \wedge v \preccurlyeq^l x \Leftrightarrow u \preccurlyeq^l ax.$$

To prove $\Rightarrow$, we choose $p = \theta(x)$, with θ as defined in Lemma 27, therefore

$$\begin{aligned} v \preccurlyeq^l \theta(x) &\Rightarrow [\text{Because } v \preccurlyeq^l p \Rightarrow u \preccurlyeq^l ap] \\ u \preccurlyeq^l a\theta(x) &\Leftrightarrow \\ u \preccurlyeq^l \mu(a\theta(x)) &\Leftrightarrow \\ u \preccurlyeq^l \theta(ax) &\Leftrightarrow \\ u \preccurlyeq^l ax. & \end{aligned}$$

to prove $\Leftarrow$, we choose $v = \theta(x)$, therefore $v \preccurlyeq^l x$ holds by reflexivity and Lemma 27. To prove the left part of the conjunction observe that, for any $p \in P$

$$\begin{aligned} \theta(x) \preccurlyeq^l p &\Leftrightarrow \\ x \preccurlyeq^l p &\Rightarrow [\text{Because } \preccurlyeq^l \text{ is a left quasiorder}] \\ ax \preccurlyeq^l ap &\Rightarrow [\text{By transitivity, since } u \preccurlyeq^l ax] \\ u \preccurlyeq^l ap. & \end{aligned}$$

■

Lemma 36: Given a language $L \subseteq \Sigma^*$, a left finite index quasiorder $\preccurlyeq^l$ and a base of $\sim^l$ P, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$. Let $\mathrm{T}\big(L, \preccurlyeq^l, P\big) = (\Sigma, Q, I, F, \delta)$, then

$$\mathcal{L}\big(\mathrm{T}\big(L, \preccurlyeq^l, P\big)\big) = \mathrm{cl}_{\preccurlyeq^l}(L \cap P)$$

Proof of Lemma 36:

$$\begin{aligned} \mathcal{L}\big(\mathrm{T}\big(L, \preccurlyeq^l, P\big)\big) &= [\text{By Definition 4}] \\ W_{I,F} &= [\text{By Definition 34}] \\ \left(\bigcup_{\substack{[\![u]\!]\in Q \\ u\in L}} W_{u,F}\right) \cup \left(\bigcup_{\substack{[\![u]\!]\notin Q \\ u\in L}} \bigcap_{\substack{[\![z]\!]\in Q \\ z\prec^l u}} W_{z,F}\right) &= [\text{By Theorem 35}] \\ \left(\bigcup_{\substack{[\![u]\!]\in Q \\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \cup \left(\bigcup_{\substack{[\![u]\!]\notin Q \\ u\in L}} \bigcap_{\substack{[\![z]\!]\in Q \\ z\prec^l u}} \mathrm{cl}_{\preccurlyeq^l}(z)\right) &= [\text{By Lemma 33}] \\ \left(\bigcup_{\substack{[\![u]\!]\in Q \\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \cup \left(\bigcup_{\substack{[\![u]\!]\notin Q \\ u\in L}} \bigcap_{z\prec^l u} \mathrm{cl}_{\preccurlyeq^l}(z)\right) &= [\text{By Definition 9 and Lemma 32}] \\ \left(\bigcup_{\substack{[\![u]\!]\in Q \\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \cup \left(\bigcup_{\substack{[\![u]\!]\notin Q \\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) &= [u \text{ must be in } P \text{ for } [\![u]\!] \text{ to be well-defined}] \end{aligned}$$

$$\bigcup_{u \in L \cap P} \mathrm{cl}_{\preccurlyeq^l}(u) = \mathrm{cl}_{\preccurlyeq^l}(L \cap P).$$

■

Theorem 37 (Correctness of $\mathrm{T}\big(L, \preccurlyeq^l, P\big)$): Given a language $L \subseteq \Sigma^*$, a left finite index quasiorder $\preccurlyeq^l$ and a base of $\sim^l$ P, where $\sim^l = \preccurlyeq^l \cap \big(\preccurlyeq^l\big)^{-1}$, then

$$\mathcal{L}\big(\mathrm{T}\big(L, \preccurlyeq^l, P\big)\big) = L \quad \Leftrightarrow \quad \preccurlyeq^l \text{ is } L\text{-preserving.}$$

Proof of Theorem 37: Let $\mathrm{T}\big(L, \preccurlyeq^l, P\big) = (\Sigma, Q, I, F, \delta)$.

We easily prove that if $\preccurlyeq^l$ is L-preserving, then $\mathcal{L}\big(\mathrm{T}\big(L, \preccurlyeq^l, P\big)\big) = L$ thanks to Lemma 36.

Then we prove that if $\preccurlyeq^l$ is not L-preserving, then $\mathcal{L}\big(\mathrm{T}\big(L, \preccurlyeq^l, P\big)\big) \neq L$. $\preccurlyeq^l$ being not preserving means that $\exists u, v \in \Sigma^*.\ u \in L \wedge u \preccurlyeq^l v \wedge v \notin L$. Now there are three cases:

$\theta(u) \in L$: In this case, we have that $\theta(u) \in L \cap P$ and $v \in \mathrm{cl}_{\preccurlyeq^l}(\theta(u))$, therefore $v \in \mathcal{L}\big(T\big(L, \preccurlyeq^l, P\big)\big)$ from Lemma 36.

$\theta(u) \notin L \wedge \exists p \in P \cap L.\ p \preccurlyeq^l u$: In this case, we have $\theta(u) \in \mathrm{cl}_{\preccurlyeq^l}(p)$, therefore $\theta(u) \in \mathcal{L}\big(T\big(L, \preccurlyeq^l, P\big)\big)$ from Lemma 36.

$\theta(u) \notin L \wedge \nexists p \in P \cap L.\ p \preccurlyeq^l u$: In this case, we have that $\nexists p \in L \cap P.\ u \in \mathrm{cl}_{\preccurlyeq^l}(p)$, therefore $u \notin \mathcal{L}\big(T\big(L, \preccurlyeq^l, P\big)\big)$ from Lemma 36.

■

Theorem 38 ($\mathrm{T}\big(L, \preccurlyeq^l, P\big)$ and $\mathrm{A}\big(L, \preccurlyeq^l\big)$ are isomorphic): Given a language $L \subseteq \Sigma^*$, a left L-preserving finite index quasiorder $\preccurlyeq^l$ and a base of $\sim^l$ P, where $\sim^l = \preccurlyeq^l \cap \big(\preccurlyeq^l\big)^{-1}$, then $T = \mathrm{T}\big(L, \preccurlyeq^l, P\big) = \big(\Sigma, Q^T, I^T, F^T, \delta^T\big)$ and $A = \mathrm{A}\big(L, \preccurlyeq^l\big) = \big(\Sigma, Q^A, I^A, F^A, \delta^A\big)$ are isomorphic.

Proof of Theorem 38: Let's consider a mapping γ, defined as $\gamma(\llbracket u \rrbracket) = \mathrm{cl}_{\preccurlyeq^l}(u)$ for any $\llbracket u \rrbracket \in Q^T$. We are going to prove that γ is an isomorphism between T and A.

When considering $\mathrm{cl}_{\preccurlyeq^l}(u) \in Q^A$, we can assume w.l.o.g. that $u \in P$, since $\forall w \in \Sigma^*.\ \exists p \in P.\ w \sim^l p$ and, therefore, $\forall w \in \Sigma^*.\ \exists p \in P.\ \mathrm{cl}_{\preccurlyeq^l}(w) = \mathrm{cl}_{\preccurlyeq^l}(p)$. The same applies to members of Q^a_u.

We start by showing that $\llbracket u \rrbracket \in Q^T \Leftrightarrow \gamma(\llbracket u \rrbracket) \in Q^A$, for any $u \in P$.

$$\begin{aligned} &\llbracket u \rrbracket \in Q^T \Leftrightarrow \\ &\llbracket u \rrbracket \text{ prime} \Leftrightarrow \text{[By Lemma 32]} \\ &\mathrm{cl}_{\preccurlyeq^l}(u) \text{ prime} \Leftrightarrow \end{aligned}$$

$$\mathrm{cl}_{\preccurlyeq^l}(u) \in Q^A \Leftrightarrow$$
$$\gamma([\![u]\!]) \in Q^A.$$

Then we prove that $\gamma(I^T) = I^A$.

$$\gamma(I^T) =$$
$$\gamma\left(\left(\bigvee_{\substack{[\![u]\!]\in Q^T \\ u\in L}} [\![u]\!]\right) \vee \left(\bigvee_{\substack{[\![u]\!]\notin Q^T \\ u\in L}} \bigwedge_{\substack{[\![z]\!]\in Q^T \\ z\prec^l u}} [\![z]\!]\right)\right) =$$
$$\gamma\left(\bigvee_{\substack{[\![u]\!]\in Q^T \\ u\in L}} [\![u]\!]\right) \vee \gamma\left(\bigvee_{\substack{[\![u]\!]\notin Q^T \\ u\in L}} \bigwedge_{\substack{[\![z]\!]\in Q^T \\ z\prec^l u}} [\![z]\!]\right) = [\text{Since } \gamma(Q^T) = Q^A]$$
$$\left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q^A \\ u\in L}} \gamma([\![u]\!])\right) \vee \left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q^A \\ u\in L}} \gamma\left(\bigwedge_{\substack{[\![z]\!]\in Q^T \\ z\prec^l u}} [\![z]\!]\right)\right) = [\text{Since } \gamma(Q^T) = Q^A]$$
$$\left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q^A \\ u\in L}} \gamma([\![u]\!])\right) \vee \left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q^A \\ u\in L}} \bigwedge_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z)\in Q^A \\ z\prec^l u}} \gamma([\![z]\!])\right) =$$
$$\left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\in Q^A \\ u\in L}} \mathrm{cl}_{\preccurlyeq^l}(u)\right) \vee \left(\bigvee_{\substack{\mathrm{cl}_{\preccurlyeq^l}(u)\notin Q^A \\ u\in L}} \bigwedge_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z)\in Q^A \\ z\prec^l u}} \mathrm{cl}_{\preccurlyeq^l}(z)\right) =$$
$$I^A.$$

Note about the last passage: we assumed $u \in P$ but this condition does not appear in the definition of I^A. This is fine because $\preccurlyeq^l$ is L-preserving, therefore $\forall w \in L.\ \exists u \in L \cap P.\ \mathrm{cl}_{\preccurlyeq^l}(w) = \mathrm{cl}_{\preccurlyeq^l}(u)$. Namely, $u = \theta(w)$.

Then we prove that $[\![u]\!] \in F^T \Leftrightarrow \gamma([\![u]\!]) \in F^A$, for any $u \in P$.

$$[\![u]\!] \in F^T \Leftrightarrow u \preccurlyeq^l \varepsilon \Leftrightarrow \mathrm{cl}_{\preccurlyeq^l}(u) \in F^A \Leftrightarrow \gamma([\![u]\!]) \in F^A.$$

Then we prove that for any $u, v \in P$, $[\![v]\!] \in P^a_u \Leftrightarrow \gamma([\![v]\!]) \in Q^a_u$, using the function θ defined in Lemma 27.

$$[\![v]\!] \in P^a_u \Leftrightarrow$$
$$\forall p \in P.\ v \preccurlyeq^l p \Rightarrow u \preccurlyeq^l ap \Leftrightarrow [\Leftarrow \text{ holds because } \forall p \in P.\ \theta(p) = p]$$
$$\forall w \in \Sigma^*.\ v \preccurlyeq^l \theta(w) \Rightarrow u \preccurlyeq^l a\theta(w) \Leftrightarrow [\text{By Lemma 27}]$$
$$\forall w \in \Sigma^*.\ v \preccurlyeq^l w \Rightarrow u \preccurlyeq^l aw \Leftrightarrow$$
$$a\mathrm{cl}_{\preccurlyeq^l}(v) \subseteq \mathrm{cl}_{\preccurlyeq^l}(u) \Leftrightarrow$$

$$\mathrm{cl}_{\preccurlyeq^l}(v) \in Q_u^a \Leftrightarrow$$
$$\gamma([\![v]\!]) \in Q_u^a.$$

Finally we prove that for any $u \in \Sigma^*$, $a \in \Sigma$, $\gamma(\delta^T([\![u]\!], a)) = \delta^A(\gamma([\![u]\!]), a)$.

$$\gamma(\delta^T([\![u]\!], a)) =$$
$$\gamma\left(\left(\bigvee_{[\![v]\!] \in Q^T \cap P_u^a} [\![v]\!]\right) \vee \left(\bigvee_{[\![v]\!] \in P_u^a \setminus Q^T} \bigwedge_{\substack{[\![z]\!] \in Q^T \\ [\![v]\!] \sqsubset [\![z]\!]}} [\![z]\!]\right)\right) =$$
$$\gamma\left(\bigvee_{[\![v]\!] \in Q^T \cap P_u^a} [\![v]\!]\right) \vee \gamma\left(\bigvee_{[\![v]\!] \in P_u^a \setminus Q^T} \bigwedge_{\substack{[\![z]\!] \in Q^T \\ [\![v]\!] \sqsubset [\![z]\!]}} [\![z]\!]\right) =$$
[Since $\gamma(Q^T) = Q^A$ and $\gamma(P_u^a) = Q_u^a$]
$$\left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q^A \cap Q_u^a} \mathrm{cl}_{\preccurlyeq^l}(v)\right) \vee \gamma\left(\bigvee_{[\![v]\!] \in P_u^a \setminus Q^T} \bigwedge_{\substack{[\![z]\!] \in Q^T \\ [\![v]\!] \sqsubset [\![z]\!]}} [\![z]\!]\right) =$$
[Since $\gamma(Q^T) = Q^A$ and $\gamma(P_u^a) = Q_u^a$]
$$\left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q^A \cap Q_u^a} \mathrm{cl}_{\preccurlyeq^l}(v)\right) \vee \left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q_u^a \setminus Q^A} \gamma\left(\bigwedge_{\substack{[\![z]\!] \in Q^T \\ [\![v]\!] \sqsubset [\![z]\!]}} [\![z]\!]\right)\right) =$$
[Since $\gamma(Q^T) = Q^A$ and by Lemma 29]
$$\left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q^A \cap Q_u^a} \mathrm{cl}_{\preccurlyeq^l}(v)\right) \vee \left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q_u^a \setminus Q^A} \bigwedge_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z) \in Q^A \\ \mathrm{cl}_{\preccurlyeq^l}(v) \subset \mathrm{cl}_{\preccurlyeq^l}(z)}} \mathrm{cl}_{\preccurlyeq^l}(z)\right) =$$
$$\left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q^A \cap Q_u^a} \mathrm{cl}_{\preccurlyeq^l}(v)\right) \vee \left(\bigvee_{\mathrm{cl}_{\preccurlyeq^l}(v) \in Q_u^a \setminus Q^A} \bigwedge_{\substack{\mathrm{cl}_{\preccurlyeq^l}(z) \in Q^A \\ \mathrm{cl}_{\preccurlyeq^l}(v) \subset \mathrm{cl}_{\preccurlyeq^l}(z)}} \mathrm{cl}_{\preccurlyeq^l}(z)\right) =$$
$$\delta^A(\mathrm{cl}_{\preccurlyeq^l}(u), a) =$$
$$\delta^A(\gamma([\![u]\!]), a).$$

■

Chapter 7.

Learning Algorithm

In this chapter we first show that the quasiorder defined in Chapter 5 we define can be refined to correspond to the Nerode's quasiorder through equivalence queries on automata constructed using the method defined in Chapter 6, allowing us to compute the canonical automaton.

AL_φ:

1 **let** $\mathcal{R} = \{\varepsilon\}$
2 **let** $\mathcal{C} = \{\varepsilon\}$
3 **repeat**
4 **let** $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$
5 **if** $\mathcal{T}$'s columns are **not** closed on the left **then**
6 **find** $c \in \mathcal{C}, a \in \Sigma$ s.t. $\nexists c' \in \mathcal{C}.\ \mathrm{col}(c') = \mathrm{col}(ac)$
7 **insert** ac **in** $\mathcal{C}$
8 **else**
9 **if** $\mathcal{T}$'s rows are **not** closed on the right **then**
10 **find** $r \in \mathcal{R}, a \in \Sigma$ s.t. $\nexists r' \in \mathcal{R}.\ \mathrm{row}(r') = \mathrm{row}(ra)$
11 **insert** ra **in** $\mathcal{R}$
12 **else**
13 **extract** $\preccurlyeq^l_\varphi$ from $\mathcal{T}$
14 **build** $T = \mathrm{T}\left(L, \preccurlyeq^l_\varphi, \mathcal{C}\right)$
15 **if** $\mathcal{L}(T) = L$
16 **return** T
17 **else**
18 **obtain** counterexample $w \in \mathcal{L}(T) \,\Delta\, L$
19 **insert** all suffixes of w **in** $\mathcal{C}$

Algorithm 1: Learning algorithm AL_φ

The algorithm leverages the fact that for an automaton $T = \mathrm{T}\left(L, \preccurlyeq^l_\varphi, \mathcal{C}\right)$, $\mathcal{L}(T) \cap \mathcal{C} = L \cap \mathcal{C}$ (Theorem 40). In other words, T recognizes correctly the words in $\mathcal{C}$. Therefore, the algorithm expands the table until it contains enough information to build the correct automaton. One way to expand the table is by trying to close the rows and columns. When they are closed, then the algorithm builds $\mathrm{T}\left(L, \preccurlyeq^l_\varphi, \mathcal{C}\right)$ and checks if it recognize the correct language. If it does not, then expand the table by adding the suffixes of the counterexample as columns. By doing so, it will be able to build a new automaton which now correctly recognizes the counterexample as well.

Furthermore, we proved that the algorithm terminates in polynomial time (Theorem 41).

Lemma 39: Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ with columns closed on the left and rows closed on the right, let $\preccurlyeq^l_\varphi$ be the left morphism-based quasiorder induced by $\mathcal{T}$. Then, $\mathcal{C}$ is a base of $\sim^l$, where $\sim^l = \preccurlyeq^l \cap \left(\preccurlyeq^l\right)^{-1}$.

Proof of Lemma 39: We want to show that $\varepsilon \in \mathcal{C} \wedge \forall w \in \Sigma^*.\, \exists c \in \mathcal{C}.\, w \sim^l_\varphi c$. Proving that $\varepsilon \in \mathcal{C}$ is easy thanks to Definition 19. To prove the second part, we choose $c = \varphi(w)$, then

$$\begin{aligned} w \sim^l_\varphi \varphi(w) \Leftrightarrow \\ \forall r \in \mathcal{R}.\, r\varphi(w) \in L \Leftrightarrow r\varphi(\varphi(w)) \in L \Leftrightarrow \text{[Since } \varphi \text{ is idempotent]} \\ \forall r \in \mathcal{R}.\, r\varphi(w) \in L \Leftrightarrow r\varphi(w) \in L \end{aligned}$$

which is indeed true. ■

Theorem 40 ($\mathrm{T}\left(L, \preccurlyeq^l_\varphi, \mathcal{C}\right)$ is precise for words in $\mathcal{C}$): Given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$ with columns closed on the left and rows closed on the right, let $\preccurlyeq^l_\varphi$ be the left morphism-based quasiorder induced by $\mathcal{T}$. Let $T = \mathrm{T}\left(L, \preccurlyeq^l_\varphi, \mathcal{C}\right)$, then

$$\forall c \in \mathcal{C}.\, c \in \mathcal{L}(T) \Leftrightarrow c \in L.$$

Note that T is well-formed since $\mathcal{C}$ is a base of $\sim^l_\varphi$ as stated by Lemma 39 and $\preccurlyeq^l_\varphi$ is a left finite index quasiorder as stated by Theorem 23.

Proof of Theorem 40:

$$\begin{aligned} c \in \mathcal{L}(T) \cap \mathcal{C} \Leftrightarrow \text{[By Theorem 35]} \\ c \in \mathrm{cl}_{\preccurlyeq^l_\varphi}(L \cap \mathcal{C}) \cap \mathcal{C} \Leftrightarrow \\ \exists u \in L \cap \mathcal{C}.\, u \preccurlyeq^l_\varphi c \wedge c \in \mathcal{C} \Leftrightarrow \\ [\Rightarrow \text{ holds because } u \in L \wedge u \preccurlyeq^l_\varphi c \Rightarrow c \in L, \text{ for any } u, c \in \mathcal{C}] \\ c \in L \cap \mathcal{C}. \end{aligned}$$

■

Theorem 41 (Termination of Algorithm AL_φ): The algorithm terminates in $O((n+m)n^3\ |\Sigma|)$.

Proof of Theorem 41: In order to prove the theorem, we need to introduce some definitions. Given a table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$, we define

$$\mathrm{col}(u) \overset{\mathcal{R}}{\subseteq} \mathrm{col}(v) \overset{\text{def}}{\iff} \forall r \in \mathcal{R}.\ \ \mathrm{col}(u)r \Rightarrow \mathrm{col}(v)r$$

$$
\begin{aligned}
\mathrm{Cols}(\mathcal{T}) &= \{\mathrm{col}(u) \mid u \in \mathcal{C}\} \\
\mathrm{Rows}(\mathcal{T}) &= \{\mathrm{row}(u) \mid u \in \mathcal{R}\} \\
\mathrm{Primes}(\mathcal{T}) &= \left\{u \in \mathrm{Cols}(\mathcal{T}) \mid \forall c \in \mathrm{Cols}(\mathcal{T}).\ u \overset{\mathcal{R}}{\subseteq} c \Leftrightarrow \forall z \in \mathrm{Cols}(\mathcal{T}).\ z \overset{\mathcal{R}}{\subset} u \Rightarrow z \overset{\mathcal{R}}{\subseteq} c\right\} \\
\mathrm{Comp}(\mathcal{T}) &= \mathrm{Cols}(\mathcal{T}) \setminus \mathrm{Primes}(\mathcal{T}) \\
\mathrm{Edges}_p(\mathcal{T}) &= \Big\{(\mathrm{col}(u), a, \mathrm{col}(v)) \in \mathrm{Primes}(\mathcal{T}) \times \Sigma \times \mathrm{Primes}(\mathcal{T}) \mid \\
&\qquad \forall c \in \mathcal{C}.\ \ \mathrm{col}(v) \overset{\mathcal{R}}{\subseteq} \mathrm{col}(c) \Rightarrow \mathrm{col}(u) \overset{\mathcal{R}}{\subseteq} \mathrm{col}(ac)\Big\} \\
\mathrm{Edges}_c(\mathcal{T}) &= \Big\{(\mathrm{col}(u), a, \mathrm{col}(v), \mathrm{col}(z)) \in \mathrm{Primes}(\mathcal{T}) \times \Sigma \times \mathrm{Comp}(\mathcal{T}) \times \mathrm{Primes}(\mathcal{T}) \mid \\
&\qquad \mathrm{col}(v) \overset{\mathcal{R}}{\subset} \mathrm{col}(z) \wedge \forall c \in \mathcal{C}.\ \ \mathrm{col}(v) \overset{\mathcal{R}}{\subseteq} \mathrm{col}(c) \Rightarrow \mathrm{col}(u) \overset{\mathcal{R}}{\subseteq} \mathrm{col}(ac)\Big\}
\end{aligned}
$$

Observe that for $u, v \in \mathcal{C}$,

$$
\begin{aligned}
\mathrm{col}(u) \overset{\mathcal{R}}{\subseteq} \mathrm{col}(v) &\Leftrightarrow \\
\forall r \in \mathcal{R}.\ \ \mathrm{col}(u)r \Rightarrow \mathrm{col}(v)r &\Leftrightarrow \\
u \preccurlyeq^l_\varphi v &
\end{aligned}
$$

In order to prove termination of the algorithm we use a measure $M(\mathcal{T}) = \big(c, r, e_p, e_c\big)$ where $c = |\mathrm{Cols}(\mathcal{T})|$, $r = |\mathrm{Rows}(\mathcal{T})|$, $e_p = |\mathrm{Edges}_p(\mathcal{T})|$ and $e_c = |\mathrm{Edges}_c(\mathcal{T})|$. Initially, $c = r = 1$, $e_p = |\Sigma \cap L|$ and $e_c = 0$

Let n be the number of right quotient of L and let m be the number of left quotient of L. The Myhill-Nerode theorem [NM57] states that n and m must be natural if and only if L is regular, which of course is. Observe that $c \leq n$ and $r \leq m$ for any table. Furthermore, $e_p \leq c^2\ |\Sigma|$ and $e_c \leq c^3\ |\Sigma|$.

Let's now look at how the measure $\big(c, r, e_p, e_c\big)$ evolves during the execution of the algorithm. We immediately observe that c and r can never decrease, since by extending the table two columns/rows that were different cannot become equal.

We now have three cases to consider.

$\mathcal{C}$ is not closed on the left: In this case we add to the table a column that, by definition, is different from all the others. Hence, c must increase by 1.

$\mathcal{R}$ is not closed on the right: In this case we add to the table a row that, by definition, is different from all the others. Hence, r must increase by 1.

$\mathcal{C}$ is closed on the left and $\mathcal{R}$ is closed on the right: In this case we can extract the morphism-based left quasiorder $\preccurlyeq^l_\varphi$ from $\mathcal{T}$ and build the automaton $T = \mathrm{T}\big(L, \preccurlyeq^l_\varphi, \mathcal{C}\big) = (\Sigma, Q, I, F, \delta)$ before the equivalence test. If the test passes, the algorithm terminates; otherwise we get a word $w \in \mathcal{L}(T) \,\Delta\, L$ and from Theorem 40, we get that $w \notin \mathcal{C}$.

> Now, we have two cases: either c increments or c stays constant. We show that if c stays constant then either $\mathcal{C}$ is not closed on the left or $\mathcal{R}$ is not closed on the right or at least one of e_p and e_c decreases. We prove it by assuming that c, e_p and e_c stay constant, $\mathcal{C}$ is closed on the left and $\mathcal{R}$ is closed on the right to obtain a contradiction.
>
> Let $\mathcal{C}' = \mathcal{C} \cup \text{Suff}(w)$, where Suff indicates the set of suffixes of w, and let $\mathcal{T}' = (L, \mathcal{R}, \mathcal{C}')$ be the extended table. Let $T' = \text{T}\big(L, \preccurlyeq_\varphi^{l\,\prime}, \mathcal{C}'\big) = (\Sigma, Q', I', F', \delta')$ and let $\preccurlyeq_\varphi^{l\,\prime}$ be the morphism-based left quasiorder extracted from $\mathcal{T}'$. We use $[\![\cdot]\!]'$ to identify the elements of Q'.
>
> We observe that, since c stays constant we have $\text{Cols}(\mathcal{T}) = \text{Cols}(\mathcal{T}')$, which in turn implies that $\text{Primes}(\mathcal{T}) = \text{Primes}(\mathcal{T}')$, since Primes depend solely on Cols. We also observe that $\text{Edges}_p(\mathcal{T}') \subseteq \text{Edges}_p(\mathcal{T})$ and $\text{Edges}_c(\mathcal{T}') \subseteq \text{Edges}_c(\mathcal{T})$, since now their defining condition is stricter because $\mathcal{C} \subset \mathcal{C}'$. Since e_p and e_c are constant, then $\text{Edges}_p(\mathcal{T}') = \text{Edges}_p(\mathcal{T})$ and $\text{Edges}_c(\mathcal{T}') = \text{Edges}_c(\mathcal{T})$. Now observe that:
> - $\text{col}(c) \in \text{Primes}(\mathcal{T}) \Leftrightarrow [\![c]\!] \in Q$;
> - $(\text{col}(u), a, \text{col}(v)) \in \text{Edges}_p(\mathcal{T})$ iff exists an existential edge labelled with a from $[\![u]\!]$ to $[\![v]\!]$;
> - $(\text{col}(u), a, \text{col}(v), \text{col}(z)) \in \text{Edges}_c(\mathcal{T})$ iff exists an universal edge labelled with a from $[\![u]\!]$ to all $[\![z]\!]$ s.t. $[\![v]\!] \sqsubset [\![z]\!]$;
> - $\text{col}(c) \in \text{Primes}(\mathcal{T}') \Leftrightarrow [\![c]\!]' \in Q'$;
> - $(\text{col}(u), a, \text{col}(v)) \in \text{Edges}_p(\mathcal{T}')$ iff exists an existential edge labelled with a from $[\![u]\!]'$ to $[\![v]\!]'$;
> - $(\text{col}(u), a, \text{col}(v), \text{col}(z)) \in \text{Edges}_c(\mathcal{T}')$ iff exists an universal edge labelled with a from $[\![u]\!]'$ to all $[\![z]\!]'$ s.t. $[\![v]\!]' \sqsubset [\![z]\!]'$.
>
> This implies that δ and δ' can be rebuilt only using the information respectively in $\text{Edges}_p(\mathcal{T})$ and $\text{Edges}_c(\mathcal{T})$ and in $\text{Edges}_p(\mathcal{T}')$ and $\text{Edges}_c(\mathcal{T}')$. Since they are pairwise equivalent, also δ and δ' must be equivalent. Furthermore we get that $Q = Q'$, since $\text{Primes}(\mathcal{T}) = \text{Primes}(\mathcal{T}')$. From this we can deduce that T and T' must be isomorphic, which is absurd since only one of them accept w correctly.

By looking at the different cases, we observe that at each iteration either:
1. c increases;
2. r increases;
3. c and r stay constant and e_p decreases;
4. c and r stay constant and e_c decreases;
5. c and r stay constant and $\mathcal{T}$ is no longer column-closed or row-closed.

This means that every two iteration either:
1. c increases;
2. r increases;
3. c and r stay constant and e_p decreases;
4. c and r stay constant and e_c decreases.

Since c and r are bounded respectively by n and m, and e_p and e_c are bounded by $n^2\ |\Sigma|$ and $n^3\ |\Sigma|$, the algorithms terminates after at most $O(n^3(n+m)\ |\Sigma|)$ equivalence queries.

The maximal number of membership queries is the size of the table which has $O(m\ |\Sigma|)$ rows ($\mathcal{R} \cup \mathcal{R}\Sigma$) and $O(ln^3(n+m)\ |\Sigma|)$ columns ($\mathcal{C} \cup \Sigma\mathcal{C}$), where l is the maximum length of a counterexample given by the oracle. Thus having at most $O(lmn^3(n+m)\ |\Sigma|^2)$ membership queries. ■

7.1. Comparison with AL^* and AL^{**}

Algorithms AL^* and AL^{**} are also based on the construction of inclusion tables but they use different conditions to decide whether to make a membership query to the teacher or an equivalence query to the oracle. Theorem 42 proves that the condition for making a equivalence query in AL_φ is stricter than the condition used in AL^* and AL^{**}. As a consequence, AL_φ will query the oracle less frequently than AL^* and AL^{**}.

Theorem 42: AL_φ table has stricter or equal condition than AL^* and AL^{**}. If a table has rows closed on the right and columns closed on the left then that table is both closed and consistent, as defined in [Ber+17]. More formally, we want to prove that, given an inclusion table $\mathcal{T} = (L, \mathcal{R}, \mathcal{C})$, assuming

1. $\forall r \in \mathcal{R}.\ \forall a \in \Sigma.\ \exists r' \in \mathcal{R}.\ \forall c \in \mathcal{C}.\ rac \in L \Leftrightarrow r'c \in L$;
2. $\forall c \in \mathcal{C}.\ \forall a \in \Sigma.\ \exists c' \in \mathcal{C}.\ \forall r \in \mathcal{R}.\ rac \in L \Leftrightarrow rc' \in L$;

we want to show

$$\forall r, r' \in \mathcal{R}.\ \ \mathrm{row}(r) = \mathrm{row}(r') \Rightarrow \forall a \in \Sigma.\ \ \mathrm{row}(ra) = \mathrm{row}(r'a).$$

Proof of Theorem 42: For any $r, r' \in \mathcal{R}$

$$\begin{aligned}
\mathrm{row}(r) = \mathrm{row}(r') &\Leftrightarrow \\
\forall c \in \mathcal{C}.\ \psi(r)c \in L \Leftrightarrow \psi(r')c \in L &\Rightarrow [\because r, r' \in \mathcal{R}] \\
\forall c \in \mathcal{C}.\ rc \in L \Leftrightarrow r'c \in L &\Rightarrow [\because \rho(ac) \in \mathcal{C}] \\
\forall c \in \mathcal{C}.\ r\rho(ac) \in L \Leftrightarrow r'\rho(ac) \in L &\Leftrightarrow [\text{By hyp. 1}] \\
\forall c \in \mathcal{C}.\ rac \in L \Leftrightarrow r'ac \in L &\Leftrightarrow [\text{By hyp. 2}] \\
\\
\forall c \in \mathcal{C}.\ \tau(ra)c \in L \Leftrightarrow \tau(r'a)c \in L &\Leftrightarrow [\because r, r' \in \mathcal{R}] \\
\forall c \in \mathcal{C}.\ \tau(\psi(r)a)c \in L \Leftrightarrow \tau(\psi(r')a)c \in L &\Leftrightarrow \\
\forall c \in \mathcal{C}.\ \psi(ra)c \in L \Leftrightarrow \psi(r'a)c \in L &\Leftrightarrow \\
\mathrm{row}(ra) = \mathrm{row}(r'a). &
\end{aligned}$$

■

Algorithm AL_φ may yield a different result than AL^* and AL^{**}. An example of language for which the algorithms yield different results is $L = \{a, ac, bb, bc\}$. The output for AL_φ is displayed in Figure 7, while both AL^* and AL^{**} output the automaton in Figure 8.

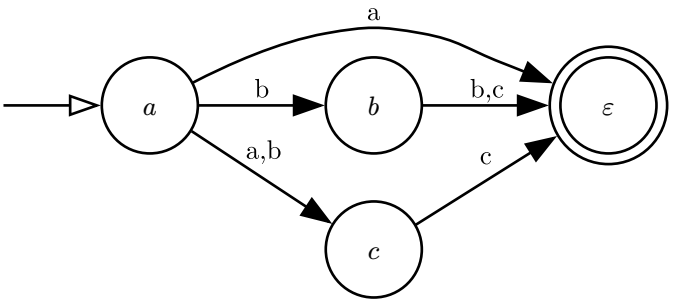


Figure 7: Automaton produced by algorithm AL_φ for language $L = \{a, ac, bb, bc\}$.

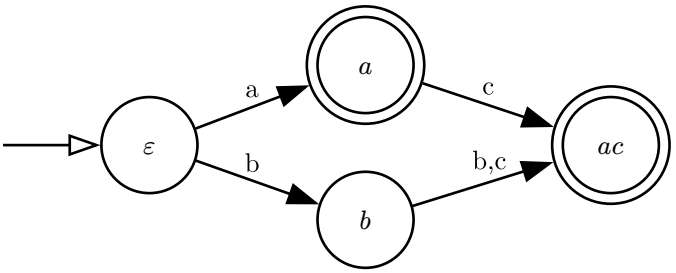


Figure 8: Automaton produced by algorithms AL^* and AL^{**} for language $L = \{a, ac, bb, bc\}$.

7.2. Example Run of Algorithm AL_φ

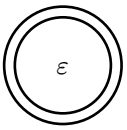
Example 9:

$$L = \{ad, af, be, bf, cf, da, eb, fa, fb\}$$

0. Begin by building a table with rows and columns equal to $\{\varepsilon\}$.

L		$\mathcal{C}$
		ε
$\mathcal{R}$	ε	$-$

1. The table's columns are closed on the left and table's rows are closed on the right. We build the automaton $\mathrm{T}\left(L, \preccurlyeq^l_\varphi, \mathcal{C}\right)$ in the figure.



Its language is $\emptyset$, therefore a counterexample is ad. Add $\{ad, d\}$ to $\mathcal{C}$.

L	$\mathcal{C}$		
	ε	ad	d

$\mathcal{R}$	ε	$-$	$+$	$-$

2. Table's rows are not closed on the right. Add a to $\mathcal{R}$.

L		$\mathcal{C}$		
		ε	ad	d
$\mathcal{R}$	ε	$-$	$+$	$-$
	a	$-$	$-$	$+$

3. Table's rows are not closed on the right. Add b to $\mathcal{R}$.

L		$\mathcal{C}$		
		ε	ad	d
$\mathcal{R}$	ε	$-$	$+$	$-$
	a	$-$	$-$	$+$
	b	$-$	$-$	$-$

4. Table's cols are not closed on the left. Add e to $\mathcal{C}$.

L		$\mathcal{C}$			
		ε	ad	d	e
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$
	a	$-$	$-$	$+$	$-$
	b	$-$	$-$	$-$	$+$

5. Table's cols are not closed on the left. Add f to $\mathcal{C}$.

L		$\mathcal{C}$				
		ε	ad	d	e	f
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$	$-$
	a	$-$	$-$	$+$	$-$	$+$
	b	$-$	$-$	$-$	$+$	$+$

6. Table's rows are not closed on the right. Add c to $\mathcal{R}$.

L		$\mathcal{C}$				
		ε	ad	d	e	f
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$	$-$
	a	$-$	$-$	$+$	$-$	$+$

	b	$-$	$-$	$-$	$+$	$+$
	c	$-$	$-$	$-$	$-$	$+$

7. Table's rows are not closed on the right. Add d to $\mathcal{R}$.

L		$\mathcal{C}$				
		ε	ad	d	e	f
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$	$-$
	a	$-$	$-$	$+$	$-$	$+$
	b	$-$	$-$	$-$	$+$	$+$
	c	$-$	$-$	$-$	$-$	$+$
	d	$-$	$-$	$-$	$-$	$-$

8. Table's cols are not closed on the left. Add a to $\mathcal{C}$.

L		$\mathcal{C}$					
		ε	ad	d	e	f	a
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$	$-$	$-$
	a	$-$	$-$	$+$	$-$	$+$	$-$
	b	$-$	$-$	$-$	$+$	$+$	$-$
	c	$-$	$-$	$-$	$-$	$+$	$-$
	d	$-$	$-$	$-$	$-$	$-$	$+$

9. Table's rows are not closed on the right. Add e to $\mathcal{R}$.

L		$\mathcal{C}$					
		ε	ad	d	e	f	a
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$	$-$	$-$
	a	$-$	$-$	$+$	$-$	$+$	$-$
	b	$-$	$-$	$-$	$+$	$+$	$-$
	c	$-$	$-$	$-$	$-$	$+$	$-$
	d	$-$	$-$	$-$	$-$	$-$	$+$
	e	$-$	$-$	$-$	$-$	$-$	$-$

10. Table's cols are not closed on the left. Add b to $\mathcal{C}$.

L	$\mathcal{C}$						
	ε	ad	d	e	f	a	b

$\mathcal{R}$	ε	−	+	−	−	−	−	−
	a	−	−	+	−	+	−	−
	b	−	−	−	+	+	−	−
	c	−	−	−	−	+	−	−
	d	−	−	−	−	−	+	−
	e	−	−	−	−	−	−	+

11. Table's rows are not closed on the right. Add f to $\mathcal{R}$.

L		$\mathcal{C}$						
		ε	ad	d	e	f	a	b
$\mathcal{R}$	ε	−	+	−	−	−	−	−
	a	−	−	+	−	+	−	−
	b	−	−	−	+	+	−	−
	c	−	−	−	−	+	−	−
	d	−	−	−	−	−	+	−
	e	−	−	−	−	−	−	+
	f	−	−	−	−	−	+	+

12. Table's rows are not closed on the right. Add aa to $\mathcal{R}$.

L		$\mathcal{C}$						
		ε	ad	d	e	f	a	b
$\mathcal{R}$	ε	−	+	−	−	−	−	−
	a	−	−	+	−	+	−	−
	b	−	−	−	+	+	−	−
	c	−	−	−	−	+	−	−
	d	−	−	−	−	−	+	−
	e	−	−	−	−	−	−	+
	f	−	−	−	−	−	+	+
	aa	−	−	−	−	−	−	−

13. Table's rows are not closed on the right. Add ad to $\mathcal{R}$.

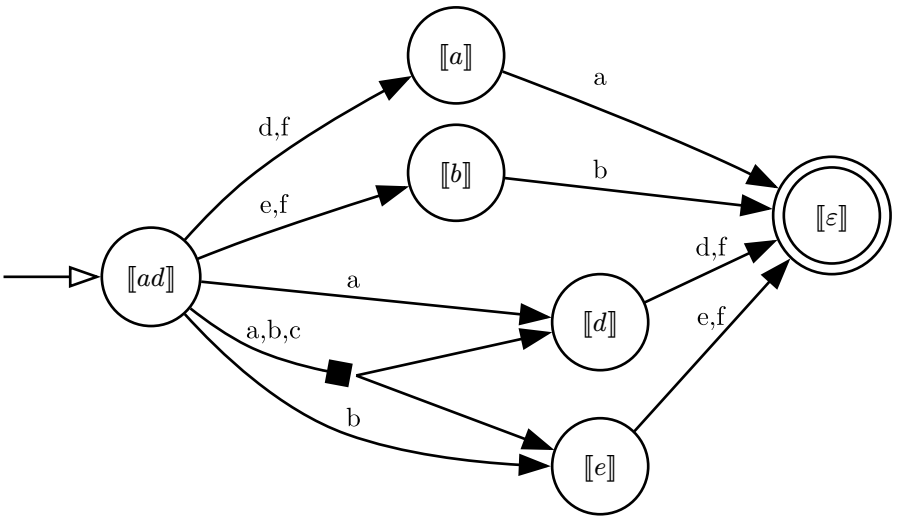
L		$\mathcal{C}$						
		ε	ad	d	e	f	a	b
$\mathcal{R}$	ε	−	+	−	−	−	−	−

	a	$-$	$-$	$+$	$-$	$+$	$-$	$-$
	b	$-$	$-$	$-$	$+$	$+$	$-$	$-$
	c	$-$	$-$	$-$	$-$	$+$	$-$	$-$
	d	$-$	$-$	$-$	$-$	$-$	$+$	$-$
	e	$-$	$-$	$-$	$-$	$-$	$-$	$+$
	f	$-$	$-$	$-$	$-$	$-$	$+$	$+$
	aa	$-$	$-$	$-$	$-$	$-$	$-$	$-$
	ad	$+$	$-$	$-$	$-$	$-$	$-$	$-$

14. Table's cols are not closed on the left. Add c to $\mathcal{C}$.

L		$\mathcal{C}$							
		ε	ad	d	e	f	a	b	c
$\mathcal{R}$	ε	$-$	$+$	$-$	$-$	$-$	$-$	$-$	$-$
	a	$-$	$-$	$+$	$-$	$+$	$-$	$-$	$-$
	b	$-$	$-$	$-$	$+$	$+$	$-$	$-$	$-$
	c	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$
	d	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$
	e	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$
	f	$-$	$-$	$-$	$-$	$-$	$+$	$+$	$-$
	aa	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
	ad	$+$	$-$	$-$	$-$	$-$	$-$	$-$	$-$

15. The table's columns are closed on the left and table's rows are closed on the right. We build the automaton $\mathrm{T}\left(L, \preccurlyeq_{\varphi}^{l}, \mathcal{C}\right)$ in the figure.



Its language is $\{ad, af, be, bf, cf, da, eb, fa, fb\}$, therefore the algorithm terminates and this is the canonical AFA for the language.

Chapter 8.

Conclusions

In this thesis, we have introduced a novel class of alternating finite automata based on finite index quasiorder principals. Building upon this foundation, we have proposed a canonical form for alternating automata. While we think that automata in this form may represent the minimal principal automata, a rigorous proof of this conjecture remains elusive and is thus deferred to future works.

A significant contribution of our work is the development of a learning algorithm that, for any given regular language, produces an alternating automaton in canonical form. Although we have demonstrated the algorithm's polynomial-time termination, a fully functional implementation remains to be realized. The development of such an implementation should be prioritized, as it would facilitate a more comprehensive comparison with existing algorithms and enable empirical assessment of its complexity. We hypothesize that the average-case complexity may be substantially lower than the established upper bound, warranting further investigation.

A natural extension of our research would involve the definition of operations on alternating automata that enable the transformation of a given automaton into its canonical form without necessitating the application of the learning algorithm. Such an approach would parallel the methodologies outlined by F. Denis, A. Lemay, and A. Terlutte [DLT02] for other classes of automata. This requires the definition of two operations on alternating automata. One is to replace a state with a universal transition, and one is to add a new transition. These two operators must be defined in such a way that, if applied repeatedly, they return a canonical automaton.

Several intriguing questions emerge from our work, presenting avenues for future exploration:

1. the apparent fundamental nature of principals for AFAs, analogous to the role of quotients in non-deterministic finite automata;
2. the divergence in behavior between residual AFAs and their non-deterministic and universal counterparts;
3. the potential implications and applications of extending the concept of principals to non-deterministic automata.

In conclusion, this thesis has advanced our understanding of alternating finite automata proposing a canonical representation and developing an Angluin-style learning algorithm. These contributions not only enrich the theoretical landscape of automata theory but also make way for practical applications in formal language learning and analysis.

Bibliography

[Ang87] D. Angluin, "Learning regular sets from queries and counterexamples," *Information and computation*, vol. 75, no. 2, pp. 87–106, 1987.

[AEF15] D. Angluin, S. Eisenstat, and D. Fisman, "Learning Regular Languages via Alternating Automata.," in *IJCAI*, 2015, pp. 3308–3314.

[Ber+05] T. Berg, B. Jonsson, M. Leucker, and M. Saksena, "Insights to Angluin's learning," *Electronic Notes in Theoretical Computer Science*, vol. 118, pp. 3–18, 2005.

[Ber+17] S. Berndt, M. Liśkiewicz, M. Lutter, and R. Reischuk, "Learning residual alternating automata," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[Bol+09] B. Bollig, P. Habermehl, C. Kern, and M. Leucker, "Angluin-style learning of NFA," 2009, pp. 1004–1009.

[CKS81] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer, "Alternation," *Journal of the ACM (JACM)*, vol. 28, no. 1, pp. 114–133, 1981.

[DLT02] F. Denis, A. Lemay, and A. Terlutte, "Residual finite state automata," *Fundamenta Informaticae*, vol. 51, no. 4, pp. 339–368, 2002.

[GGV19] P. Ganty, E. Gutiérrez, and P. Valero, "A congruence-based perspective on automata minimization algorithms," *arXiv preprint arXiv:1906.06194*, 2019.

[GGV20] P. Ganty, E. Gutiérrez, and P. Valero, "A quasiorder-based perspective on residual automata," *arXiv preprint arXiv:2007.00359*, 2020.

[Jir23] G. Jirásková, "Operations on Boolean and Alternating Finite Automata," *arXiv preprint arXiv:2309.02748*, 2023.

[LS08] S. Lombardy and J. Sakarovitch, "The universal automaton.," *Logic and automata*, vol. 2, pp. 457–504, 2008.

[LV94] A. de Luca and S. Varricchio, "Well quasi-orders and regular languages," *Acta Informatica*, vol. 31, pp. 539–557, 1994.

[NM57] A. Nerode and J. Myhill, *Fundamental Concepts in the Theory of Systems*. in ASTIA Document. Wright Air Development Center, Air Research, Development Command, United States Air Force, 1957.

[RS97] G. Rozenberg and A. Salomaa, Eds., *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*. Springer, 1997. doi: 10.1007/978-3-642-59136-5.

[SWY00] K. Salomaa, X. Wu, and S. Yu, "Efficient implementation of regular languages using reversed alternating finite automata," *Theoretical Computer Science*, vol. 231, no. 1, pp. 103–111, 2000.