Theoretical Aspects of Symbolic Automata *

Hellis Tamm[1] and Margus Veanes[2]

[1] Tallinn University of Technology
hellis@cs.ioc.ee
[2] Microsoft Research
margus@microsoft.com

Abstract. Symbolic finite automata extend classical automata by allowing infinite alphabets given by Boolean algebras and having transitions labeled by predicates over such algebras. Symbolic automata have been intensively studied recently and they have proven useful in several applications. We study some theoretical aspects of symbolic automata. Especially, we study minterms of symbolic automata, that is, the set of maximal satisfiable Boolean combinations of predicates of automata. We define canonical minterms of a language accepted by a symbolic automaton and show that these minterms can be used to define symbolic versions of some known classical automata. Also we show that canonical minterms have an important role in finding minimal nondeterministic symbolic automata. We show that Brzozowski's double-reversal method for minimizing classical deterministic automata as well as its generalization is applicable for symbolic automata.

1 Introduction

Symbolic finite automata are finite state automata with an alphabet given by a Boolean algebra which can possibly have an infinite domain, and with transitions labeled by predicates over such algebra. Symbolic finite automata are a generalization of nondeterministic finite automata (NFAs), with a motivation for their introduction coming from practical applications which require handling of large or infinite alphabets.

Automata with predicates was first mentioned in [18]. We consider symbolic finite automata as defined in [4], where predicates are drawn from a decidable Boolean algebra. These automata have been intensively studied recently, for example, in the context of minimization of deterministic symbolic automata [5], computing forward bisimulations for nondeterministic symbolic automata [6], learning symbolic automata [9], and others.

We study some theoretical aspects of symbolic automata. We call the languages accepted by symbolic automata *symbolic regular languages*. These languages can be expressed with *symbolic regular expressions*. A similar symbolic

* This work was supported by the Estonian Ministry of Education and Research institutional research grant IUT33-13.

generalization of regular expressions, called *extended regular expressions*, was used in [11] where also intersection and negation operators were supported.

We study *minterms* of symbolic automata, that is, the set of maximal satisfiable Boolean combinations of predicates of automata. It was pointed out in [4, 5] that minterms can be used as a finite alphabet when adapting classical automata algorithms to the symbolic setting. This is because a symbolic automaton has a finite number of transition predicates, implying that the set of minterms is finite as well. We show that any symbolic regular language has a minimal set of minterms, the minterms of its minimal deterministic automaton. This set of minterms is unique up to predicate equivalence. We show that this set of canonical minterms can be used in place of a finite alphabet to define symbolic versions of some known NFAs, such as the *symbolic átomaton* and *canonical symbolic residual finite state automaton* of the language. Also we show that the minterms of a language have an important role in finding minimal nondeterministic symbolic automata.

We show that Brzozowski's double-reversal method for minimizing classical deterministic automata as well as its generalization is applicable for symbolic automata.

2 Symbolic Regular Languages and Symbolic Finite Automata

An *effective Boolean algebra* B has components $(\Sigma, \Psi, [\![\_]\!], \bot, \top, \vee, \wedge, \neg)$, where Σ is a set of *domain elements*, Ψ is a set of *predicates* closed under the Boolean connectives, and $\bot, \top \in \Psi$. The *denotation function* $[\![\_]\!] : \Psi \to 2^{\Sigma}$ is such that $[\![\bot]\!] = \emptyset$, $[\![\top]\!] = \Sigma$, and for all $\varphi, \psi \in \Psi$, $[\![\varphi \vee \psi]\!] = [\![\varphi]\!] \cup [\![\psi]\!]$, $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \cap [\![\psi]\!]$, and $[\![\neg\varphi]\!] = \Sigma \setminus [\![\varphi]\!]$. If $[\![\varphi]\!] \neq \emptyset$, then φ is *satisfiable*. We require that checking satisfiability is decidable.

A predicate φ is a *subpredicate* of ψ, if $[\![\varphi]\!] \subseteq [\![\psi]\!]$.

Elements of Σ are *characters*. A *word* over Σ is a sequence $a_{i_1} \cdots a_{i_m}$, where $a_{i_j} \in \Sigma$, $j = 1, \ldots, m$. If $m = 0$, then we get the *empty* word, denoted by ε. The set of all words over Σ is denoted by Σ^*. We require that $\Sigma^n \cap \Sigma = \emptyset$ for $n \geq 2$.

We define a *symbolic regular expression* as follows:

- The constants ε and $\emptyset$ are symbolic regular expressions, denoting the languages $\{\varepsilon\}$ and $\emptyset$, respectively.
- For any predicate $\varphi \in \Psi$, φ is a symbolic regular expression, denoting the language $L(\varphi) = [\![\varphi]\!]$.
- For any symbolic regular expressions X and Y, the expressions $X + Y$, XY, and X^* are symbolic regular expressions, denoting respectively the languages $L(X) \cup L(Y)$, $L(X)L(Y)$, and $(L(X))^*$.

Any language defined by a symbolic regular expression is a *symbolic regular language*.

A *symbolic nondeterministic finite automaton (s-NFA)* is a quintuple $\mathcal{N} = (B, Q, \Delta, I, F)$, where $B = (\Sigma, \Psi, [\![\_]\!], \bot, \top, \vee, \wedge, \neg)$ is an effective Boolean algebra, called the *alphabet*, Q is a finite set of *states*, $\Delta \subseteq Q \times \Psi \times Q$ is a finite set

of *transitions*, $I \subseteq Q$ is the set of *initial states*, and $F \subseteq Q$ is the set of *final states*. The *left language* of a state q of $\mathcal{N}$, denoted by $L_{I,q}(\mathcal{N})$, is the set of words $w \in \Sigma^*$ such that, either $w = \varepsilon$ and $q \in I$, or $w = a_1 \dots a_k$ and there exist states $q_1, \dots, q_k \in Q$ such that $(q_{i-1}, \varphi_i, q_i) \in \Delta$ and $a_i \in [\![\varphi_i]\!]$, with $q_0 \in I$ and $q_k = q$. The *right language*, or simply, the *language* of a state q of $\mathcal{N}$, denoted by $L_{q,F}(\mathcal{N})$, or simply, $L_q(\mathcal{N})$, is the set of words $w \in \Sigma^*$ such that, either $w = \varepsilon$ and $q \in F$, or $w = a_1 \dots a_k$ and there exist states $q_1, \dots, q_k \in Q$ such that $(q_{i-1}, \varphi_i, q_i) \in \Delta$ and $a_i \in [\![\varphi_i]\!]$, with $q_0 = q$ and $q_k \in F$. A state is *unreachable* if its left language is empty. A state is *empty* if its right language is empty. An s-NFA is *trim* if it does not have any unreachable or empty states. The *language accepted* by $\mathcal{N}$ is $L(\mathcal{N}) = \bigcup_{q \in I} L_q(\mathcal{N})$. Two s-NFAs are *equivalent* if they accept the same language. The *reverse* of an s-NFA $\mathcal{N} = (B, Q, \Delta, I, F)$ is the s-NFA $\mathcal{N}^R = (B, Q, \Delta^R, F, I)$, where $(q, \varphi, p) \in \Delta^R$ if and only if $(p, \varphi, q) \in \Delta$ for $p, q \in Q$ and $\varphi \in \Psi$. An s-NFA $\mathcal{N}$ is *normalized for predicates* if for all $p, q \in Q$ there is at most one predicate φ such that $(p, \varphi, q) \in \Delta$. Any s-NFA $\mathcal{N}$ can be normalized, resulting in the s-NFA $\mathcal{N}^N$, where all distinct transitions (p, φ_1, q) and (p, φ_2, q) from any state p to any state q have been replaced by a single transition $(p, \varphi_1 \vee \varphi_2, q)$. An s-NFA $\mathcal{N}$ is *complete* if for every $p \in Q$ and $a \in \Sigma$, there is a transition $(p, \varphi, q) \in \Delta$ with $a \in [\![\varphi]\!]$, $q \in Q$. If $\mathcal{N}_1 = (B, Q_1, \Delta_1, I_1, F_1)$ and $\mathcal{N}_2 = (B, Q_2, \Delta_2, I_2, F_2)$ are s-NFAs, then a map π from Q_1 into Q_2 is a *morphism* from $\mathcal{N}_1$ into $\mathcal{N}_2$ if and only if $\pi(I_1) \subseteq I_2$, $\pi(F_1) \subseteq F_2$, and for all states $p, q \in Q_1$ and $a \in \Sigma$ it holds that if $(p, \varphi_1, q) \in \Delta_1$ for some φ_1 such that $a \in [\![\varphi_1]\!]$, then there is some φ_2 such that $(\pi(p), \varphi_2, \pi(q)) \in \Delta_2$ and $a \in [\![\varphi_2]\!]$.

Sometimes it is useful to allow transitions on the empty word ε in a symbolic automaton. A *symbolic nondeterministic finite automaton with epsilon transitions (s-εNFA)* is $\mathcal{N} = (B, Q, \Delta, I, F)$, where B, Q, I, and F are as in an s-NFA, and $\Delta \subseteq Q \times (\Psi \cup \{\varepsilon\}) \times Q$.

Similarly to regular languages, one can show that symbolic regular languages are accepted by s-NFAs and *vice versa*.

We can apply the well-known Thompson's construction [15, 17] to a symbolic regular expression, to obtain an s-εNFA. We present a slightly modified version of this construction in the following proposition:

Proposition 1. *Every symbolic regular language is accepted by an s-εNFA.*

Proof. An s-εNFA can be constructed from any symbolic regular expression, using structural induction which involves parts described as follows: First, the s-εNFAs for the constants ε and $\emptyset$ are respectively $\mathcal{N}_\varepsilon = (B, \{q\}, \emptyset, \{q\}, \{q\})$ and $\mathcal{N}_\emptyset = (B, \emptyset, \emptyset, \emptyset, \emptyset)$, and the s-$\varepsilon$NFA for any predicate $\varphi \in \Psi$ is $\mathcal{N}_\varphi = (B, \{q_1, q_2\}, \{(q_1, \varphi, q_2)\}, \{q_1\}, \{q_2\})$.

Now, let $\mathcal{N}_X = (B, Q_X, \Delta_X, I_X, F_X)$ be the s-εNFA for the expression X, and let $\mathcal{N}_Y = (B, Q_Y, \Delta_Y, I_Y, F_Y)$ be the s-εNFA for the expression Y, where the sets Q_X and Q_Y are disjoint. The s-εNFAs for the expressions XY, $X + Y$, and X^* are respectively $\mathcal{N}_{XY} = (B, Q_X \cup Q_Y, \Delta_X \cup \Delta_Y \cup (F_X \times \{\varepsilon\} \times I_Y), I_X, F_Y)$, $\mathcal{N}_{X+Y} = (B, Q_X \cup Q_Y \cup \{q_1, q_2\}, \Delta_X \cup \Delta_Y \cup (\{q_1\} \times \{\varepsilon\} \times (I_X \cup I_Y)) \cup ((F_X \cup F_Y) \times \{\varepsilon\} \times \{q_2\}), \{q_1\}, \{q_2\})$, and $\mathcal{N}_{X^*} = (B, Q_X \cup \{q_1, q_2\}, \Delta_X \cup (\{q_1\} \times \{\varepsilon\} \times (I_X \cup \{q_2\})) \cup (F_X \times \{\varepsilon\} \times (I_X \cup \{q_2\})), \{q_1\}, \{q_2\})$, where $q_1, q_2 \notin Q_X \cup Q_Y$. □

Similarly to finite automata accepting regular languages, an s-εNFA can be converted to an equivalent s-NFA by eliminating epsilon transitions by standard methods.

An s-NFA $\mathcal{N} = (B, Q, \Delta, I, F)$ is a *symbolic deterministic finite automaton (s-DFA)* if $|I| = 1$ and if for all transitions $(p, \varphi, q), (p', \varphi', q') \in \Delta$ it holds that if $p = p'$ and $\varphi \wedge \varphi'$ is satisfiable, then $q = q'$. An s-NFA $\mathcal{N}$ can be *determinized* to obtain an equivalent s-DFA $\mathcal{N}^D = (B, Q^D, \Delta^D, \{s_0\}, F^D)$, using a symbolic version [16] of the well-known subset construction procedure. We present here a slightly modified variant of it which produces a complete and normalized s-DFA. Similarly to the classical subset construction, this procedure gradually forms the set Q^D of states, along with the set Δ^D of transitions of $\mathcal{N}^D$, including only reachable states, starting with $Q^D = \{I\}$ and $\Delta^D = \emptyset$. For every $s \in Q^D$, we do the following steps: first, we form the set S_s of states q of $\mathcal{N}$ such that there is a transition $(p, \varphi, q) \in \Delta$ from a state $p \in s$ to q with some $\varphi \in \Psi$; then, for all $q \in S_s$, let $\varphi_{s,q} = \bigvee_{(p,\varphi,q)\in\Delta, p\in s} \varphi$; for every $s' \subseteq S_s$, let $\varphi_{s,s'} = (\bigwedge_{q\in s'} \varphi_{s,q}) \wedge (\bigwedge_{q\in S_s\setminus s'} \neg\varphi_{s,q})$; if $\varphi_{s,s'}$ is satisfiable, then we add s' to Q^D (if $s' \notin Q^D$) and add the transition $(s, \varphi_{s,s'}, s')$ to Δ^D. Finally, we let $s_0 = I$ and $F^D = \{s \in Q^D \mid s \cap F \neq \emptyset\}$.

We assume for the rest of the paper that s-DFAs are complete. An s-DFA is *minimal* if it has the minimal number of states among all equivalent s-DFAs. We also require that a minimal s-DFA is normalized for predicates. A minimal s-DFA is unique up to renaming of states and equivalence of predicates [5]. In the minimal s-DFA, the languages of any two distinct states are different from each other. It is easy to see that every predicate occurring in any s-DFA with reachable states only, is a subpredicate of some predicate of the minimal s-DFA of the same language. This is because the minimal s-DFA can be obtained from any such s-DFA by merging some states and transitions.

3 Brzozowski's Theorem for Symbolic Automata

In this section we consider non-empty symbolic regular languages. We show the symbolic version of a (slightly modified) classical result by Brzozowski [2]:

Theorem 1. *If an s-NFA $\mathcal{N}$ has no empty states and $\mathcal{N}^R$ is an s-DFA, then $\mathcal{N}^D$ is minimal.*

Proof. Let $\mathcal{N} = (B, Q, \Delta, I, F)$ be an s-NFA with no empty states such that its reverse s-NFA $\mathcal{N}^R = (B, Q, \Delta^R, F, I)$ is an s-DFA. We note that F is a singleton set. If $|Q| = 1$, then $\mathcal{N}$ and $\mathcal{N}^R$ are the same automata. The determinized version $\mathcal{N}^D$ of $\mathcal{N}$ is complete and normalized, and has one or two states. In the former case $\mathcal{N}^D$ is clearly minimal. In the latter case, one of the states of $\mathcal{N}^D$ is the initial state with a non-empty language, and the other is an empty state, so the languages of these two states are different, implying that $\mathcal{N}^D$ is minimal.

We now consider the case where $|Q| \geq 2$. Let $q, q' \in Q$, with $q \neq q'$. We show that $L_q(\mathcal{N}) \cap L_{q'}(\mathcal{N}) = \emptyset$. Indeed, suppose that there is a word $w \in \Sigma^*$ such

that $w \in L_q(\mathcal{N})$ and $w \in L_{q'}(\mathcal{N})$. If $w = \varepsilon$, then both q and q' must be final, but since $\mathcal{N}$ has only one final state, we have a contradiction. If $w = a_1 \dots a_k$, where $k \geq 1$, then there are some states $q_{i-1}, q'_{i-1}, q_i \in Q$ such that $q_{i-1} \neq q'_{i-1}$, and transitions $(q_{i-1}, \varphi, q_i), (q'_{i-1}, \varphi', q_i) \in \Delta$ with $a_i \in [\![\varphi]\!]$ and $a_i \in [\![\varphi']\!]$. Therefore, $\varphi \wedge \varphi'$ is satisfiable, implying that $\mathcal{N}^R$ is not deterministic, a contradiction.

Now, let s_1 and s_2 be any two states of $\mathcal{N}^D$, where $s_1 \neq s_2$. We show that $L_{s_1}(\mathcal{N}^D) \neq L_{s_2}(\mathcal{N}^D)$. Indeed, because both s_1 and s_2 are subsets of Q, there is a state $q \in Q$ such that either $q \in s_1$ and $q \notin s_2$, or $q \in s_2$ and $q \notin s_1$. Since $L_{s_1}(\mathcal{N}^D) = \bigcup_{q \in s_1} L_q(\mathcal{N})$ and $L_{s_2}(\mathcal{N}^D) = \bigcup_{q \in s_2} L_q(\mathcal{N})$, and because we assumed that $L_q(\mathcal{N}) \neq \emptyset$ for any $q \in Q$, and showed above that $L_q(\mathcal{N}) \cap L_{q'}(\mathcal{N}) = \emptyset$ for every $q, q' \in Q$ with $q \neq q'$, it holds that $L_{s_1}(\mathcal{N}^D) \neq L_{s_2}(\mathcal{N}^D)$. Therefore, $\mathcal{N}^D$ is minimal. □

Based on Theorem 1, similarly to its classical version, it is possible to get a minimization algorithm for s-DFAs which we call *Brzozowski's minimization* or *double-reversal minimization* algorithm. By this algorithm, the minimal s-DFA of a language can be obtained from any s-NFA $\mathcal{N}$ by first applying the determinization procedure to the reverse $\mathcal{N}^R$ of $\mathcal{N}$ to obtain an s-DFA $\mathcal{N}^{RD}$ of the reverse language, and then applying determinization to its reverse $\mathcal{N}^{RDR}$ to obtain the s-DFA $\mathcal{N}^{RDRD}$. By Theorem 1, $\mathcal{N}^{RDRD}$ is a minimal s-DFA.

4 Minterms of Symbolic Automata

Let L be a symbolic regular language. Let $\mathcal{N}$ be an s-NFA of L and let $\mathcal{N}^N$ be the s-NFA obtained from $\mathcal{N}$ by predicate normalization. Let $\varphi_1, \dots, \varphi_k$ be the predicates occurring in $\mathcal{N}^N$. Any satisfiable predicate $(\bigwedge_{i \in S} \varphi_i) \wedge (\bigwedge_{i \in \overline{S}} \neg \varphi_i)$ with $S \subseteq \{1, \dots, k\}$ and $\overline{S} = \{1, \dots, k\} \setminus S$ is a *minterm* of $\mathcal{N}$. Obviously, $\mathcal{N}$ and $\mathcal{N}^N$ have the same set of minterms. Also, it is easy to see that every predicate of $\mathcal{N}^N$ is a disjunction of minterms of $\mathcal{N}$. The minterms of $\mathcal{N}$ provide a partition of Σ.

We show that the minterms of any s-NFA of L are a refinement of the minterms of the minimal s-DFA of L.

Proposition 2. *Any minterm of an s-NFA of L is a subpredicate of some minterm of the minimal s-DFA of L.*

Proof. Let us first consider an s-NFA $\mathcal{N}$ that has only reachable states, and its normalized variant $\mathcal{N}^N$. We notice that the determinized versions $\mathcal{N}^D$ and $\mathcal{N}^{ND}$ of these two automata are the same s-DFAs (up to predicate equivalence). Let $s \subseteq Q$ be a state of $\mathcal{N}^{ND}$. Let $\varphi_1, \dots, \varphi_k$ be the predicates occurring in $\mathcal{N}^N$ as the labels of out-transitions of the states $p \in s$, and let $\psi_1, \dots, \psi_\ell$ be the predicates in $\mathcal{N}^{ND}$ which occur as the labels of out-transitions of s. According to how $\mathcal{N}^{ND}$ is constructed from $\mathcal{N}^N$, we notice that every ψ_j is a disjunction of predicates of the form $(\bigwedge_{i \in S} \varphi_i) \wedge (\bigwedge_{i \in \overline{S}} \neg \varphi_i)$, where $S \subseteq \{1, \dots, k\}$ and $\overline{S} = \{1, \dots, k\} \setminus S$, therefore any predicate $(\bigwedge_{i \in S} \varphi_i) \wedge (\bigwedge_{i \in \overline{S}} \neg \varphi_i)$ is a subpredicate of some ψ_j. Since $\mathcal{N}^{ND}$ is deterministic, $\psi_h \wedge \psi_j$ is not satisfiable if $h \neq j$, and thus

$(\bigwedge_{i\in S}\varphi_i)\wedge(\bigwedge_{i\in\overline{S}}\neg\varphi_i)$ is a subpredicate of $\psi_j\wedge(\bigwedge_{h\in\{1,\ldots,\ell\},h\neq j}\neg\psi_h)$. Because every minterm of $\mathcal{N}^N$ is a conjunction of predicates of the form $(\bigwedge_{i\in S}\varphi_i)\wedge(\bigwedge_{i\in\overline{S}}\neg\varphi_i)$, and every minterm of $\mathcal{N}^{ND}$ is a conjunction of predicates of the form $\psi_j\wedge(\bigwedge_{h\in\{1,\ldots,\ell\},h\neq j}\neg\psi_h)$, it is easy to see that every minterm of $\mathcal{N}^N$ is a subpredicate of some minterm of $\mathcal{N}^{ND}$. Because the minterms of $\mathcal{N}$ and $\mathcal{N}^N$ are equal, and the same holds for the minterms of $\mathcal{N}^D$ and $\mathcal{N}^{ND}$, we get that every minterm of $\mathcal{N}$ is a subpredicate of some minterm of $\mathcal{N}^D$.

Furthermore, because every predicate occurring in any s-DFA that has only reachable states, is a subpredicate of some predicate of the minimal s-DFA (see the end of Section 2), it is implied that any minterm of $\mathcal{N}^D$ is a subpredicate of some minterm of the minimal s-DFA of L. Thus, any minterm of $\mathcal{N}$ is a subpredicate of some minterm of the minimal s-DFA of L.

Now, let us consider the case where an s-NFA $\mathcal{N}$ has some unreachable states. It was shown above that any minterm of the reachable part of $\mathcal{N}$ is a subpredicate of a minterm of the minimal s-DFA. It is clear that every minterm of $\mathcal{N}$ is a subpredicate of some minterm of the reachable part of $\mathcal{N}$. We conclude that any minterm of $\mathcal{N}$ is a subpredicate of some minterm of the minimal s-DFA of L. □

Proposition 3. *Let $\mathcal{N}$ be an s-NFA and let $\mathcal{D}$ be the minimal s-DFA of L. Any minterm of $\mathcal{D}$ is a disjunction of minterms of $\mathcal{N}$.*

Proof. The minterms of $\mathcal{N}$ partition Σ, and so do the minterms of $\mathcal{D}$. Since by Proposition 2, any minterm of $\mathcal{N}$ is a subpredicate of some minterm of $\mathcal{D}$, we conclude that any minterm of $\mathcal{D}$ is a disjunction of minterms of $\mathcal{N}$. □

Considering that the minterms of any s-NFA of L are a refinement of the minterms of the minimal s-DFA of L, we call the latter the (canonical) minterms of L. Denoting the reverse language of L by L^R, we can show the following:

Proposition 4. *The minterms of L and L^R are the same.*

Proof. Let $\mathcal{D}$ be the minimal s-DFA of L. By Theorem 1, the minimal s-DFA of L^R can be obtained by reversing $\mathcal{D}$ and determinizing the resulting s-NFA $\mathcal{D}^R$ to get $\mathcal{D}^{RD}$. The set of transition predicates of $\mathcal{D}^R$ is the same as the set of transition predicates of $\mathcal{D}$, because reversing an automaton does not change predicates. Similarly, we get that the minterms of $\mathcal{D}^R$ are the minterms of $\mathcal{D}$. The predicates of $\mathcal{D}^{RD}$ are formed by using Boolean operations on predicates of $\mathcal{D}^R$, and the resulting predicates are disjunctions of minterms of $\mathcal{D}^R$. Thus, the minterms of $\mathcal{D}^{RD}$ are disjunctions of the minterms of $\mathcal{D}$.

By a similar reasoning as above we can obtain that the minterms of $\mathcal{D}$ are disjunctions of the minterms of $\mathcal{D}^{RD}$. We conclude that the minterms of $\mathcal{D}$ and $\mathcal{D}^{RD}$ are the same, that is, the minterms of L and L^R are the same. □

We note that although a symbolic regular language L can be defined over an infinite alphabet, the set of minterms of any s-NFA of L is finite, because an s-NFA has a finite number of transitions. It is pointed out in [4, 5] that minterms can be used as an alphabet when adapting classical automata algorithms to the

symbolic setting. Based on the results above, every symbolic regular language has a minimal set of minterms which is unique (up to predicate equivalence), with the reverse language having the same set of minterms. This set of minterms can be used in place of a finite alphabet to define several kind of symbolic automata for a given language as will be shown in Section 6.

5 Quotients and Atoms of Symbolic Languages

Similarly to the case of regular languages, the *left quotient*, or simply *quotient*, of a symbolic regular language L by a word $w \in \Sigma^*$ is the language $w^{-1}L = \{x \in \Sigma^* \mid wx \in L\}$. There is one *initial* quotient, $\varepsilon^{-1}L = L$. A quotient is *final* if it contains ε. Left quotients of L are the languages of the states of the minimal s-DFA of L.

Atoms of regular languages were introduced in [1] as non-empty intersections of complemented or uncomplemented quotients of the language. In [10] it was shown that atoms are the left congruence classes of the language. In the same way, we can define atoms of symbolic regular languages. For a symbolic regular language L, the *left congruence* ${}_L\!\equiv$ of L is defined as follows: for $x, y \in \Sigma^*$, $x\,{}_L\!\equiv y$ if for every $u \in \Sigma^*$, $ux \in L$ if and only if $uy \in L$. An *atom* of L is a left congruence class of L. Thus, an atom is a set of words which belong exactly to the same quotients. That is, an atom of a language L with quotients $K_1, \ldots, K_n$ is any non-empty language of the form $\widetilde{K_1} \cap \cdots \cap \widetilde{K_n}$, where $\widetilde{K_i}$ is either K_i or $\overline{K_i}$, and $\overline{K_i}$ is the complement of K_i with respect to Σ^*. It is easy to see that every quotient K_i is a union of atoms. An atom is *initial* if it has L (rather than $\overline{L}$) as a term; it is *final* if it contains ε. There is exactly one final atom, the atom $\widehat{K_1} \cap \cdots \cap \widehat{K_n}$, where $\widehat{K_i} = K_i$ if $\varepsilon \in K_i$, and $\widehat{K_i} = \overline{K_i}$ otherwise.

For any s-NFA $\mathcal{N} = (B, Q, \Delta, I, F)$ with a state set $Q = \{q_1, \ldots, q_n\}$ we can define a language equation for each state q_i as

$$L_i = \bigcup_{(q_i, \varphi, q_j) \in \Delta} [\![\varphi]\!] L_j \cup L_i^{\varepsilon}, \quad i = 1, \ldots, n, \tag{1}$$

where $L_i^{\varepsilon} = \{\varepsilon\}$ if $q_i \in F$, and $L_i^{\varepsilon} = \emptyset$ otherwise. This is similar to the way the language equations were defined for NFAs in [1]. Also similarly to what was done in [1], we can obtain equations for atoms of L, using the language equations of the minimal s-DFA of L. Namely, because quotients are the languages of the states of the minimal s-DFA, and atoms are intersections of complemented or uncomplemented quotients, we can express atoms by taking intersections of the right sides of the equations of the minimal s-DFA, or their negations.

We consider the minimal s-DFA $\mathcal{D} = (B, Q, \Delta, I, F)$ of L, with a state set $Q = \{q_1, \ldots, q_n\}$. Since the language of any state q_i of $\mathcal{D}$ is some quotient K_i, the equations

$$K_i = \bigcup_{(q_i, \varphi_{i_j}, q_j) \in \Delta} [\![\varphi_{i_j}]\!] K_j \cup K_i^{\varepsilon}, \quad i = 1, \ldots, n, \tag{2}$$

hold, where $K_i^\varepsilon = \{\varepsilon\}$ if $\varepsilon \in K_i$, and $K_i^\varepsilon = \emptyset$ otherwise. Because any atom A_h can be presented as an intersection $A_h = \bigcap_{i\in S_h} K_i \cap \bigcap_{i\in\overline{S_h}} \overline{K_i}$, where $S_h \subseteq \{1,\dots,n\}$ and $\overline{S_h} = \{1,\dots,n\} \setminus S_h$, we can compute the language equation for A_h from the following expression:

$$A_h = \bigcap_{i\in S_h} \Big(\bigcup_{(q_i,\varphi_{i_j},q_j)\in\Delta} [\![\varphi_{i_j}]\!] K_j \cup K_i^\varepsilon \Big) \cap \bigcap_{i\in\overline{S_h}} \overline{\Big(\bigcup_{(q_i,\varphi_{i_j},q_j)\in\Delta} [\![\varphi_{i_j}]\!] K_j \cup K_i^\varepsilon \Big)}. \quad (3)$$

Since atoms are intersections of complemented or uncomplemented quotients, we can convert formula (3), similarly to how a logical formula is converted into its full disjunctive normal form, into the expression

$$A_h = [\![\varphi_{h_1}]\!] A_{h_1} \cup \cdots \cup [\![\varphi_{h_k}]\!] A_{h_k} \cup A_h^\varepsilon, \quad (4)$$

where $\varphi_{h_1},\dots,\varphi_{h_k}$ are obtained by applying Boolean operations on the predicates appearing in (3), $A_{h_1},\dots,A_{h_k}$ are some atoms of L, and $A_h^\varepsilon = \{\varepsilon\}$ if $\varepsilon \in A_h$, and $A_h^\varepsilon = \emptyset$ otherwise. Clearly, $\varphi_{h_1},\dots,\varphi_{h_k}$ are disjunctions of minterms of L. Based on this observation, we can state the following proposition:

Proposition 5. *Let φ be a minterm of L. If $aA_j \subseteq A_i$ holds for some $a \in [\![\varphi]\!]$ and atoms A_i, A_j of L, then $[\![\varphi]\!]A_j \subseteq A_i$ holds.*

More generally, we show the following:

Proposition 6. *Let φ be a minterm of L. If $aL_j \subseteq L_i$ holds for some $a \in [\![\varphi]\!]$ and unions of atoms L_i, L_j of L, then $[\![\varphi]\!]L_j \subseteq L_i$ holds.*

Proof. Let $aL_j \subseteq L_i$ hold for some $a \in [\![\varphi]\!]$ and languages L_i, L_j consisting of unions of atoms of L. Then for every $A_h \subseteq L_j$ there is an atom $A_g \subseteq L_i$ such that $aA_h \subseteq A_g$. By Proposition 5, for every $A_h \subseteq L_j$ there is an atom $A_g \subseteq L_i$ such that $[\![\varphi]\!]A_h \subseteq A_g$ holds. Therefore, $[\![\varphi]\!]L_j \subseteq L_i$ holds. □

We will make use of Proposition 6 in the next section, where we define several s-NFAs for a given language.

6 Generating Symbolic Automata

In this section we consider the symbolic version of a method presented in [14] for generating NFAs from a set of languages. Similarly as in [14], we show that with this method we can define symbolic versions of some known NFAs. For the method to be able to work in the symbolic setting, minterms of a symbolic language prove to be very useful.

Let L be a symbolic regular language. We define a set $\{L_1,\dots,L_k\}$ of languages to be a *cover* of L, if every quotient K_j of L is a union of some L_i's. We say that a cover is *atomic* if every L_i is a union of atoms of L. We note that since L is the quotient of itself by the empty word ε, L is a union of some L_i's. We define the s-NFA based on an atomic cover $\{L_1,\dots,L_k\}$ as follows:

Definition 1. *The s-NFA generated by an atomic cover* $\{L_1, \ldots, L_k\}$ *of* L *is defined by* $\mathcal{G} = (B, Q, \Delta, I, F)$*, where* $Q = \{q_1, \ldots, q_k\}$*,* $I = \{q_i \mid L_i \subseteq L\}$*,* $F = \{q_i \mid \varepsilon \in L_i\}$*, and* $(q_i, \varphi, q_j) \in \Delta$ *if and only if* $[\![\varphi]\!]L_j \subseteq L_i$ *for* $q_i, q_j \in Q$ *and a minterm* φ *of* L*.*

Next, we present some properties of an s-NFA $\mathcal{G} = (B, Q, \Delta, I, F)$ generated by an atomic cover $\{L_1, \ldots, L_k\}$ of L. These results, originally presented for NFAs and general covers in [14], also fit into the symbolic setting. Proofs can be found in [14]; in the symbolic version, only minor adjustments are needed.

Proposition 7. *The following properties hold for s-NFA* $\mathcal{G}$*:*

1. $L_{q_i}(\mathcal{G}) \subseteq L_i$ *for every* $q_i \in Q$*.*
2. $L(\mathcal{G}) \subseteq L$*.*

We note that because of Proposition 6, it holds for every pair L_i, L_j of Definition 1 that whenever the inclusion $L_j \subseteq a^{-1}L_i$ holds for some $a \in \Sigma$, there is a transition (q_i, φ, q_j) of $\mathcal{G}$ such that $a \in [\![\varphi]\!]$. This property ensures that the following proposition holds:

Proposition 8. *The equality* $L_{q_i}(\mathcal{G}) = L_i$ *holds for every* $q_i \in Q$ *if and only if* $a^{-1}L_i$ *is a union of* L_j*'s for every* L_i *and* $a \in \Sigma$*.*

Next property easily follows from Proposition 8:

Proposition 9. *If* $a^{-1}L_i$ *is a union of* L_j*'s for every* L_i *and* $a \in \Sigma$*, then* $\mathcal{G}$ *accepts* L*.*

A simple example of an atomic cover is the set $A = \{A_1, \ldots, A_m\}$ of atoms of L, where A_m is the final atom. We can define a symbolic version of the NFA called the *átomaton* [1], as follows:

Definition 2. *The* symbolic átomaton *of* L *is the s-NFA* $\mathcal{A} = (B, Q, \Delta, I, \{q_m\})$*, where* $Q = \{q_1, \ldots, q_m\}$*,* $I = \{q_i \mid A_i \subseteq L\}$*, and* $(q_i, \varphi, q_j) \in \Delta$ *if and only if* $[\![\varphi]\!]A_j \subseteq A_i$ *for* $A_i, A_j \in A$ *and a minterm* φ *of* L*.*

It is known that for every atom A_i and $a \in \Sigma$, $a^{-1}A_i$ is a union of atoms [1]. Thus, by Proposition 8 it holds that $L_{q_i}(\mathcal{A}) = A_i$ for every $q_i \in Q$, and it follows from Proposition 9 that $L(\mathcal{A}) = L$. Also similarly to the classical case in [1], one can see that the predicate-normalized version of $\mathcal{A}^R$ is a minimal s-DFA of the reverse language L^R.

As another example of an atomic cover, consider the set $K' = \{K'_1, \ldots, K'_k\}$ of *prime* quotients of L, that is, those non-empty quotients of L which are not unions of other quotients. Based on this cover, we define an s-NFA as follows:

Definition 3. *The* canonical symbolic residual finite state automaton (canonical s-RFSA) *of* L *is the s-NFA* $\mathcal{R} = (B, Q, \Delta, I, F)$*, where* $Q = \{q_1, \ldots, q_k\}$*,* $I = \{q_i \mid K'_i \subseteq L\}$*, and* $(q_i, \varphi, q_j) \in \Delta$ *if and only if* $[\![\varphi]\!]K'_j \subseteq K'_i$ *for* $K'_i, K'_j \in K'$ *and a minterm* φ *of* L*.*

Since every quotient of L is a union of some prime quotients of L, for every prime quotient K'_i and $a \in \Sigma$, $a^{-1}K'_i$ is a union of prime quotients. In the same way as in the example above, one can see that the right language of a state $q_i \in Q$ is some prime quotient K'_i, and that $L(\mathcal{R}) = L$. The classical NFA version of $\mathcal{R}$ is known as the *canonical residual finite state automaton* [7] of a language. *Residual finite state automata (RFSAs)* are NFAs where the languages of its states are some *residuals*, that is, quotients of the language. Some properties of RFSAs in the learning context have been studied in [8]. It would be interesting to study symbolic versions of these automata as well.

6.1 Generating Minimal s-NFAs

We show that atomic covers and minterms of the language can be used to find minimal s-NFAs. Our approach here is similar to the way of finding minimal NFAs in [14].

Let $\mathcal{N} = (B, Q, \Delta, I, F)$ be a trim s-NFA accepting a symbolic regular language L, with $Q = \{q_1, \ldots, q_k\}$. For every state q_i of $\mathcal{N}$, we define a language $C_i = \bigcap_{L_{q_i}(\mathcal{N}) \subseteq K_h} K_h$ as the intersection of all quotients of L which contain the right language of q_i as a subset. Clearly, the inclusion $L_{q_i}(\mathcal{N}) \subseteq C_i$ holds. Since every quotient is a union of atoms, C_i is also a union of atoms. Because the set of right languages of the states of $\mathcal{N}$ obviously forms a cover for L, the set of C_i's has the same property. We note that there may be some states q_i and q_j of $\mathcal{N}$, such that $q_i \neq q_j$, but $C_i = C_j$. Let the set of distinct C_i's be C.

Let $\mathcal{G}_C = (B, Q_C, \Delta_C, I_C, F_C)$ be the s-NFA generated by the cover C for the language L. We note that $|Q_C| \leq |Q|$. Let $\pi : Q \to Q_C$ be the mapping assigning to state q_i of $\mathcal{N}$, the state q_{C_i} of $\mathcal{G}_C$, corresponding to C_i.

Proposition 10. *The mapping π is a morphism from $\mathcal{N}$ into $\mathcal{G}_C$.*

Proof. First, if $q_i \in I$, then $L_{q_i}(\mathcal{N}) \subseteq K_1$, where $K_1 = L$. Since the inclusion $C_i \subseteq K_1$ holds, q_{C_i} is initial, that is, $\pi(q_i) \in I_C$.

Similarly, if $q_i \in F$, then $\varepsilon \in L_{q_i}(\mathcal{N})$, implying that $\varepsilon \in C_i$, and thus $q_{C_i} \in F_C$, that is, $\pi(q_i) \in F_C$.

We also show that for all states $q_i, q_j \in Q$ and $a \in \Sigma$, if $(q_i, \varphi, q_j) \in \Delta$ for some φ such that $a \in [\![\varphi]\!]$, then there is some φ' such that $(\pi(q_i), \varphi', \pi(q_j)) \in \Delta_C$ and $a \in [\![\varphi']\!]$. Let $(q_i, \varphi, q_j) \in \Delta$ such that $a \in [\![\varphi]\!]$ for some $q_i, q_j \in Q$ and $a \in \Sigma$. Then it holds that $L_{q_j}(\mathcal{N}) \subseteq a^{-1}L_{q_i}(\mathcal{N}) \subseteq a^{-1}C_i = a^{-1}\bigcap_{L_{q_i}(\mathcal{N}) \subseteq K_h} K_h = \bigcap_{L_{q_i}(\mathcal{N}) \subseteq K_h} a^{-1}K_h$. Therefore, $L_{q_j}(\mathcal{N})$ is a subset of some quotients $a^{-1}K_h$ such that $L_{q_i}(\mathcal{N}) \subseteq K_h$, implying that $C_j \subseteq \bigcap_{L_{q_i}(\mathcal{N}) \subseteq K_h} a^{-1}K_h$, that is, $C_j \subseteq a^{-1}C_i$. Thus, $aC_j \subseteq C_i$, and by Proposition 6, $[\![\varphi']\!]C_j \subseteq C_i$, where φ' is a minterm of L such that $a \in [\![\varphi']\!]$. It is implied that $(q_{C_i}, \varphi', q_{C_j}) \in \Delta_C$, that is, $(\pi(q_i), \varphi', \pi(q_j)) \in \Delta_C$.

We conclude that π is a morphism from $\mathcal{N}$ into $\mathcal{G}_C$. □

Corollary 1. *For every state q_i of $\mathcal{N}$, the inclusion $L_{q_i}(\mathcal{N}) \subseteq L_{q_{C_i}}(\mathcal{G}_C)$ holds. Also, $L(\mathcal{G}_C) = L$.*

Proof. The morphism $\pi : Q \to Q_C$ implies that for every $q_i \in Q$, the inclusion $L_{q_i}(\mathcal{N}) \subseteq L_{q_{C_i}}(\mathcal{G}_C)$ holds, and also that $L(\mathcal{N}) \subseteq L(\mathcal{G}_C)$ holds.

Since $L(\mathcal{N}) = L$, and by Proposition 7, $L(\mathcal{G}_C) \subseteq L$, we conclude that $L(\mathcal{G}_C) = L$. □

Now, let L_i be a union of some atoms of L. We define the *maximized* version of L_i to be the language $max(L_i) = \bigcap_{L_i \subseteq K_h} K_h$, that is, the intersection of all quotients which contain L_i as a subset. Since any quotient is a disjoint union of atoms, $max(L_i)$ is a union of atoms.

Based on the results above, we can obtain a minimal s-NFA of L as follows: we find an atomic cover $\{L_1, \ldots, L_k\}$ of L, consisting of the minimal number of languages L_i, then maximize L_i's to get the atomic cover $\{max(L_1), \ldots, max(L_k)\}$, and generate an s-NFA $\mathcal{G}$ using this maximized cover. If $\mathcal{G}$ accepts L, then $\mathcal{G}$ is a minimal s-NFA of L, otherwise it is not, and we try other covers in the order of increasing size until we generate an s-NFA which accepts L. The first such generated s-NFA is a minimal s-NFA for L. We note that a minimal cover and a minimal s-NFA are not necessarily unique.

7 Generalization of Brzozowski's Theorem

In this section we present a generalization of Theorem 1, which is similar to its classical version in [1]. Namely, we characterize the class of s-NFAs for which determinization produces a minimal s-DFA. The approach we take here is similar to what was used in [13] to prove an analogous result about obtaining a canonical RFSA.

First, we show the following proposition:

Proposition 11. *Given an s-NFA $\mathcal{N} = (B, Q, \Delta, I, F)$, the left language of a state s of $\mathcal{N}^D$ is $L_{\{I\},s}(\mathcal{N}^D) = \bigcap_{q \in s} L_{I,q}(\mathcal{N}) \cap \bigcap_{q \notin s} \overline{L_{I,q}(\mathcal{N})}$.*

Proof. Let $\mathcal{N} = (B, Q, \Delta, I, F)$ be an s-NFA. Let s be a state of $\mathcal{N}^D$ and let $w \in L_{\{I\},s}(\mathcal{N}^D)$ be a word in the left language of s. We prove the proposition by induction on the length of w. If $w = \varepsilon$, then $s = I$ is the initial state of $\mathcal{N}^D$. Also, since $\varepsilon \in L_{I,q}(\mathcal{N})$ for every $q \in I$, and $\varepsilon \notin L_{I,q}(\mathcal{N})$ for every $q \notin I$, it is clear that $\varepsilon \in L_{\{I\},s}(\mathcal{N}^D)$ if and only if $\varepsilon \in \bigcap_{q \in s} L_{I,q}(\mathcal{N}) \cap \bigcap_{q \notin s} \overline{L_{I,q}(\mathcal{N})}$.

Now, let $w = ua$ with $u \in \Sigma^*$ and $a \in \Sigma$. Let us assume that $u \in L_{\{I\},s'}(\mathcal{N}^D)$ holds for a state s' of $\mathcal{N}^D$ if and only if $u \in \bigcap_{q \in s'} L_{I,q}(\mathcal{N}) \cap \bigcap_{q \notin s'} \overline{L_{I,q}(\mathcal{N})}$. If $w \in L_{\{I\},s}(\mathcal{N}^D)$, then there is a state s' of $\mathcal{N}^D$, such that $u \in L_{\{I\},s'}(\mathcal{N}^D)$ and there is a transition $(s', \varphi_{s',s}, s)$ in $\mathcal{N}^D$ with $a \in [\![\varphi_{s',s}]\!]$. According to how $\mathcal{N}^D$ is constructed, $\varphi_{s',s} = (\bigwedge_{q \in s} \varphi_{s',q}) \wedge (\bigwedge_{q \in S_{s'} \setminus s} \neg \varphi_{s',q})$, where $\varphi_{s',q} = \bigvee_{(p,\varphi,q) \in \Delta, p \in s'} \varphi$ and $S_{s'}$ is the set of states q of $\mathcal{N}$ such that there is a transition from some state $p \in s'$ to q. Since by the induction assumption $u \in L_{I,q}(\mathcal{N})$ holds for $q \in s'$ and $u \notin L_{I,q}(\mathcal{N})$ holds for $q \notin s'$, it is clear that $ua \in L_{I,q}(\mathcal{N})$ holds for $q \in s$ and $ua \notin L_{I,q}(\mathcal{N})$ holds for $q \notin s$. This is equivalent to that $ua \in \bigcap_{q \in s} L_{I,q}(\mathcal{N}) \cap \bigcap_{q \notin s} \overline{L_{I,q}(\mathcal{N})}$. □

Now, let us consider the minimal s-DFA $\mathcal{D} = (B, S, \Gamma, \{s_1\}, S_f)$ of a symbolic regular language L, with a state set $S = \{s_1, \ldots, s_n\}$. Let $L_i = L_{\{s_1\},s_i}(\mathcal{D})$ be the left language of a state s_i, for $i = 1, \ldots, n$. It is easy to see that $L_i \cap L_j = \emptyset$ if $s_i \neq s_j$. Also, it is clear that for any s-DFA $\mathcal{D}'$ of L, there is a one-to-many correspondence between the L_i's and the states of $\mathcal{D}'$, such that every L_i is the union of the left languages of the corresponding states of $\mathcal{D}'$. We note that only if $\mathcal{D}'$ is minimal, this correspondence is one-to-one.

The following proposition holds:

Proposition 12. *For an s-NFA $\mathcal{N}$, $\mathcal{N}^D$ is minimal if and only if every left language of $\mathcal{N}$ is a union of L_i's.*

Proof. Let $\mathcal{N} = (B, Q, \Delta, I, F)$ be an s-NFA such that its determinized version $\mathcal{N}^D$ is a minimal s-DFA. Then for any state s_i of $\mathcal{N}^D$, the left language of s_i is L_i. Suppose that there is a state q_j of $\mathcal{N}$ such that its left language is not a union of L_i's. That is, for some word $u \in L_h$, $u \in L_{I,q_j}(\mathcal{N})$, but $L_h \not\subseteq L_{I,q_j}(\mathcal{N})$. Let s_u be a state of $\mathcal{N}^D$ such that $u \in L_{\{I\},s_u}(\mathcal{N}^D)$. Since $\mathcal{N}^D$ is minimal, we know that $L_{\{I\},s_u}(\mathcal{N}^D) = L_h$. By Proposition 11, we also know that $q_j \in s_u$ and $L_{\{I\},s_u}(\mathcal{N}^D) \subseteq L_{I,q_j}(\mathcal{N})$. Therefore the inclusion $L_h \subseteq L_{I,q_j}(\mathcal{N})$ holds, a contradiction.

Conversely, let all the left languages of $\mathcal{N}$ be unions of L_i's. Since by Proposition 11 any left language of $\mathcal{N}^D$ is a Boolean combination of some left languages of $\mathcal{N}$, and because it is obvious that any left language of $\mathcal{N}^D$ is a subset of some L_i, we conclude that the left languages of $\mathcal{N}^D$ are exactly L_i's. Thus, $\mathcal{N}^D$ is minimal. □

Similarly as in [1], we say that an s-NFA $\mathcal{N} = (B, Q, \Delta, I, F)$ is *atomic* if for every $q \in Q$, $L_q(\mathcal{N})$ is a union of atoms of $L(\mathcal{N})$.

Now we can state the following theorem:

Theorem 2. *For any s-NFA $\mathcal{N}$, $\mathcal{N}^D$ is minimal if and only if $\mathcal{N}^R$ is atomic.*

Proof. By properties of the symbolic átomaton (see Section 6), atoms of a language are equal to the reversed left languages of the states of a minimal s-DFA of the reverse language. Therefore, $\mathcal{N}^R$ is atomic if and only if every left language of $\mathcal{N}$ is a union of L_i's. We conclude by Proposition 12 that $\mathcal{N}^D$ is minimal if and only if $\mathcal{N}^R$ is atomic. □

8 Related and Future Work

The work in [12] discusses the use of symbolic regular expressions over large but finite alphabets in the context of using regular expression derivatives [3] for matching. The paper does not use minterms to create predicates but a regular expression derivative induced equivalence relation over characters to create predicates. The paper states that in general it is not possible to compute such predicates without doing work that depends linearly on the size of the alphabet. This is clearly not possible if the alphabet is infinite. An interesting future work would be to extend the derivative based approach to symbolic regular expressions over arbitrary symbolic alphabets.

References

1. Brzozowski, J., Tamm, H.: Theory of átomata. Theor. Comput. Sci. **539** (2014) 13–27
2. Brzozowski, J.: Canonical regular expressions and minimal state graphs for definite events. In: Proceedings of the Symposium on Mathematical Theory of Automata. Volume 12 of MRI Symposia Series, Polytechnic Press, Polytechnic Institute of Brooklyn, N.Y. (1963) 529–561
3. Brzozowski, J.A.: Derivatives of regular expressions. J. ACM **11**(4) (1964) 481–494
4. D'Antoni, L., Veanes, M.: The power of symbolic automata and transducers. In: Computer Aided Verification, 29th International Conference (CAV 2017)
5. D'Antoni, L., Veanes, M.: Minimization of symbolic automata. In: The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014. (2014) 541–554
6. D'Antoni, L., Veanes, M.: Forward bisimulations for nondeterministic symbolic finite automata. In: Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, Part I. (2017) 518–534
7. Denis, F., Lemay, A., Terlutte, A.: Residual finite state automata. Fund. Inform. **51** (2002) 339–368
8. Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using RFSAs. Theor. Comput. Sci. **313**(2) (2004) 267–294
9. Drews, S., D'Antoni, L.: Learning symbolic automata. In: Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, Part I. (2017) 173–189
10. Iván, S.: Complexity of atoms, combinatorially. Inform. Process. Lett. **116** (2016) 356–360
11. Keil, M., Thiemann, P.: Symbolic solving of extended regular expression inequalities. In: 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India. (2014) 175–186
12. Owens, S., Reppy, J., Turon, A.: Regular-expression derivatives re-examined. Journal of Functional Programming **19**(2) (2009) 173–190
13. Tamm, H.: Generalization of the double-reversal method of finding a canonical residual finite state automaton. In: Descriptional Complexity of Formal Systems - 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25-27, 2015. Proceedings. (2015) 268–279
14. Tamm, H.: New interpretation and generalization of the Kameda-Weiner method. In: 43rd Int. Colloq. on Automata, Languages, and Programming (ICALP 2016). Volume 55 of Leibniz Int. Proc. in Informatics (LIPIcs), Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2016) 116:1–116:12
15. Thompson, K.: Regular expression search algorithm. Commun. ACM **11**(6) (1968) 419–422
16. Veanes, M., de Halleux, P., Tillmann, N.: Rex: Symbolic regular expression explorer. In: Third International Conference on Software Testing, Verification and Validation, ICST 2010, IEEE Computer Society (2010) 498–507
17. Watson, B.W.: A taxonomy of finite automata construction algorithms. Computing science report 93/43, Eindhoven University of Technology (1995)
18. Watson, B.W.: Implementing and using finite automata toolkits. In: Extended finite state models of language, Cambridge University Press (1999) 19–36