Syntactic Minimization Of Nondeterministic Finite Automata

Robert S. R. Myers
London, United Kingdom

Henning Urbat
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract

Nondeterministic automata may be viewed as succinct programs implementing deterministic automata, i.e. complete specifications. Converting a given deterministic automaton into a small nondeterministic one is known to be computationally very hard; in fact, the ensuing decision problem is PSPACE-complete. This paper stands in stark contrast to the status quo. We restrict attention to *subatomic* nondeterministic automata, whose individual states accept unions of syntactic congruence classes. They are general enough to cover almost all structural results concerning nondeterministic state-minimality. We prove that converting a monoid recognizing a regular language into a small subatomic acceptor corresponds to an NP-complete problem. The NP certificates are solutions of simple equations involving relations over the syntactic monoid. We also consider the subclass of *atomic* nondeterministic automata introduced by Brzozowski and Tamm. Given a deterministic automaton and another one for the reversed language, computing small atomic acceptors is shown to be NP-complete with analogous certificates. Our complexity results emerge from an algebraic characterization of (sub)atomic acceptors in terms of deterministic automata with semilattice structure, combined with an equivalence of categories leading to succinct representations.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases Algebraic language theory, Nondeterministic automata, NP-completeness

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.78

Related Version *Full Version*: `http://arxiv.org/abs/2107.03229`

Funding *Henning Urbat*: Supported by Deutsche Forschungsgemeinschaft (DFG) under project SCHR 1118/15-1.

1 Introduction

Regular languages arise from a multitude of different perspectives: operationally via finite-state machines, model-theoretically via monadic second-order logic, and algebraically via finite monoids. In practice, *deterministic* finite automata (dfas) and *nondeterministic* finite automata (nfas) are two of the most common representations. Although the former may be exponentially larger than the latter, there is no known efficient procedure for converting dfas into small nfas, e.g. state-minimal ones. Jiang and Ravikumar proved the corresponding decision problem (does an equivalent nfa with a given number of states exist?) to be PSPACE-complete [14, 15], suggesting that exhaustively enumerating candidates is necessary. One possible strategy towards tractability is to restrict the target automata to suitable *subclasses* of nfas. The challenge is to identify subclasses permitting more efficient computation (e.g. lowering the PSPACE bound to an NP bound, enabling the use of SAT solvers), while still being general enough to cover succinct acceptors of regular languages.

In our present paper we will show that the class of *subatomic nfas* naturally meets the above requirements. An nfa accepting the language L is subatomic if each individual state accepts a union of syntactic congruence classes of L. In recent work [26] we observed that almost all known results on the structure of small nfas, e.g. for unary [6, 13], bideterministic [30],

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 78; pp. 78:1–78:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

topological [1] and biRFSA languages [19], implicitly construct small subatomic nfas. This firmly indicates that the latter form a rich class of acceptors despite their seemingly restrictive definition, i.e. in many settings computing small nfas amounts to computing small subatomic ones. Restricting to subatomic nfas yields useful additional structure; in fact, their theory is tightly linked to the algebraic theory of regular languages and the representation theory of monoids. This suggests an *algebraic* counterpart of the dfa to nfa conversion problem: given a finite monoid recognizing some regular language, compute an equivalent small subatomic nfa. Denoting its decision version (does an equivalent subatomic nfa with a given number of states exist?) by $\mathbf{MON} \to \mathbf{NFA_{syn}}$, our main result is:

▶ **Theorem.** The problem $\mathbf{MON} \to \mathbf{NFA_{syn}}$ is NP-complete.

In addition we also investigate *atomic nfas*, a subclass of subatomic nfas earlier introduced by Brzozowski and Tamm [4]. Similar to the subatomic case, their specific structure naturally invokes the problem of converting a pair of dfas accepting mutually reversed languages into a small atomic nfa. Denoting its decision version by $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$, we get:

▶ **Theorem.** The problem $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$ is NP-complete.

The short certificates witnessing that both problems are in NP are solutions of *equations* involving relations over the syntactic congruence or the Nerode left congruence, respectively.

The above two theorems sharply contrast the PSPACE-completeness of the general dfa to nfa conversion problem, but also previous results on its sub-PSPACE variants. The latter are either concerned with particular regular languages such as finite or unary ones [11, 13], or with target nfas admitting only very weak forms of nondeterminism, such as unambiguous automata [15] or dfas with multiple initial states [22]. In contrast, our present work applies to all regular languages and the restriction to (sub)atomic nfas is a purely *semantic* one.

Our results are fundamentally based upon a category-theoretic perspective on atomic and subatomic acceptors. At its heart are two equivalences of categories as indicated below:

$$\mathbf{JSL_f}^{\mathsf{op}} \xleftrightarrow[\text{Structure theory}]{\simeq} \mathbf{JSL_f} \xleftrightarrow[\text{Complexity theory}]{\simeq} \mathbf{Dep}.$$

As shown in [26], the *structure theory* of (sub)atomic nfas emerges by interpreting them as dfas endowed with semilattice structure, and relating them to their dual automata under the familiar self-duality of the category $\mathbf{JSL_f}$ of finite semilattices. Similarly, the *complexity theory* of (sub)atomic nfas developed in the present paper rests on the equivalence between $\mathbf{JSL_f}$ and a category $\mathbf{Dep}$ (see Definition 3.1) that yields succinct relational representations of finite semilattices by their irreducible elements. To derive the NP-completeness theorems, we reinterpret semilattice automata associated to (sub)atomic nfas inside $\mathbf{Dep}$. We regard this conceptually simple and natural categorical approach as a key contribution of our paper.

2 Atomic and Subatomic NFAs

We start by setting up the notation and terminology used in the rest of the paper, including the key concept of a (sub)atomic nfa that underlies our complexity results. Readers are assumed to be familiar with basic category [21].

Semilattices. A *(join-)semilattice* is a poset $(S, \leq_S)$ in which every finite subset $X \subseteq S$ has a least upper bound (a.k.a. join) $\bigvee X$. A *morphism* between semilattices is a map preserving finite joins. If S is finite as we often assume, every subset $X \subseteq S$ also has a greatest lower bound (a.k.a. meet) $\bigwedge X$, given by the join of its lower bounds. In particular, S has a least element $\bot_S = \bigvee \emptyset$ and a greatest element $\top_S = \bigwedge \emptyset$. An element $j \in S$ is *join-irreducible* if $j = \bigvee X$ implies $j \in X$ for every subset $X \subseteq S$. Dually, $m \in S$ is *meet-irreducible* if $m = \bigwedge X$ implies $m \in X$. We put

$$J(S) = \{\, j \in S \;:\; j \text{ is join-irreducible} \,\} \quad \text{and} \quad M(S) = \{\, m \in S \;:\; m \text{ is meet-irreducible} \,\}.$$

Note $\bot_S \notin J(S)$ and $\top_S \notin M(S)$. The join-irreducibles form the least set of *join-generators* of S, i.e. every element of S is a join of elements from $J(S)$, and every other subset $J \subseteq S$ with that property contains $J(S)$. Dually, $M(S)$ is the least set of *meet-generators* of S.

Let $\mathbb{2} = \{0, 1\}$ be the two-element semilattice with $0 \leq 1$. Morphisms $i \colon \mathbb{2} \to S$ correspond to elements of S via $i \mapsto i(1)$. Morphisms $f \colon S \to \mathbb{2}$ correspond to *prime filters* via $f \mapsto f^{-1}[1]$. If S is finite, these are precisely the subsets $F_{s_0} = \{s \in S : s \not\leq_S s_0\}$ for any $s_0 \in S$.

We denote by **JSL** the category of join-semilattices and their morphisms. Its full subcategory $\mathbf{JSL_f}$ of finite semilattices is self-dual [17]: there is an equivalence functor

$$\mathbf{JSL_f^{op}} \xrightarrow{\simeq} \mathbf{JSL_f}$$

mapping $(S, \leq_S)$ to the *opposite semilattice* $S^{\mathsf{op}} = (S, \geq_S)$ obtained by reversing the order, and a morphism $f \colon S \to T$ to the morphism $f_* \colon T^{\mathsf{op}} \to S^{\mathsf{op}}$ sending $t \in T$ to the $\leq_S$-greatest element $s \in S$ with $f(s) \leq_T t$. Thus, f and f_* satisfy the adjoint relationship

$$f(s) \leq_T t \qquad \text{iff} \qquad s \leq_S f_*(t)$$

for all $s \in S$ and $t \in T$. The morphism f is injective (equivalently a $\mathbf{JSL_f}$-monomorphism) iff f_* is surjective (equivalently a $\mathbf{JSL_f}$-epimorphism).

Relations. A *relation* between sets X and Y is a subset $\mathcal{R} \subseteq X \times Y$. We write $\mathcal{R}(x, y)$ if $(x, y) \in \mathcal{R}$. For $x \in X$ and $A \subseteq X$ we put

$$\mathcal{R}[x] = \{\, y \in Y : \mathcal{R}(x, y) \,\} \qquad \text{and} \qquad \mathcal{R}[A] = \bigcup_{x \in A} \mathcal{R}[x].$$

The *converse* of $\mathcal{R}$ is the relation $\breve{\mathcal{R}} \subseteq Y \times X$ (alternatively $\mathcal{R}^{\smile}$) where $\breve{\mathcal{R}}(y, x)$ iff $\mathcal{R}(x, y)$ for $x \in X$ and $y \in Y$. The *composite* of $\mathcal{R} \subseteq X \times Y$ and $\mathcal{S} \subseteq Y \times Z$ is the relation $\mathcal{R};\mathcal{S} \subseteq X \times Z$ where $\mathcal{R}(x, z)$ iff there exists $y \in Y$ with $\mathcal{R}(x, y)$ and $\mathcal{S}(y, z)$. Let **Rel** denote the category whose objects are sets and whose morphisms are relations with the above composition. The identity morphism on X is the identity relation $\mathsf{id}_X \subseteq X \times X$ with $\mathsf{id}_X(x, y)$ iff $x = y$.

A *biclique* of a relation $\mathcal{R} \subseteq X \times Y$ is subset of the form $B_1 \times B_2 \subseteq \mathcal{R}$, where $B_1 \subseteq X$ and $B_2 \subseteq Y$. A set $\mathcal{C}$ of bicliques forms a *biclique cover* if $\mathcal{R} = \bigcup \mathcal{C}$. The *bipartite dimension* of $\mathcal{R}$, denoted $\mathsf{dim}(\mathcal{R})$, is the minimum cardinality of any biclique cover.

Languages. Let Σ^* be the set of finite words over an alphabet Σ including the empty word ε. A *language* is a subset L of Σ^*. We let $\overline{L} = \Sigma^* \setminus L$ denote the *complement* and $L^{\mathsf{r}} = \{w^{\mathsf{r}} : w \in L\}$ the *reverse* of L, where $\varepsilon^{\mathsf{r}} = \varepsilon$ and $w^{\mathsf{r}} = a_n \ldots a_1$ for $w = a_1 \ldots a_n$. The *left derivatives* and *two-sided derivatives* of L are, respectively, given by $u^{-1}L = \{w \in \Sigma^* : uw \in L\}$ and $u^{-1}Lv^{-1} = \{w \in \Sigma^* : uwv \in L\}$ for $u, v \in \Sigma^*$; moreover for $U \subseteq \Sigma^*$ put $U^{-1}L = \bigcup_{u \in U} u^{-1}L$. For each fixed $L \subseteq \Sigma^*$, the following sets of languages will play a prominent role:

$$\mathsf{LD}(L) \subseteq \mathsf{SLD}(L) \subseteq \mathsf{BLD}(L) \subseteq \mathsf{BLRD}(L)$$

where $\mathsf{LD}(L) = \{u^{-1}L : u \in \Sigma^*\}$ is the set of left derivatives, and $\mathsf{SLD}(L)$, $\mathsf{BLD}(L)$, $\mathsf{BLRD}(L)$ denote its closure under finite unions, all set-theoretic boolean operations, and all set-theoretic boolean operations and two-sided derivatives, respectively. The final three form $\cup$-semilattices, and the final two are boolean algebras w.r.t. the set-theoretic operations.

A language L is *regular* if $\mathsf{LD}(L)$ is a finite set; then the other three sets are finite too. The finite semilattices $\mathsf{SLD}(L)$ and $\mathsf{SLD}(L^{\mathsf{r}})$ are related by the fundamental isomorphism

$$\mathrm{dr}_L\colon [\mathsf{SLD}(L^{\mathsf{r}})]^{\mathsf{op}} \xrightarrow{\cong} \mathsf{SLD}(L), \qquad K \mapsto (\overline{K^{\mathsf{r}}})^{-1}L, \tag{2.1}$$

see [26, Proposition 3.13]. Equivalently, the map dr_L sends $V^{-1}L^{\mathsf{r}} \in \mathsf{SLD}(L^{\mathsf{r}})$ to the largest element of $\mathsf{SLD}(L)$ disjoint from V^{r}. It is closely connected to the *dependency relation* of L,

$$\mathcal{DR}_L \subseteq \mathsf{LD}(L) \times \mathsf{LD}(L^{\mathsf{r}}), \qquad \mathcal{DR}_L(u^{-1}L, v^{-1}L^{\mathsf{r}}) \;:\!\iff\; uv^{\mathsf{r}} \in L \quad \text{for } u, v \in \Sigma^*. \tag{2.2}$$

In fact, by [26, Theorem 3.15] we have

$$\mathcal{DR}_L(u^{-1}L, v^{-1}L^{\mathsf{r}}) \iff u^{-1}L \nsubseteq \mathrm{dr}_L(v^{-1}L^{\mathsf{r}}) \qquad \text{for } u, v \in \Sigma^*. \tag{2.3}$$

Since the boolean algebra $\mathsf{BLD}(L)$ is generated by the left derivatives of L, its atoms (= join-irreducibles) are the congruence classes of the *Nerode left congruence* $\sim_L \subseteq \Sigma^* \times \Sigma^*$,

$$u \sim_L v \quad \text{iff} \quad \forall x \in \Sigma^* : u \in x^{-1}L \Leftrightarrow v \in x^{-1}L \quad \text{iff} \quad (u^{\mathsf{r}})^{-1}L^{\mathsf{r}} = (v^{\mathsf{r}})^{-1}L^{\mathsf{r}}. \tag{2.4}$$

Note that this relation is left-invariant, i.e. $u \sim_L v$ implies $wu \sim_L wv$ for all $w \in \Sigma^*$.

Similarly, the atoms of $\mathsf{BLRD}(L)$ are the congruence classes of the *syntactic congruence* $\equiv_L \subseteq \Sigma^* \times \Sigma^*$, i.e. the monoid congruence on the free monoid Σ^* defined by

$$u \equiv_L v \quad \text{iff} \quad \forall x, y \in \Sigma^* : u \in x^{-1}Ly^{-1} \Leftrightarrow v \in x^{-1}Ly^{-1}. \tag{2.5}$$

The quotient monoid $\mathsf{syn}(L) = \Sigma^*/{\equiv_L}$ is called the *syntactic monoid* of L, and the canonical map $\mu_L\colon \Sigma^* \twoheadrightarrow \mathsf{syn}(L)$ sending $u \in \Sigma^*$ to its congruence class $[u]_{\equiv_L}$ is the *syntactic morphism*.

Automata. Fix a finite alphabet Σ. A *nondeterministic finite automaton* (a.k.a. *nfa*) $N = (Q, \delta, I, F)$ consists of a finite set Q (the states), relations $\delta = (\delta_a \subseteq Q \times Q)_{a\in\Sigma}$ (the transitions), and sets $I, F \subseteq Q$ (the initial states and final states). We write $q_1 \xrightarrow{a} q_2$ whenever $q_2 \in \delta_a[q_1]$. The language $L(N, q)$ *accepted* by a state $q \in Q$ consists of all words $w \in \Sigma^*$ such that $\delta_w[q] \cap F \neq \emptyset$, where $\delta_w \subseteq Q \times Q$ is the extended transition relation $\delta_{a_1}; \ldots ; \delta_{a_n}$ for $w = a_1 \ldots a_n$ and $\delta_\varepsilon = \mathsf{id}_Q$. The language *accepted* by N is defined $L(N) = \bigcup_{i\in I} L(N, i)$.

An nfa N is a *deterministic finite automaton* (a.k.a. *dfa*) if $I = \{q_0\}$ is a singleton set and each transition relation is a function $\delta_a\colon Q \to Q$. A dfa is a $\mathbf{JSL}$-*dfa* if Q is a finite semilattice, each $\delta_a\colon Q \to Q$ is a semilattice morphism, and $F \subseteq Q$ forms a prime filter. It is often useful to represent a $\mathbf{JSL}$-dfa in terms of morphisms

$$\mathbb{2} \xrightarrow{i} Q \xrightarrow{\delta_a} Q \xrightarrow{f} \mathbb{2}$$

where i is the unique morphism with $i(1) = q_0$ and f is given by $f(q) = 1$ iff $q \in F$. A $\mathbf{JSL}$-dfa *morphism* from $A = (Q, \delta, i, f)$ to $A' = (Q', \delta', i', f')$ is a $\mathbf{JSL_f}$-morphism $h\colon Q \to Q'$ preserving transitions via $h \circ \delta_a = \delta'_a \circ h$, preserving the initial state via $i' = h \circ i$, and both preserving and reflecting the final states via $f = f' \circ h$. Equivalently, h is a dfa morphism that is also a semilattice morphism, so in particular $L(A) = L(A')$. If Q is a subsemilattice of Q' and $h\colon Q \rightarrowtail Q'$ is the inclusion map, then A is called a *sub* $\mathbf{JSL}$-*dfa* of A'.

Fix a regular language L. Viewed as a $\cup$-semilattice, $\mathsf{BLRD}(L)$ carries the structure of a $\mathbf{JSL}$-dfa with transitions $K \xrightarrow{a} a^{-1}K$, initial state L, and finals $\{K : \varepsilon \in K\}$. This restricts to sub $\mathbf{JSL}$-dfas $\mathsf{BLD}(L)$ and $\mathsf{SLD}(L)$. Moreover $\mathsf{LD}(L)$ forms a sub-dfa of $\mathsf{SLD}(L)$, well-known [5] to be the *state-minimal dfa* for L, so we denote it by $\mathsf{dfa}(L)$. The syntactic monoid $\mathsf{syn}(L)$ is isomorphic to the *transition monoid* of $\mathsf{dfa}(L)$, i.e. the monoid of all extended transition maps $\delta_w\colon \mathsf{LD}(L) \to \mathsf{LD}(L)$ ($w \in \Sigma^*$) with multiplication given by composition [27].

Analogously $\mathsf{SLD}(L)$ is the *state-minimal* $\mathbf{JSL}$*-dfa* for L. Up to isomorphism, it is the unique $\mathbf{JSL}$-dfa for L that is $\mathbf{JSL}$*-reachable* (i.e. every state is a join of states reachable from the initial state via transitions) and *simple* (i.e. distinct states accept distinct languages).

Nfas, dfas and $\mathbf{JSL}$-dfas are expressively equivalent and accept precisely the regular languages. In particular, to every $\mathbf{JSL}$-dfa $A = (Q, \delta, q_0, F)$ one can associate an equivalent nfa $J(A)$, the *nfa of join-irreducibles* [1, 2, 25]. Its states are given by the set $J(Q)$ of join-irreducibles of Q; for any $q_1, q_2 \in J(Q)$ and $a \in \Sigma$ there is a transition $q_1 \xrightarrow{a} q_2$ in $J(A)$ iff $q_2 \leq_Q \delta_a(q_1)$; a state $q \in J(Q)$ is initial iff $q \leq_S q_0$, and final iff $q \in F$. For any $q \in J(Q)$, we have $L(A, q) = L(J(A), q)$. The *canonical residual finite state automaton* [7] for a regular language L is given by $N_L = J(\mathsf{SLD}(L))$, the nfa of join-irreducibles of its minimal $\mathbf{JSL}$-dfa.

Atomic and subatomic nfas. An nfa accepting the language $L \subseteq \Sigma^*$ is called *atomic* [4] if each state accepts a language from $\mathsf{BLD}(L)$, and *subatomic* [26] if each state accepts a language from $\mathsf{BLRD}(L)$. The *nondeterministic atomic complexity* $\mathsf{natm}(L)$ of a regular language L is the least number of states of any atomic nfa accepting L. The *nondeterministic syntactic complexity* $\mathsf{nsyn}(L)$ is the least number of states of any subatomic nfa accepting L. Subatomic nfas are intimately connected to syntactic monoids: the atoms of $\mathsf{BLRD}(L)$ are the elements of $\mathsf{syn}(L)$, so an nfa accepting L is subatomic iff its individual states accept unions of syntactic congruence classes. Additionally $\mathsf{nsyn}(L)$ can be characterized via *boolean representations* of $\mathsf{syn}(L)$, i.e. monoid morphisms $\varrho\colon \mathsf{syn}(L) \to \mathbf{JSL_f}(S, S)$ into the endomorphisms of a finite semilattice [26]. For a detailed exposition we refer to *op. cit.*

These complexity measures are related to the *nondeterministic state complexity* $\mathsf{ns}(L)$, i.e. the least number of states of any (unrestricted) nfa accepting L. In particular,

$$\mathsf{dim}(\mathcal{DR}_L) \leq \mathsf{ns}(L) \leq \mathsf{nsyn}(L) \leq \mathsf{natm}(L). \tag{2.6}$$

The first inequality is due to Gruber and Holzer [10] (see also [26, Theorem 4.8] for a purely algebraic proof), while the others arise by restricting admissible nondeterministic acceptors.

Importantly, small atomic and subatomic nfas can be characterized in terms of $\mathbf{JSL}$-dfas. The following theorem involves two commuting diagrams of semilattice morphisms, whose lower and upper paths are the canonical $\mathbf{JSL}$-dfas described earlier.

▶ **Theorem 2.1.** *Let $L \subseteq \Sigma^*$ be a regular language.*

1. $\mathsf{natm}(L)$ *is the least number k such that there exists a finite semilattice S with $|J(S)| \leq k$ and $\mathbf{JSL_f}$-morphisms p, q and τ_a ($a \in \Sigma$) making the left-hand diagram below commute.*
2. $\mathsf{nsyn}(L)$ *is the least number k such that there exists a finite semilattice S with $|J(S)| \leq k$ and $\mathbf{JSL_f}$-morphisms p, q and τ_a ($a \in \Sigma$) making the right-hand diagram below commute.*

$$
\begin{tikzcd}
 & \mathsf{BLD}(L) \arrow[r, "\delta'_a"] & \mathsf{BLD}(L) \arrow[dr, "f'"] & \\
\mathfrak{2} \arrow[ur, "i'"] \arrow[dr, "i"'] & S \arrow[u, dashed, "q"] \arrow[r, dashed, "\tau_a"] & S \arrow[u, dashed, "q"'] & \mathfrak{2} \\
 & \mathsf{SLD}(L) \arrow[u, dashed, "p"] \arrow[r, dashed, "\delta_a"'] & \mathsf{SLD}(L) \arrow[u, dashed, "p"'] \arrow[ur, "f"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \mathsf{BLRD}(L) \arrow[r, "\delta''_a"] & \mathsf{BLRD}(L) \arrow[dr, "f''"] & \\
\mathfrak{2} \arrow[ur, "i''"] \arrow[dr, "i"'] & S \arrow[u, dashed, "q"] \arrow[r, dashed, "\tau_a"] & S \arrow[u, dashed, "q"'] & \mathfrak{2} \\
 & \mathsf{SLD}(L) \arrow[u, dashed, "p"] \arrow[r, dashed, "\delta_a"'] & \mathsf{SLD}(L) \arrow[u, dashed, "p"'] \arrow[ur, "f"'] &
\end{tikzcd}
$$

Proof. We only prove part (1), the proof of (2) being completely analogous.

Suppose there exists a finite semilattice S with $|J(S)| = k$ and $\mathbf{JSL_f}$-morphisms p, q and $(\tau_a)_{a\in\Sigma}$ making the left diagram commute. Then $A = (S, \tau, p \circ i, f' \circ q)$ is a $\mathbf{JSL}$-dfa and $p\colon \mathsf{SLD}(L) \to A$ and $q\colon A \to \mathsf{BLD}(L)$ are $\mathbf{JSL}$-dfa morphisms. Since $\mathbf{JSL}$-dfa morphisms preserve the accepted language, and every state $K \in \mathsf{BLD}(L)$ accepts the language K, it follows that A accepts L and every state of A accepts a language from $\mathsf{BLD}(L)$. Thus the nfa $J(A)$ of join-irreducibles corresponding to A is an atomic nfa for L with k states.

Conversely, assume $N = (Q, \delta, I, F)$ is a k-state atomic nfa accepting L. Form the $\cup$-semilattice $S = \mathsf{langs}(N)$ of all languages $L(N, X)$ accepted by subsets $X \subseteq Q$. Note that $\mathsf{SLD}(L) \subseteq S \subseteq \mathsf{BLD}(L)$: the first inclusion holds because $u^{-1}L = L(N, \delta_w[I]) \in S$ for every $u \in \Sigma^*$, and the second one because N is atomic. We define the semilattice endomorphisms

$$\tau_a\colon S \to S \qquad \text{by} \qquad \tau_a(K) = a^{-1}K \quad \text{for } K \in S,$$

Letting $p\colon \mathsf{SLD}(L) \rightarrowtail S$ and $q\colon S \rightarrowtail \mathsf{BLD}(L)$ denote the inclusions, the left diagram commutes. Moreover $|J(S)| \leq k$ since S is join-generated by the elements $L(N, q)$ for $q \in Q$. ◀

3 Representing Finite Semilattices as Finite Relations

We have seen that atomic and subatomic nfas amount to certain dfas with semilattice structure. To obtain our NP-completeness results concerning the computation of small (sub)atomic acceptors we will study succinct representations of the corresponding $\mathbf{JSL_f}$-diagrams from Theorem 2.1. For this purpose, we start with the following key observation:

> Any finite semilattice S is completely determined by its *poset of irreducibles* [23], i.e. the relation $\nleq_S \subseteq J(S) \times M(S)$ between join-irreducibles and meet-irreducibles.

We now prove that this extends to an *equivalence* between the category $\mathbf{JSL_f}$ of finite semilattices and another category called $\mathbf{Dep}$. Its objects are the relations between finite sets and its morphisms represent semilattice morphisms as relations. The equivalence is inspired by Moshier's *categories of contexts* [16, 24] and will serve as the conceptual basis of our work.

▶ **Definition 3.1** (The category of dependency relations)**.** The objects of the category $\mathbf{Dep}$ are the relations $\mathcal{R} \subseteq \mathcal{R}_\mathsf{s} \times \mathcal{R}_\mathsf{t}$ between finite sets. Far less obviously,

> a morphism $\mathcal{P}\colon \mathcal{R} \to \mathcal{S}$ is a relation $\mathcal{P} \subseteq \mathcal{R}_\mathsf{s} \times \mathcal{S}_\mathsf{t}$ that factorizes through $\mathcal{R}$ and $\mathcal{S}$, i.e. the left $\mathbf{Rel}$-diagram below commutes for some $\mathcal{P}_l \subseteq \mathcal{R}_\mathsf{s} \times \mathcal{S}_\mathsf{s}$ and $\mathcal{P}_u \subseteq \mathcal{S}_\mathsf{t} \times \mathcal{R}_\mathsf{t}$.

The identity morphism for $\mathcal{R}$ is $\mathsf{id}_\mathcal{R} = \mathcal{R}$, see the central diagram below. The composite $\mathcal{P} ; \mathcal{Q}\colon \mathcal{R} \to \mathcal{T}$ of $\mathcal{P}\colon \mathcal{R} \to \mathcal{S}$ and $\mathcal{Q}\colon \mathcal{S} \to \mathcal{T}$ is any of the five equivalent relational compositions starting from the bottom left corner and ending at the top right corner of the rightmost diagram below; that is, $\mathcal{P} \fatsemi \mathcal{Q} := \mathcal{P}_l ; \mathcal{Q}_l ; \mathcal{T} = \mathcal{P}_l ; \mathcal{Q} = \mathcal{P}_l ; \mathcal{S} ; \mathcal{Q}_u^{\smile} = \mathcal{P} ; \mathcal{Q}_u^{\smile} = \mathcal{R} ; \mathcal{P}_u^{\smile} ; \mathcal{Q}_u^{\smile}$. (Note that we use the symbol $\fatsemi$ for composition in **Dep** and ; for composition in **Rel**, and recall that $(-)^{\smile}$ denotes the converse relation.)

$$
\begin{tikzcd}
\mathcal{R}_{\mathsf{t}} \arrow[r, dashed, "\mathcal{P}_u^{\smile}"] & \mathcal{S}_{\mathsf{t}} \\
\mathcal{R}_{\mathsf{s}} \arrow[u, "\mathcal{R}"] \arrow[ur, "\mathcal{P}"] \arrow[r, dashed, "\mathcal{P}_l"'] & \mathcal{S}_{\mathsf{s}} \arrow[u, "\mathcal{S}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{R}_{\mathsf{t}} \arrow[r, dashed, "\mathsf{id}^{\smile}"] & \mathcal{R}_{\mathsf{t}} \\
\mathcal{R}_{\mathsf{s}} \arrow[u, "\mathcal{R}"] \arrow[ur, "\mathcal{R}"] \arrow[r, dashed, "\mathsf{id}"'] & \mathcal{R}_{\mathsf{s}} \arrow[u, "\mathcal{R}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{R}_{\mathsf{t}} \arrow[r, dashed, "\mathcal{P}_u^{\smile}"] & \mathcal{S}_{\mathsf{t}} \arrow[r, dashed, "\mathcal{Q}_u^{\smile}"] & \mathcal{T}_{\mathsf{t}} \\
\mathcal{R}_{\mathsf{s}} \arrow[u, "\mathcal{R}"] \arrow[ur, "\mathcal{P}"] \arrow[r, dashed, "\mathcal{P}_l"'] & \mathcal{S}_{\mathsf{s}} \arrow[u, "\mathcal{S}"'] \arrow[ur, "\mathcal{Q}"] \arrow[r, dashed, "\mathcal{Q}_l"'] & \mathcal{T}_{\mathsf{s}} \arrow[u, "\mathcal{T}"']
\end{tikzcd}
$$

One readily verifies that **Dep** is a well-defined category; in particular, the composition is independent of the choice of the lower and upper witnesses $(-)_l$ and $(-)_u$.

▶ **Remark 3.2.**

1. Using the converse upper witness may seem strange. Although technically unnecessary, it fits the self-duality of **Dep** taking the converse on objects and morphisms. Moreover $f ; \nleq_T = \nleq_S ; f_*^{\smile}$ for any $\mathbf{JSL_f}$-morphism $f \colon S \to T$ via the adjoint relationship; that is, f induces a **Dep**-morphism from $\nleq_S$ to $\nleq_T$ with lower witness f and upper witness f_*.
2. The witnesses of a **Dep**-morphism $\mathcal{P} \colon \mathcal{R} \to \mathcal{S}$ are closed under unions. The maximal lower witness $\mathcal{P}_- \subseteq \mathcal{R}_{\mathsf{s}} \times \mathcal{S}_{\mathsf{s}}$ is given by

$$\mathcal{P}_-(x, y) \; :\Longleftrightarrow \; \mathcal{S}[y] \subseteq \mathcal{P}[x] \qquad \text{for} \qquad x \in \mathcal{R}_{\mathsf{s}},\, y \in \mathcal{S}_{\mathsf{s}},$$

and the maximal upper witness $\mathcal{P}_+ \subseteq \mathcal{S}_{\mathsf{t}} \times \mathcal{R}_{\mathsf{t}}$ by

$$\mathcal{P}_+(y, x) \; :\Longleftrightarrow \; \breve{\mathcal{R}}[x] \subseteq \breve{\mathcal{P}}[y] \qquad \text{for} \qquad x \in \mathcal{R}_{\mathsf{t}},\, y \in \mathcal{S}_{\mathsf{t}}.$$

▶ **Theorem 3.3** (Fundamental equivalence)**.** *The categories* $\mathbf{JSL_f}$ *and* **Dep** *are equivalent.*

1. *The equivalence functor* $\mathsf{Pirr} \colon \mathbf{JSL_f} \to \mathbf{Dep}$ *maps a finite semilattice* S *to the* **Dep***-object*

$$\mathsf{Pirr}(S) \; := \nleq_S \subseteq \; J(S) \times M(S),$$

and a $\mathbf{JSL_f}$*-morphism* $f \colon S \to T$ *to the* **Dep***-morphism*

$$\mathsf{Pirr}(f) \colon \mathsf{Pirr}(S) \to \mathsf{Pirr}(T), \qquad \mathsf{Pirr}(f)(j, m) :\Leftrightarrow f(j) \nleq_T m \quad \textit{for } j \in J(S),\, m \in M(T).$$

2. *The inverse* $\mathsf{Open} \colon \mathbf{Dep} \to \mathbf{JSL_f}$ *maps a* **Dep***-object* $\mathcal{R}$ *to its* semilattice of open sets

$$\mathsf{Open}(\mathcal{R}) \; := \; (\{\mathcal{R}[X] : X \subseteq \mathcal{R}_{\mathsf{s}}\}, \subseteq),$$

and a **Dep***-morphism* $\mathcal{P} \colon \mathcal{R} \to \mathcal{S}$ *to the* $\mathbf{JSL_f}$*-morphism*

$$\mathsf{Open}(\mathcal{P}) \colon \mathsf{Open}(\mathcal{R}) \to \mathsf{Open}(\mathcal{S}), \qquad \mathsf{Open}(\mathcal{P})(O) := \mathcal{P}_+^{\smile}[O] \quad \textit{for } O \in \mathsf{Open}(\mathcal{R}),$$

where $\mathcal{P}_+ \subseteq \mathcal{S}_{\mathsf{t}} \times \mathcal{R}_{\mathsf{t}}$ *is the maximal upper witness of* $\mathcal{P}$*.*

▶ **Remark 3.4.** In the definition of $\mathsf{Pirr}(S)$ one may replace $J(S)$ and $M(S)$ by any two sets $J, M \subseteq S$ of join- and meet-generators modulo **Dep**-isomorphism. Indeed, since the equivalence functor Open reflects isomorphisms, this follows immediately from the $\mathbf{JSL_f}$-isomorphism $\mathsf{Open}(\nleq_S \cap\, J \times M) \cong \mathsf{Open}(\nleq_S \cap\, J(S) \times M(S))$ given by $O \mapsto O \cap M(S)$.

▶ **Remark 3.5.** Bijectively relabeling the domain and codomain of a relation defines a **Dep**-isomorphism, the witnesses being the relabelings.

We now show that for every regular language L, the semilattices $\mathsf{SLD}(L)$, $\mathsf{BLD}(L)$ and $\mathsf{BLRD}(L)$ equipped with their canonical **JSL**-dfa structure (see Section 2) translate under the equivalence functor Pirr into familiar concepts from automata theory. The translations are summarized in Table 1 and explained in Examples 3.6–3.8 below.

Table 1 Canonical **JSL**-dfas and their corresponding **Dep**-structures.

$\mathbf{JSL_f}$	$\mathbf{Dep}$
$\mathbb{2} \xrightarrow{i} \mathsf{SLD}(L) \xrightarrow{\delta_a} \mathsf{SLD}(L) \xrightarrow{f} \mathbb{2}$	$\mathsf{id}_1 \xrightarrow{\mathcal{I}} \mathcal{DR}_L \xrightarrow{\mathcal{DR}_{L,a}} \mathcal{DR}_L \xrightarrow{\mathcal{F}} \mathsf{id}_1$
$\mathbb{2} \xrightarrow{i'} \mathsf{BLD}(L) \xrightarrow{\delta'_a} \mathsf{BLD}(L) \xrightarrow{f'} \mathbb{2}$	$\mathsf{id}_1 \xrightarrow{\mathcal{I}'} \mathsf{id}_{\Sigma^*/\sim_L} \xrightarrow{\mathcal{D}'_a} \mathsf{id}_{\Sigma^*/\sim_L} \xrightarrow{\mathcal{F}'} \mathsf{id}_1$
$\mathbb{2} \xrightarrow{i''} \mathsf{BLRD}(L) \xrightarrow{\delta''_a} \mathsf{BLRD}(L) \xrightarrow{f''} \mathbb{2}$	$\mathsf{id}_1 \xrightarrow{\mathcal{I}''} \mathsf{id}_{\mathsf{syn}(L)} \xrightarrow{\mathcal{D}''_a} \mathsf{id}_{\mathsf{syn}(L)} \xrightarrow{\mathcal{F}''} \mathsf{id}_1$

▶ **Example 3.6** (State-minimal **JSL**-dfa vs. dependency relation $\mathcal{DR}_L$)**.** Let us start with the observation that $\mathsf{SLD}(L)$ is join-generated by $\mathsf{LD}(L)$ and meet-generated by $\mathrm{dr}_L[\mathsf{LD}(L^{\mathsf{r}})]$. The latter follows via the fundamental isomorphism (2.1). Then

$$\mathsf{Pirr}(\mathsf{SLD}(L))(u^{-1}L, \mathrm{dr}_L(v^{-1}L^{\mathsf{r}})) \overset{\text{def.}}{\iff} u^{-1}L \nsubseteq \mathrm{dr}_L(v^{-1}L^{\mathsf{r}}) \overset{(2.3)}{\iff} \mathcal{DR}_L(u^{-1}L, v^{-1}L^{\mathsf{r}})$$

for every $u^{-1}L \in J(\mathsf{SLD}(L))$ and $v^{-1}L^{\mathsf{r}} \in J(\mathsf{SLD}(L^{\mathsf{r}}))$. Thus,

$\mathsf{Pirr}(\mathsf{SLD}(L))$ is a bijective relabeling of $\mathcal{DR}_L$ restricted to $J(\mathsf{SLD}(L)) \times J(\mathsf{SLD}(L^{\mathsf{r}}))$.

By Remark 3.4 we know $\mathsf{Pirr}(\mathsf{SLD}(L))$ is isomorphic to the domain-codomain extension $\nsubseteq\; \subseteq \mathsf{LD}(L) \times \mathrm{dr}_L[\mathsf{LD}(L^r)]$ and thus also to the dependency relation $\mathcal{DR}_L$ by Remark 3.5. Then the **JSL**-dfa structure of the semilattice $\mathsf{SLD}(L)$ translates into the category of dependency relations as shown in Table 1, where id_1 is the identity relation on $1 = \{*\}$ and

$$\begin{array}{lll} \mathcal{I} \subseteq 1 \times \mathsf{LD}(L^{\mathsf{r}}), & \mathcal{DR}_{L,a} \subseteq \mathsf{LD}(L) \times \mathsf{LD}(L^{\mathsf{r}}), & \mathcal{F} \subseteq \mathsf{LD}(L) \times 1, \\ \mathcal{I}(*, v^{-1}L^{\mathsf{r}}) \Leftrightarrow v \in L^{\mathsf{r}}, & \mathcal{DR}_{L,a}(u^{-1}L, v^{-1}L^{\mathsf{r}}) \Leftrightarrow uav^{\mathsf{r}} \in L, & \mathcal{F}(u^{-1}L, *) \Leftrightarrow u \in L. \end{array}$$

▶ **Example 3.7** ($\mathsf{BLD}(L)$ vs. the Nerode left congruence $\sim_L$)**.** In Section 2 we observed that the atoms of the boolean algebra $\mathsf{BLD}(L)$ are the congruence classes of the Nerode left congruence. Then the co-atoms are their relative complements, and

$$\mathsf{Pirr}(\mathsf{BLD}(L))([u]_{\sim_L}, \overline{[v]_{\sim_L}}) \overset{\text{def.}}{\iff} [u]_{\sim_L} \nsubseteq \overline{[v]_{\sim_L}} \iff [u]_{\sim_L} = [v]_{\sim_L}.$$

By Remark 3.5, we see that $\mathsf{BLD}(L)$ corresponds to the **Dep**-object $\mathsf{id}_{\Sigma^*/\sim_L}$, and its **JSL**-dfa structure translates into the category of dependency relations as indicated in Table 1, where

$$\begin{array}{lll} \mathcal{I}' \subseteq 1 \times \Sigma^*/\sim_L, & \mathcal{D}'_a \subseteq \Sigma^*/\sim_L \times \Sigma^*/\sim_L, & \mathcal{F}' \subseteq \Sigma^*/\sim_L \times 1, \\ \mathcal{I}'(*, [u]_{\sim_L}) \Leftrightarrow u \in L, & \mathcal{D}'_a([u]_{\sim_L}, [v]_{\sim_L}) \Leftrightarrow [v]_{\sim_L} \subseteq a^{-1}[u]_{\sim_L}, & \mathcal{F}'([u]_{\sim_L}, *) \Leftrightarrow u \sim_L \varepsilon. \end{array}$$

We note that the above relations induce an nfa

$(\Sigma^*/\sim_L, (\mathcal{D}'_a)_{a\in\Sigma}, \mathcal{I}'[*], \breve{\mathcal{F}}'[*])$ known as the *átomaton* for the language L [4].

▶ **Example 3.8** ($\mathsf{BLRD}(L)$ vs. the syntactic monoid $\mathsf{syn}(L)$)**.** Analogously, the boolean algebra $\mathsf{BLRD}(L)$ corresponds to the **Dep**-object $\mathsf{id}_{\mathsf{syn}(L)}$. Its semilattice dfa structure translates into the category of dependency relations as shown in Table 1, where

$$\begin{array}{lll} \mathcal{I}'' \subseteq 1 \times \mathsf{syn}(L), & \mathcal{D}''_a \subseteq \mathsf{syn}(L) \times \mathsf{syn}(L), & \mathcal{F}'' \subseteq \mathsf{syn}(L) \times 1, \\ \mathcal{I}''(*, [u]_{\equiv_L}) \Leftrightarrow u \in L, & \mathcal{D}''_a([u]_{\equiv_L}, [v]_{\equiv_L}) \Leftrightarrow [v]_{\equiv_L} \subseteq a^{-1}[u]_{\equiv_L}, & \mathcal{F}''([u]_{\equiv_L}, *) \Leftrightarrow u \equiv_L \varepsilon. \end{array}$$

We conclude this section with two lemmas establishing important properties of the equivalence. The first concerns the bipartite dimension of relations (see Section 2):

▶ **Lemma 3.9.** *Let $\mathcal{R}$ be a relation between finite sets.*

1. $\mathsf{dim}(\mathcal{R})$ *is the least* $|J(S)|$ *of any injective* $\mathbf{JSL_f}$*-morphism* $m\colon \mathsf{Open}(\mathcal{R}) \rightarrowtail S$.
2. $\mathsf{dim}(\mathcal{R})$ *is invariant under isomorphism, i.e.* $\mathcal{R} \cong \mathcal{S}$ *in* **Dep** *implies* $\mathsf{dim}(\mathcal{R}) = \mathsf{dim}(\mathcal{S})$.

The second explicitly describes the join- and meet-irreducibles of the semilattice $\mathsf{Open}(\mathcal{R})$.

▶ **Notation 3.10.** For $\mathcal{R} \subseteq \mathcal{R}_\mathsf{s} \times \mathcal{R}_\mathsf{t}$ we define the following operator on the power set of $\mathcal{R}_\mathsf{t}$:

$$\mathsf{in}_\mathcal{R}\colon \mathcal{P}(\mathcal{R}_\mathsf{t}) \to \mathcal{P}(\mathcal{R}_\mathsf{t}), \qquad Y \;\mapsto\; \bigcup\{\mathcal{R}[X] \;:\; X \subseteq \mathcal{R}_\mathsf{s} \text{ and } \mathcal{R}[X] \subseteq Y\}.$$

Thus, $\mathsf{in}_\mathcal{R}(Y)$ is the largest open set of $\mathcal{R}$ contained in $Y \subseteq \mathcal{R}_\mathsf{t}$.

▶ **Lemma 3.11.** *Let $\mathcal{R} \subseteq \mathcal{R}_\mathsf{s} \times \mathcal{R}_\mathsf{t}$ be a relation between finite sets.*

1. $J(\mathsf{Open}(\mathcal{R}))$ *consists of all sets* $\mathcal{R}[x]$ *($x \in \mathcal{R}_\mathsf{s}$) that cannot be expressed as a union of smaller such sets, i.e.* $\mathcal{R}[x] = \bigcup_{i\in I} \mathcal{R}[x_i]$ *implies* $\mathcal{R}[x] = \mathcal{R}[x_i]$ *for some* $i \in I$.
2. $M(\mathsf{Open}(\mathcal{R}))$ *consists of all sets* $\mathsf{in}_\mathcal{R}(\mathcal{R}_\mathsf{t} \setminus \{y\})$ *such that* $\breve{\mathcal{R}}[y]$ *lies in* $J(\mathsf{Open}(\breve{\mathcal{R}}))$.

4 Nuclear Languages and Lattice Languages

As a further technical tool, we now introduce two classes of regular languages. They are well-behaved w.r.t. their small nfas and will emerge at the heart of our NP-completeness proofs in Section 5. Their definition rests on the notion of a *nuclear morphism* in $\mathbf{JSL_f}$, originating from the theory of symmetric monoidal closed categories [12, 28]. Recall that a finite semilattice is a *distributive lattice* if $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ for all elements x, y, z.

▶ **Definition 4.1** (Nuclear language)**.** A $\mathbf{JSL_f}$-morphism $f\colon S \to T$ is *nuclear* if it factorizes through a finite distributive lattice, i.e. $f = (S \xrightarrow{g} D \xrightarrow{h} T)$ for some finite distributive lattice D and $\mathbf{JSL_f}$-morphisms g, h. A regular language $L \subseteq \Sigma^*$ is *nuclear* if the transition morphisms $\delta_a = a^{-1}(-)\colon \mathsf{SLD}(L) \to \mathsf{SLD}(L)$ ($a \in \Sigma$) of its minimal **JSL**-dfa are nuclear.

▶ **Example 4.2** (BiRFSA languages)**.** A regular language L is *biRFSA* [19] if $(N_L)^\mathsf{r} \cong N_{L^\mathsf{r}}$, that is, the canonical residual finite state automata for L and L^r (see Section 2) are reverse-isomorphic. In [26, Example 5.7] we proved that the biRFSA languages are precisely those whose semilattice $\mathsf{SLD}(L)$ is distributive. Thus biRFSA languages are nuclear.

There is a natural subclass of nuclear languages which need not be biRFSA:

▶ **Definition 4.3** (Lattice language)**.** For any $S \in \mathbf{JSL_f}$ we define the language $L(S) \subseteq \Sigma^*$,

$$\Sigma := \{\langle j| : j \in J(S)\} \cup \{|m\rangle : m \in M(S)\} \qquad \text{and} \qquad L(S) := \bigcap_{j \le_S m} \overline{\Sigma^* \langle j|\,|m\rangle \Sigma^*}.$$

Then Σ is the disjoint union of $J(S)$ and $M(S)$ (with the notation $\langle j|$ and $|m\rangle$ used to distinguish between elements of the two summands), and $L(S)$ consists of all words over Σ not containing any factor $\langle j|\,|m\rangle$ with $j \le_S m$.

▶ **Lemma 4.4.** *For any $S \in \mathbf{JSL_f}$, the language $L(S)$ is nuclear and $S \cong \mathsf{SLD}(L(S))$.*

Crucially, for nuclear and lattice languages some of the relations (2.6) hold with equality:

▶ **Proposition 4.5.**

1. *If L is a nuclear language then* $\mathsf{ns}(L) = \mathsf{dim}(\mathcal{DR}_L)$.
2. *If $L = L(S)$ is a lattice language then* $\mathsf{natm}(L) = \mathsf{nsyn}(L) = \mathsf{ns}(L) = \mathsf{dim}(\mathcal{DR}_L)$.

These equalities are the key fact making our reductions in the next section work.

5 Complexity of Computing Small (Sub)Atomic Acceptors

We are ready to present our main complexity results on small (sub)atomic nfas. First we consider the slightly simpler atomic case, phrased as the following decision problem:

$\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$
Input: Two dfas A and B such that $L(A) = L(B)^{\mathsf{r}}$ and a natural number k.
Task: Decide whether there exists a k-state atomic nfa equivalent to A, i.e. $\mathsf{natm}(L(A)) \leq k$.

▶ **Remark 5.1.** Taking mutually reverse dfas (A, B) as input permits an efficient computation of the dependency relation $\mathcal{DR}_L \subseteq \mathsf{LD}(L) \times \mathsf{LD}(L^{\mathsf{r}})$ of $L = L(A)$. One may assume A and B are minimal dfas, so that their state sets Q_A and Q_B are in bijective correspondence with $\mathsf{LD}(L)$ and $\mathsf{LD}(L^{\mathsf{r}})$. For $p \in Q_A$ choose some $w_A(p) \in \Sigma^*$ sending the initial state to p; analogously choose $w_B(q) \in \Sigma^*$ for $q \in Q_B$. Then $\mathcal{DR}_L$ is a bijective relabeling of

$$\widetilde{\mathcal{DR}_L} \subseteq Q_A \times Q_B \qquad \text{where} \qquad \widetilde{\mathcal{DR}_L}(p, q) \;:\iff\; A \text{ accepts } w_A(p)w_B(q)^{\mathsf{r}},$$

so it is computable in polynomial time from A and B. A completely analogous argument applies to the relations $\mathcal{I}$, $\mathcal{DR}_{L,a}$ and $\mathcal{F}$ from Example 3.6.

▶ **Theorem 5.2.** *The problem* $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$ *is* NP*-complete.*

We establish the upper and lower bound separately in the next two propositions. Both their proofs are based on the fundamental equivalence between $\mathbf{JSL_f}$ and $\mathbf{Dep}$.

▶ **Proposition 5.3.** *The problem* $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$ *is in* NP.

Proof.

1. One can check in polynomial time whether a given pair (A, B) of dfas forms a valid input, i.e. satisfies $L(A) = L(B)^{\mathsf{r}}$. In fact, this condition is equivalent to $\overline{L(A)} \cap L(B)^{\mathsf{r}} = \overline{L(B)} \cap L(A)^{\mathsf{r}} = \emptyset$. Using the standard methods for complementing dfas and reversing and intersecting nfas, one can construct nfas for $\overline{L(A)} \cap L(B)^{\mathsf{r}}$ and $\overline{L(B)} \cap L(A)^{\mathsf{r}}$ of size polynomial in $|A|$ and $|B|$, the number of states of A and B, and check for emptyness by verifying that no final state is reachable from the initial states.
2. Let A and B be dfas accepting the languages L and L^{r}, respectively, and let k be a natural number. We claim the following three statements to be equivalent:
   a. There exists an atomic nfa accepting L with at most k states.
   b. There exists a finite semilattice S with $|J(S)| \leq k$ and $\mathbf{JSL_f}$-morphisms p, q and τ_a $(a \in \Sigma)$ making the left diagram below commute.
   c. There exists a $\mathbf{Dep}$-object $\mathcal{S} \subseteq \mathcal{S}_{\mathsf{s}} \times \mathcal{S}_{\mathsf{t}}$ with $|\mathcal{S}_{\mathsf{s}}| \leq k$ and $|\mathcal{S}_{\mathsf{t}}| \leq |B|$ and $\mathbf{Dep}$-morphisms $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{T}_a$ $(a \in \Sigma)$ making the right diagram below commute (cf. Example 3.6/3.7).

$$\begin{tikzcd} & \mathsf{BLD}(L) \arrow[r, "\delta'_a"] \arrow[dr, phantom, ""] & \mathsf{BLD}(L) \arrow[dr, "f'"] & \\ \mathbb{2} \arrow[ur, "i'"] \arrow[dr, "i"'] & S \arrow[u, dashed, "q"] \arrow[r, dashed, "\tau_a"] & S \arrow[u, dashed, "q"'] & \mathbb{2} \\ & \mathsf{SLD}(L) \arrow[u, dashed, "p"] \arrow[r, dashed, "\delta_a"'] & \mathsf{SLD}(L) \arrow[u, dashed, "p"'] \arrow[ur, "f"'] & \end{tikzcd} \qquad \begin{tikzcd} & \mathsf{id}_{\Sigma^*/\sim_L} \arrow[r, "\mathcal{D}'_a"] & \mathsf{id}_{\Sigma^*/\sim_L} \arrow[dr, "\mathcal{F}'"] & \\ \mathsf{id}_1 \arrow[ur, "\mathcal{I}'"] \arrow[dr, "\mathcal{I}"'] & \mathcal{S} \arrow[u, dashed, "\mathcal{Q}"] \arrow[r, dashed, "\mathcal{T}_a"] & \mathcal{S} \arrow[u, dashed, "\mathcal{Q}"'] & \mathsf{id}_1 \\ & \mathcal{DR}_L \arrow[u, dashed, "\mathcal{P}"] \arrow[r, "\mathcal{DR}_{L,a}"'] & \mathcal{DR}_L \arrow[u, dashed, "\mathcal{P}"'] \arrow[ur, "\mathcal{F}"'] & \end{tikzcd} \tag{5.1}$$

In fact, (a)⇔(b) was shown in Theorem 2.1(1), and (b)⇔(c) follows from the equivalence between $\mathbf{JSL_f}$ and $\mathbf{Dep}$. To see this, note that in the left diagram we may assume q to be injective; otherwise, factorize q as $q = q' \circ e'$ with e surjective and q' injective and work with q' instead of q. By the self-duality of $\mathbf{JSL_f}$, dualizing q yields a surjective morphism from $\mathsf{BLD}(L) \cong \mathsf{BLD}(L)^{\mathsf{op}}$ to S^{op}. Thus,

$$|M(S)| = |J(S^{\mathsf{op}})| \leq |J(\mathsf{BLD}(L))| = |\Sigma^*/{\sim_L}| = |\mathsf{LD}(L^{\mathsf{r}})| \leq |B|.$$

In the two last steps, we use that the congruence classes of $\sim_L$ correspond bijectively to left derivatives of L^{r} by (2.4), and that $\mathsf{LD}(L^{\mathsf{r}})$ is the set of states of the minimal dfa for L^{r}.

By Example 3.6 and 3.7 the upper and lower path of the left diagram in $\mathbf{JSL_f}$ correspond under the equivalence functor Pirr to the upper and lower path of the right diagram in $\mathbf{Dep}$. Therefore, Theorem 3.3 shows the two diagrams to be equivalent.

3. From (a)⇔(c) we deduce that the relations $\mathcal{S}$, $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{T}_a$ $(a \in \Sigma)$ constitute a short certificate for the existence of an atomic nfa for L with at most k states. Commutativity of the right diagram can be checked in polynomial time because all the relations appearing in the upper and lower path can be efficiently computed from the given dfas A and B. Indeed, for the lower path we have already noted this in Remark 5.1, and the upper path emerges from the minimal dfa for L^{r}, using that $\Sigma^*/{\sim_L} \cong \mathsf{LD}(L^{\mathsf{r}})$. ◀

▶ **Remark 5.4.** An alternative proof that $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$ is in NP uses the following characterization of atomic nfas. Given an nfa N, let $\mathsf{rsc}(N^{\mathsf{r}})$ denote the dfa obtained by determinizing the reverse nfa N^{r} via the subset construction and restricting to its reachable part. Then N is atomic iff $\mathsf{rsc}(N^{\mathsf{r}})$ is a minimal dfa [4, Corollary 2]. Thus, given a pair (A, B) of mutually reversed dfas, to decide whether $\mathsf{natm}(L(A)) \leq k$ one may guess a k-state nfa N and verify that $\mathsf{rsc}(N^{\mathsf{r}})$ is a minimal dfa equivalent to B. One advantage of our above categorical argument is that it yields simple certificates in the form of $\mathbf{Dep}$-morphisms subject to certain commutative diagrams, which amount to solutions of equations in $\mathbf{Rel}$. The latter may be directly computed using a SAT solver, leading to a practical approach to finding small atomic acceptors (cf. [9]). To this effect, let us note that the proof of Proposition 5.3 actually shows how to *construct* small atomic nfas rather than just deciding their existence: every certificate $\mathcal{S}, \mathcal{P}, \mathcal{Q}, \mathcal{T}_a$ $(a \in \Sigma)$ yields an atomic nfa with states $\mathcal{S}_{\mathsf{s}}$, transitions given by $(\mathcal{T}_a)_- \subseteq \mathcal{S}_{\mathsf{s}} \times \mathcal{S}_{\mathsf{s}}$ for $a \in \Sigma$, initial states $(\mathcal{I} \mathbin{;} \mathcal{P})_-[*] \subseteq \mathcal{S}_{\mathsf{s}}$ and final states $(Q \mathbin{;} \mathcal{F}')^{\smile}_-[*] \subseteq \mathcal{S}_{\mathsf{s}}$. (Recall that $\mathbin{;}$ denotes composition in $\mathbf{Dep}$ and $(-)_-$ denotes the maximum lower witness of a $\mathbf{Dep}$-morphism, see Remark 3.2.) In fact, this is precisely the nfa of join-irreducibles of the $\mathbf{JSL}$-dfa $(S, \tau, p \circ i, f' \circ q)$ induced by the left diagram in (5.1). Analogous reasoning also applies to the computation of small subatomic nfas treated in Theorem 5.7 below.

▶ **Proposition 5.5.** *The problem* $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{atm}}$ *is* NP*-hard.*

Proof. We devise a polynomial-time reduction from the NP-complete problem **BICLIQUE COVER** [8]: given a pair $(\mathcal{R}, k)$ of a relation $\mathcal{R} \subseteq \mathcal{R}_{\mathsf{s}} \times \mathcal{R}_{\mathsf{t}}$ between finite sets and a natural number k, decide whether $\mathcal{R}$ has a biclique cover of size at most k, i.e. $\mathsf{dim}(\mathcal{R}) \leq k$.

For any $(\mathcal{R}, k)$, let $S = \mathsf{Open}(\mathcal{R})$ be the finite semilattice of open sets corresponding to the $\mathbf{Dep}$-object $\mathcal{R}$, cf. Theorem 3.3, and let $L = L(S)$ be its lattice language. We claim that the desired reduction is given by

$$(\mathcal{R}, k) \quad \longmapsto \quad (\mathsf{dfa}(L), \mathsf{dfa}(L^{\mathsf{r}}), k),$$

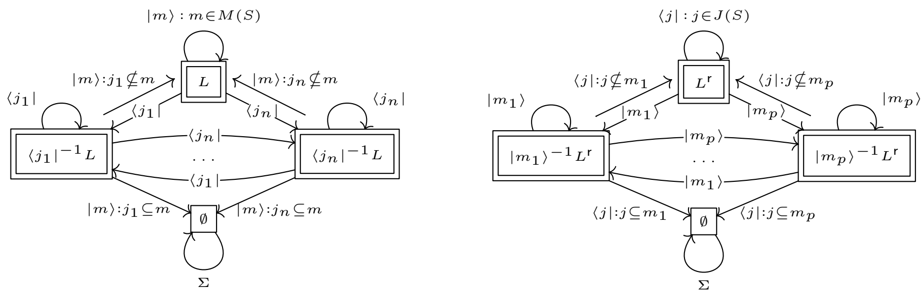
where $\mathsf{dfa}(L)$ and $\mathsf{dfa}(L^{\mathsf{r}})$ are the minimal dfas for L and L^{r}. Thus, we need to prove that (a) $\mathsf{dim}(\mathcal{R}) = \mathsf{natm}(L)$, and (b) the two dfas can be computed in polynomial time from $\mathcal{R}$.

Ad (a). We have the following sequence of **Dep**-isomorphisms:

$$\mathcal{R} \underset{\text{Thm 3.3}}{\overset{\cong}{\longleftrightarrow}} \mathsf{Pirr}(\mathsf{Open}(\mathcal{R})) = \mathsf{Pirr}(S) \underset{\text{Lem 4.4}}{\overset{\cong}{\longleftrightarrow}} \mathsf{Pirr}(\mathsf{SLD}(L(S))) = \mathsf{Pirr}(\mathsf{SLD}(L)) \underset{\text{Ex 3.6}}{\overset{\cong}{\longleftrightarrow}} \mathcal{DR}_L.$$

Lemma 3.9(2) and Proposition 4.5 then imply $\mathsf{dim}(\mathcal{R}) = \mathsf{dim}(\mathcal{DR}_L) = \mathsf{natm}(L)$.

Ad (b). Let $J(\mathsf{Open}(\mathcal{R})) = \{j_1, \ldots, j_n\}$ and $M(\mathsf{Open}(\mathcal{R})) = \{m_1, \ldots, m_p\}$. Then $\mathsf{dfa}(L)$ and $\mathsf{dfa}(L^{\mathsf{r}})$ are the automata depicted below, where L and L^{r} are their respective initial states.



Both automata can be computed in polynomial time from $\mathcal{R}$ using Lemma 3.11. ◀

Next, we turn to the computation of small subatomic nfas. While in the atomic case the input language was specified by a pair of dfas, we now assume an algebraic representation:

▶ **Definition 5.6.** A *monoid recognizer* is a triple (M, h, F) of a finite monoid M, a map $h\colon \Sigma \to M$ and a subset $F \subseteq M$. The language *recognized* by (M, h, F) is given by $L(M, h, f) = \overline{h}^{-1}[F]$, where $\overline{h}\colon \Sigma^* \to M$ is the unique extension of h to a monoid morphism.

It is well-known [27] that a language L is regular iff it has a monoid recognizer. In this case, a *minimal* monoid recognizer for L is given by $(\mathsf{syn}(L), \mu_L, F_L)$ where $\mu_L\colon \Sigma \to \mathsf{syn}(L)$ is the domain restriction of the syntactic morphism and $F_L = \{[w]_{\equiv_L} : w \in L\}$. It satisfies $|\mathsf{syn}(L)| \leq |M|$ for every recognizer (M, h, F) of L. Consider the following decision problem:

MON $\to$ **NFA$_{\mathbf{syn}}$**
Input: A monoid recognizer (M, h, F) and a natural number k.
Task: Decide whether there exists a k-state subatomic nfa accepting $L(M, h, F)$.

Here we assume that the monoid M is explicitly given by its multiplication table.

▶ **Theorem 5.7.** *The problem* **MON** $\to$ **NFA$_{\mathbf{syn}}$** *is* NP*-complete.*

Proof sketch. The proof is conceptually similar to the one of Theorem 5.2. To show the problem to be in NP, one uses the algebraic characterization of $\mathsf{nsyn}(L)$ in Theorem 2.1(2) and translates the ensuing $\mathbf{JSL_f}$-diagram into **Dep**. To show NP-hardness, one reduces from **BICLIQUE COVER** via

$$(\mathcal{R}, k) \qquad \mapsto \qquad ((\mathsf{syn}(L), \mu_L, F_L), k),$$

where again $L = L(\mathsf{Open}(\mathcal{R}))$. ◀

Our complexity results indicate a trade-off, i.e. computing small subatomic nfas requires a less succinct representation of the input language. Generally, $|\mathsf{dfa}(L)|, |\mathsf{dfa}(L^r)| \leq |\mathsf{syn}(L)|$ and the syntactic monoid can be far larger – even for nuclear languages.

▶ **Example 5.8.** For any natural number n consider the dfa $A_n = (\{0, \ldots, n-1\}, \delta, 1, \{1\})$ over the alphabet $\Sigma = \{\pi, \tau\}$ with $\delta_\pi(i) = i+1 \mod n$ for $i = 0, \cdots n-1$, and $\delta_\tau(0) = 1$, $\delta_\tau(1) = 0$, $\delta_\tau(i) = i$ otherwise. Let $L_n = L(A_n)$ denote its accepted language. Then:

1. Both A_n and its reverse nfa are minimal dfas; in particular, $|\mathsf{dfa}(L_n)| = |\mathsf{dfa}(L_n^\mathsf{r})| = n$.
2. We have $|\mathsf{syn}(L_n)| = n!$. To see this, recall that $\mathsf{syn}(L_n)$ is the transition monoid of $A_n \cong \mathsf{dfa}(L_n)$. It is generated by the n-cycle $\delta_\pi = (0\ 1\ \cdots\ n-1)$ and the transposition $\delta_\tau = (0\ 1)$; then it equals the symmetric group S_n on n letters.
3. By part (1) the language L_n is *bideterministic* [30], i.e. accepted by a dfa whose reverse nfa is deterministic. This implies that the left derivatives of L_n are pairwise disjoint, so $\mathsf{SLD}(L_n)$ is a boolean algebra. In particular, L_n is a nuclear language.

We finally further justify the inputs of $\mathbf{DFA} + \mathbf{DFA}^\mathsf{r} \to \mathbf{NFA_{atm}}$ and $\mathbf{MON} \to \mathbf{NFA_{syn}}$: the two modified problems $\mathbf{DFA} \to \mathbf{NFA_{atm}}$ and $\mathbf{DFA} \to \mathbf{NFA_{syn}}$ where only a (single) dfa is given are computationally much harder.

▶ **Theorem 5.9.** $\mathbf{DFA} \to \mathbf{NFA_{atm}}$ *and* $\mathbf{DFA} \to \mathbf{NFA_{syn}}$ *are* PSPACE*-complete.*

Proof. This follows by inspecting Jiang and Ravikumar's [15] argument that $\mathbf{DFA} \to \mathbf{NFA}$ is PSPACE-complete. These authors give a polynomial-time reduction from the PSPACE-complete problem **UNIVERSALITY OF MULTIPLE DFAS**, which asks whether a given list $A_1, \ldots, A_n$ of dfas over the same alphabet Σ satisfies $\bigcup_i L(A_i) = \Sigma^*$. For any $A_1, \ldots, A_n$ they construct a dfa A over some alphabet Γ and a natural number k such that:

1. If $\bigcup_i L(A_i) \neq \Sigma^*$, then every nfa accepting $L(A)$ requires at least $k+1$ states.
2. If $\bigcup_i L(A_i) = \Sigma^*$, then there exists an nfa accepting $L(A)$ with k states.

In the proof of (2), an explicit k-state nfa $N = (Q, \delta, \{q_0\}, F)$ with $L(N) = L(A)$ is given, see [15, Fig. 1]. It has the property that, after ε-elimination, for every state q there exists $w \in \Gamma^*$ with $\delta_w[q_0] = \{q\}$. This implies that every state q accepts a left derivative $w^{-1}L(N)$, i.e. N is a *residual* nfa [7]. In particular, N is both atomic and subatomic. Consequently, $(A_1, \ldots, A_n) \mapsto (A, k)$ is also a reduction to both $\mathbf{DFA} \to \mathbf{NFA_{atm}}$ and $\mathbf{DFA} \to \mathbf{NFA_{syn}}$. ◀

6 Applications

We conclude this paper by outlining some useful consequences of our NP-completeness results concerning the computation of small nfas for specific classes of regular languages.

6.1 Nuclear Languages

As shown above, nuclear languages form a natural common generalization of bideterministic, biRFSA, and lattice languages. Let $\mathbf{DFA} + \mathbf{DFA}^\mathsf{r} \to \mathbf{NFA}$ be the variant of $\mathbf{DFA} + \mathbf{DFA}^\mathsf{r} \to \mathbf{NFA_{atm}}$ where the target nfas are arbitrary, i.e. the task is to decide $\mathsf{ns}(L(A)) \leq k$. Then:

▶ **Theorem 6.1.** *For nuclear languages, the problem* $\mathbf{DFA} + \mathbf{DFA}^\mathsf{r} \to \mathbf{NFA}$ *is* NP*-complete.*

In fact, by Proposition 4.5(1) we have $\mathsf{ns}(L) = \mathsf{dim}(\mathcal{DR}_L)$ for nuclear languages, so NP certificates are given by biclique covers. The NP-hardness proof is identical to the one of Theorem 5.2: the reduction involves a lattice language, which is nuclear by Lemma 4.4.

6.2 Unary languages

For unary regular languages $L \subseteq \{a\}^*$, every two-sided derivative $(a^i)^{-1}L(a^j)^{-1}$ is equal to the left derivative $(a^{i+j})^{-1}L$. Therefore, we have $\mathsf{natm}(L) = \mathsf{nsyn}(L)$ and the minimal dfa for L is the dfa structure of the syntactic monoid. From Theorem 5.7 we thus derive

▶ **Theorem 6.2.** *For unary languages, the problem* $\mathbf{DFA} \to \mathbf{NFA_{syn}}$ *is in* NP*.*

This theorem generalizes the best-known complexity result for unary nfas, which asserts that the problem $\mathbf{DFA} \to \mathbf{NFA}$ is in NP for *unary cyclic languages* [13], i.e. unary regular languages whose minimal dfa is a cycle. In fact, for any such language L we have shown in [26, Example 5.1] that $\mathsf{nsyn}(L) = \mathsf{ns}(L)$, hence $\mathbf{DFA} \to \mathbf{NFA}$ coincides with $\mathbf{DFA} \to \mathbf{NFA_{syn}}$.

6.3 Group languages

A regular language is called a *group language* if its syntactic monoid forms a group. Several equivalent characterizations of group languages are known; for instance, they are precisely the languages accepted by measure-once quantum finite automata [3]. Concerning their state-minimal (sub)atomic acceptors, we have the following result:

▶ **Proposition 6.3.** *For any group language* L*, we have* $\mathsf{nsyn}(L) = \mathsf{natm}(L)$*.*

Therefore, Theorem 5.2 implies

▶ **Theorem 6.4.** *For group languages,* $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{syn}}$ *is in* NP*.*

The complexity of the general $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA_{syn}}$ problem is left as an open problem.

7 Conclusion and Future Work

Approaching from an algebraic and category-theoretic angle we have studied the complexity of computing small (sub)atomic nondeterministic machines. We proved this to be much more tractable than the general case, viz. NP-complete as opposed to PSPACE-complete, provided that one works with a representation of the input language by a pair of dfas or a finite monoid, respectively. There are several interesting directions for future work.

The particular form of our main two NP-complete problems suggests an investigation of their variants $\mathbf{DFA} + \mathbf{DFA}^{\mathsf{r}} \to \mathbf{NFA}$ and $\mathbf{MON} \to \mathbf{NFA}$ computing unrestricted nfas. The reductions used in the proof of Theorem 5.2 and 5.7 show both problems to be NP-hard, and we have seen in Theorem 6.1 that they are in NP for nuclear languages. The complexity of the general case is left as an open problem.

The classical algorithm for state minimization of nfas is the Kameda-Weiner method [18], recently given a fresh perspective based on atoms of regular languages [29]. The algorithm involves an enumeration of biclique covers of the dependency relation $\mathcal{DR}_L$. Since our base equivalence $\mathbf{JSL_f} \simeq \mathbf{Dep}$ reveals a close relationship between biclique covers and semilattice morphisms (e.g. Lemma 3.9), we envision a purely algebraic account of the Kameda-Weiner method. We should also compare our canonical machines to the Universal Automaton [20], a language-theoretic presentation of the Kameda-Weiner algorithm. For example, our morphisms preserve the language whereas the Universal Automaton uses simulations.

Finally, the classes of nuclear and lattice languages – introduced as technical tools for our NP-completeness proofs – deserve to be studied in their own right. For instance, we expect to uncover connections between lattice languages and the characterization of finite simple non-unital semirings which are not rings [31, Theorem 1.7].

References

1 Jiří Adámek, Stefan Milius, Robert S. R. Myers, and Henning Urbat. On continuous nondeterminism and state minimality. In Bart Jacobs, Alexandra Silva, and Sam Staton, editors, *Proc. 30th Conference on Mathematical Foundations of Programming Science (MFPS'14)*, volume 308 of *Electron. Notes Theor. Comput. Sci.*, pages 3–23. Elsevier, 2014.

2 Michael A. Arbib and Ernest G. Manes. Fuzzy machines in a category. *Bulletin of the Australian Mathematical Society*, 13(2):169–210, 1975.

3 Alex Brodsky and Nicholas Pippenger. Characterizations of 1-way quantum finite automata. *SIAM J. Comput.*, 31:73–91, 1999.

4 Janusz Brzozowski and Hellis Tamm. Theory of átomata. *Theoretical Computer Science*, 539:13–27, 2014.

5 Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964.

6 Marek Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47:149–158, 1986.

7 François Denis, Aurélien Lemay, and Alain Terlutte. Residual finite state automata. In Afonso Ferreira and Horst Reichel, editors, *Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01)*, pages 144–157. Springer, 2001.

8 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

9 Jaco Geldenhuys, Brink van der Merwe, and Lynette van Zijl. Reducing nondeterministic finite automata with SAT solvers. In Anssi Yli-Jyrä, András Kornai, Jacques Sakarovitch, and Bruce Watson, editors, *Proc. 8th International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP'09)*, pages 81–92. Springer, 2010.

10 Hermann Gruber and Markus Holzer. Finding lower bounds for nondeterministic state complexity is hard. In Oscar H. Ibarra and Zhe Dang, editors, *Proc. 10th International Conference on Developments in Language Theory (DLT'06)*, pages 363–374. Springer, 2006.

11 Hermann Gruber and Markus Holzer. Computational complexity of NFA minimization for finite and unary languages. In Remco Loos, Szilárd Zsolt Fazekas, and Carlos Martín-Vide, editors, *Proc. 1st International Conference on Language and Automata Theory and Applications (LATA'07)*, pages 261–272. Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, 2007.

12 D.A. Higgs and K.A. Rowe. Nuclearity in the category of complete semilattices. *Journal of Pure and Applied Algebra*, 57(1):67–78, 1989.

13 Tao Jiang, Edward McDowell, and B. Ravikumar. The structure and complexity of minimal NFA's over a unary alphabet. *International Journal of Foundations of Computer Science*, 02(02):163–182, 1991.

14 Tao Jiang and B. Ravikumar. Minimal NFA problems are hard. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez Artalejo, editors, *Proc. 18th International Colloquium on Automata, Languages, and Programming (ICALP'91)*, pages 629–640. Springer, 1991.

15 Tao Jiang and B. Ravikumar. Minimal NFA problems are hard. *SIAM Journal on Computing*, 22(6):1117–1141, 1993.

16 Peter Jipsen. Categories of algebraic contexts equivalent to idempotent semirings and domain semirings. In Wolfram Kahl and Timothy G. Griffin, editors, *Proc. 13th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS'12)*, pages 195–206. Springer, 2012.

17 Peter T. Johnstone. *Stone spaces*. Cambridge University Press, 1982.

18 T. Kameda and P. Weiner. On the state minimization of nondeterministic finite automata. *IEEE Transactions on Computers*, C-19(7):617–627, 1970.

19 Michel Latteux, Yves Roos, and Alain Terlutte. Minimal NFA and biRFSA languages. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 43(2):221–237, 2009.

20 Sylvain Lombardy and Jacques Sakarovitch. The universal automaton. In *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, pages 457–504, 2008.

21 Saunders Mac Lane. *Categories for the working mathematician*. Springer, 2 edition, 1998.

22 Andreas Malcher. Minimizing finite automata is computationally hard. *Theor. Comput. Sci.*, 327(3):375–390, 2004.

23 George Markowsky. The factorization and representation of lattices. *Transactions of the American Mathematical Society*, 203:185–200, 1975.

24 M. Andrew Moshier. A relational category of formal contexts (preprint), 2016.

25 Robert S. R. Myers, Jiří Adámek, Stefan Milius, and Henning Urbat. Coalgebraic constructions of canonical nondeterministic automata. *Theor. Comput. Sci.*, 604:81–101, 2015.

26 Robert. S. R. Myers, Stefan Milius, and Henning Urbat. Nondeterministic syntactic complexity. In Stefan Kiefer and Christine Tasson, editors, *Proc. 24th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'21)*. Springer, 2021. `arXiv:2101.03039`.

27 Jean-Éric Pin. Mathematical foundations of automata theory. Available at `http://www.liafa.jussieu.fr/~jep/PDF/MPRI/MPRI.pdf`, September 2020.

28 K. A. Rowe. Nuclearity. *Canadian Mathematical Bulletin*, 31(2):227–235, 1988.

29 Hellis Tamm. New interpretation and generalization of the Kameda-Weiner method. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *Proc. 43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, volume 55 of *LIPIcs*, pages 116:1–116:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

30 Hellis Tamm and Esko Ukkonen. Bideterministic automata and minimal representations of regular languages. *Theoretical Computer Science*, 328(1):135–149, 2004.

31 Jens Zumbrägel. Classification of finite congruence-simple semirings with zero. *Journal of Algebra and Its Applications*, 7, March 2007.