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 \( \text{PSPACE} \)-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 \( \text{NP} \)-complete problem. The \( \text{NP} \) 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 \( \text{NP} \)-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


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 \( \text{PSPACE} \)-complete \cite{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 \( \text{PSPACE} \) bound to an \( \text{NP} \) 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 \cite{26} we observed that almost all known results on the structure of small nfas, e.g. for unary \cite{6,13}, bideterministic \cite{30},...
topological [1] and biRFSAs 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 \( \text{MON} \rightarrow \text{NFA}_{\text{syn}} \), our main result is:

**Theorem.** The problem \( \text{MON} \rightarrow \text{NFA}_{\text{syn}} \) is \( \text{NP} \)-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 \( \text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}} \), we get:

**Theorem.** The problem \( \text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}} \) is \( \text{NP} \)-complete.

The short certificates witnessing that both problems are in \( \text{NP} \) are solutions of equations involving relations over the syntactic congruence or the Nerode left congruence, respectively.

The above two theorems sharply contrast the \( \text{PSPACE} \)-completeness of the general DFA to NFA conversion problem, but also previous results on its sub-\( \text{PSPACE} \) 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:

\[
\begin{array}{c}
\text{JSL}_{\text{f}} \quad \overset{\sim}{\longleftrightarrow} \quad \text{Structure theory} \quad \overset{\sim}{\longleftrightarrow} \quad \text{Complexity theory} \\
\text{JSL}_{\text{f}} \quad \overset{\sim}{\longleftrightarrow} \quad \text{Dep}.
\end{array}
\]

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 \( \text{JSL}_{\text{f}} \) of finite semilattices. Similarly, the *complexity theory* of (sub)atomic NFAs developed in the present paper rests on the equivalence between \( \text{JSL}_{\text{f}} \) and a category \( \text{Dep} \) (see Definition 3.1) that yields succinct relational representations of finite semilattices by their irreducible elements. To derive the \( \text{NP} \)-completeness theorems, we reinterpret semilattice automata associated to (sub)atomic NFAs inside \( \text{Dep} \). 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 \emph{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 \emph{meet-generators} of \( S \).

Let \( 2 = \{0, 1\} \) be the two-element semilattice with \( 0 \leq 1 \). Morphisms \( i : 2 \to S \) correspond to elements of \( S \) via \( i \mapsto i(1) \). Morphisms \( f : S \to 2 \) correspond to \emph{prime filters} via \( f \mapsto f^{-1}[1] \). If \( S \) is finite, these are precisely the subsets \( F_s = \{ s \in S : s \notin S \} \) for any \( s \in S \).

We denote by \( JSL \) the category of join-semilattices and their morphisms. Its full subcategory \( JSL_f \) of finite semilattices is self-dual \cite{17}: there is an equivalence functor

\[
JSL_f^\op \cong JSL_f
\]

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

\[
f(s) \leq_T t \quad \text{iff} \quad s \leq_S f^\circ(t)
\]

for all \( s \in S \) and \( t \in T \). The morphism \( f \) is injective (equivalently a \( JSL_f \)-monomorphism) iff \( f^\circ \) is surjective (equivalently a \( JSL_f \)-epimorphism).

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

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

The \emph{converse} of \( R \) is the relation \( \hat{R} \subseteq Y \times X \) (alternatively \( R^\circ \)) where \( \hat{R}(y, x) \) iff \( R(x, y) \) for \( x \in X \) and \( y \in Y \). The \emph{composite} of \( R \subseteq X \times Y \) and \( S \subseteq Y \times Z \) is the relation \( R; S \subseteq X \times Z \) where \( R(x, z) \) iff there exists \( y \in Y \) with \( R(x, y) \) and \( S(y, z) \). Let \( \text{Rel} \) 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 \( \text{id}_X \subseteq X \times X \) with \( \text{id}_X(x, y) \) iff \( x = y \).

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

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

\[
LD(L) \subseteq SLD(L) \subseteq BLD(L) \subseteq BLRD(L)
\]
where \( \text{LD}(L) = \{ u^{-1}L : u \in \Sigma^* \} \) is the set of left derivatives, and \( \text{SLD}(L) \), \( \text{BLD}(L) \), \( \text{BLRD}(L) \) 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 \( \text{LD}(L) \) is a finite set; then the other three sets are finite too. The finite semilattices \( \text{SLD}(L) \) and \( \text{SLD}(L') \) are related by the fundamental isomorphism

\[
\text{dr}_L : [\text{SLD}(L')]^\circ \cong \text{SLD}(L), \quad K \mapsto (K^c)^{-1}L,
\]

(2.1)

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

\[
\mathcal{DR}_L \subseteq \text{LD}(L) \times \text{LD}(L'), \quad \mathcal{DR}_L(u^{-1}L, v^{-1}L') :\iff uv' \in L \text{ for } u, v \in \Sigma^*.
\]

(2.2)

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

\[
\mathcal{DR}_L(u^{-1}L, v^{-1}L') \iff u^{-1}L \not\subseteq \text{dr}_L(v^{-1}L') \text{ for } u, v \in \Sigma^*.
\]

(2.3)

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

\[
u \sim_L v \text{ iff } \forall x \in \Sigma^* : u \in x^{-1}L \Leftrightarrow v \in x^{-1}L \text{ iff } (u')^{-1}L' = (v')^{-1}L'.
\]

(2.4)

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

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

\[
u \equiv_L v \text{ iff } \forall x, y \in \Sigma^* : u \in x^{-1}Ly^{-1} \Leftrightarrow v \in x^{-1}Ly^{-1}.
\]

(2.5)

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

Automata. Fix a finite alphabet \( \Sigma \). 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 \( \delta_a[q] \cap F \neq \emptyset \), 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 \( \delta_c = \text{id}_Q \). 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 \( \delta_a : Q \rightarrow Q \). A dfa is a JSL-dfa if \( Q \) is a finite semilattice, each \( \delta_a : Q \rightarrow Q \) is a semilattice morphism, and \( F \subseteq Q \) forms a prime filter. It is often useful to represent a JSL-dfa in terms of morphisms

\[
2 \xrightarrow{i} Q \xrightarrow{\delta_0} Q \xrightarrow{f} 2
\]

where \( i \) is the unique morphism with \( i(1) = q_0 \) and \( f \) is given by \( f(q) = 1 \) iff \( q \in F \). A JSL-dfa morphism from \( A = (Q, \delta, i, f) \) to \( A' = (Q', \delta', i', f') \) is a JSL\(_I\)-morphism \( h : Q \rightarrow Q' \) 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 \( h : Q \hookrightarrow Q' \) is the inclusion map, then \( A \) is called a sub JSL-dfa of \( A' \).
Fix a regular language \( L \). Viewed as a \( \cup \)-semilattice, \( \text{BLRD}(L) \) carries the structure of a \( \text{JSL} \)-\( \text{dfa} \) with transitions \( K \stackrel{\alpha}{\to} \alpha^{-1}K, \) initial state \( L \), and finals \( \{ K : \varepsilon \in K \} \). This restricts to sub \( \text{JSL} \)-\( \text{dfas} \) \( \text{BLD}(L) \) and \( \text{SLD}(L) \). Moreover \( \text{LD}(L) \) forms a sub-\( \text{dfa} \) of \( \text{SLD}(L) \), well-known [5] to be the state-minimal \( \text{dfa} \) for \( L \), so we denote it by \( \text{dfa}(L) \). The syntactic monoid \( \text{syn}(L) \) is isomorphic to the transition monoid of \( \text{dfa}(L) \), i.e. the monoid of all extended transition maps \( \delta_w : \text{LD}(L) \to \text{LD}(L) (w \in \Sigma^*) \) with multiplication given by composition [27].

Analogously \( \text{SLD}(L) \) is the state-minimal \( \text{JSL} \)-\( \text{dfa} \) for \( L \). Up to isomorphism, it is the unique \( \text{JSL} \)-\( \text{dfa} \) for \( L \) that is \( \text{JSL} \)-\( \text{reachable} \) (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).

\( \text{Nfas} \), \( \text{dfas} \) and \( \text{JSL} \)-\( \text{dfas} \) are expressively equivalent and accept precisely the regular languages. In particular, to every \( \text{JSL} \)-\( \text{dfa} \) \( A = (Q, \delta, q_0, F) \) one can associate an equivalent \( \text{nfa} \) \( J(A) \), the \( \text{nfa} \) 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 \( q \leq s \), 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 \( N_L = J(\text{SLD}(L)) \), the \( \text{nfa} \) of join-irreducibles of its minimal \( \text{JSL} \)-\( \text{dfa} \).

Atomic and subatomic nfas. An \( \text{nfa} \) accepting the language \( L \subseteq \Sigma^* \) is called atomic [4] if each state accepts a language from \( \text{BLRD}(L) \), and subatomic [26] if each state accepts a language from \( \text{BLRD}(L) \). The nondeterministic atomic complexity \( \text{natm}(L) \) of a regular language \( L \) is the least number of states of any atomic \( \text{nfa} \) accepting \( L \). The nondeterministic syntactic complexity \( \text{nsyn}(L) \) is the least number of states of any subatomic \( \text{nfa} \) accepting \( L \). Subatomic \( \text{nfas} \) are intimately connected to syntactic monoids: the atoms of \( \text{BLRD}(L) \) are the elements of \( \text{syn}(L) \), so an \( \text{nfa} \) accepting \( L \) is subatomic if its individual states accept unions of syntactic congruence classes. Additionally \( \text{nsyn}(L) \) can be characterized via boolean representations of \( \text{syn}(L) \), i.e. monoid morphisms \( \varphi : \text{syn}(L) \to \text{JSL}_\Sigma(S, S) \) 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 \( \text{ns}(L) \), i.e. the least number of states of any (unrestricted) \( \text{nfa} \) accepting \( L \). In particular,

\[
\text{dim}(\text{DR}_L) \leq \text{ns}(L) \leq \text{nyn}(L) \leq \text{natm}(L). \tag{2.6}
\]

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 \( \text{nfas} \) can be characterized in terms of \( \text{JSL} \)-\( \text{dfas} \). The following theorem involves two commuting diagrams of semilattice morphisms, whose lower and upper paths are the canonical \( \text{JSL} \)-\( \text{dfas} \) described earlier.

\begin{itemize}
  \item \textbf{Theorem 2.1.} Let \( L \subseteq \Sigma^* \) be a regular language.
  \begin{enumerate}
    \item \( \text{natm}(L) \) is the least number \( k \) such that there exists a finite semilattice \( S \) with \( |J(S)| \leq k \) and \( \text{JSL}_\Sigma \)-morphisms \( p, q \) and \( \tau_a (a \in \Sigma) \) making the left-hand diagram below commute.
    \item \( \text{nyn}(L) \) is the least number \( k \) such that there exists a finite semilattice \( S \) with \( |J(S)| \leq k \) and \( \text{JSL}_\Sigma \)-morphisms \( p, q \) and \( \tau_a (a \in \Sigma) \) making the right-hand diagram below commute.
  \end{enumerate}
\end{itemize}
We now prove that this extends to an identity morphism for \( \mathcal{A} \). The identity morphism for \( \mathcal{A} \) is an equivalence that extends the left diagram commute. Then \( A = (\mathcal{A}, \tau, p \circ i, f' \circ q) \) is a \( \mathcal{A} \)-dfa and \( p: \mathcal{A} \to \mathcal{B} \) and \( q: \mathcal{A} \to \mathcal{B} \) are \( \mathcal{A} \)-dfa morphisms. Since \( \mathcal{A} \)-dfa morphisms preserve the accepted language, and every state \( K \in \mathcal{B} \) accepts the language \( K \), it follows that \( A \) accepts \( L \) and every state of \( A \) accepts a language from \( \mathcal{B} \). 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 \( \mathcal{S} = \text{langs}(N) \) of all languages \( \mathcal{L}(N, X) \) accepted by subsets \( X \subseteq Q \). Note that \( \mathcal{A}(\mathcal{S}) \subseteq \mathcal{L}(\mathcal{S}) \subseteq \mathcal{B}(\mathcal{L}) \): the first inclusion holds because \( u^{-1}L = L(N, \delta_u[I]) \in \mathcal{S} \) for every \( u \in \Sigma^* \), and the second one because \( N \) is atomic. We define the semilattice endomorphisms \( \tau_a: \mathcal{S} \to \mathcal{S} \) by \( \tau_a(K) = a^{-1}K \) for \( K \in \mathcal{S} \).

Letting \( p: \mathcal{A}(\mathcal{S}) \to \mathcal{S} \) and \( q: \mathcal{S} \to \mathcal{B}(\mathcal{S}) \) 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 \( \mathcal{A} \)-dfa-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 \( \leq_S \subseteq J(S) \times M(S) \) between join-irreducibles and meet-irreducibles.

We now prove that this extends to an equivalence between the category \( \mathcal{A} \) of finite semilattices and another category called \( \text{Dep} \). 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 \( \text{Dep} \) are the relations \( \mathcal{R} \subseteq \mathcal{R}_a \times \mathcal{R}_t \) between finite sets. Far less obviously,

- a morphism \( \mathcal{P}: \mathcal{R} \to \mathcal{S} \) is a relation \( \mathcal{P} \subseteq \mathcal{R}_a \times \mathcal{S}_t \) that factorizes through \( \mathcal{R} \) and \( \mathcal{S} \), i.e. the left Rel-diagram below commutes for some \( \mathcal{P}_f \subseteq \mathcal{R}_a \times \mathcal{S}_t \) and \( \mathcal{P}_u \subseteq \mathcal{S}_t \times \mathcal{R}_t \).

The identity morphism for \( \mathcal{R} \) is \( \text{id}_\mathcal{R} = \mathcal{R} \), see the central diagram below. The composite \( \mathcal{P}_\mathcal{Q}: \mathcal{R} \to \mathcal{T} \) of \( \mathcal{P}: \mathcal{R} \to \mathcal{S} \) and \( \mathcal{Q}: \mathcal{S} \to \mathcal{T} \) 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, \( \mathcal{P}_1 \rightarrow \mathcal{Q}_1 ; \mathcal{T}_1 = \mathcal{P}_1 ; \mathcal{Q}_1 \). 
(Note that we use the symbol \( \uparrow \) for composition in \( \text{Dep} \) and ; for composition in \( \text{Rel} \), and recall
that \(-\)\(^\uparrow\) denotes the converse relation.)

\[
\begin{array}{cccc}
\mathcal{R}_1 & \longrightarrow & \mathcal{S}_1 & \longrightarrow & \mathcal{T}_1 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\mathcal{R}_2 & \longrightarrow & \mathcal{S}_2 & \longrightarrow & \mathcal{T}_2 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\mathcal{R}_3 & \longrightarrow & \mathcal{S}_3 & \longrightarrow & \mathcal{T}_3
\end{array}
\]

One readily verifies that \( \text{Dep} \) is a well-defined category; in particular, the composition is
independent of the choice of the lower and upper witnesses \((-\)\(_1\)) and \((-\)\(_u\)).

\begin{itemize}
\item \textbf{Remark 3.2.}
1. Using the converse upper witness may seem strange. Although technically unnecessary,
it fits the self-duality of \( \text{Dep} \) taking the converse on objects and morphisms. Moreover \( f; \notin S = \notin T; f_\ast \) for any \( J\text{SL}_f \)-morphism \( f : S \rightarrow T \) via the adjoint relationship; that is, \( f \)
duces a \( \text{Dep} \)-morphism from \( \notin S \) to \( \notin T \) with lower witness \( f \) and upper witness \( f_\ast \).

2. The witnesses of a \( \text{Dep} \)-morphism \( \mathcal{P} : \mathcal{R} \rightarrow \mathcal{S} \) are closed under unions. The maximal
lower witness \( \mathcal{P}_- \subseteq \mathcal{R}_x \times \mathcal{S}_x \) is given by
\[
\mathcal{P}_-(x,y) :\iff S[y] \subseteq \mathcal{P}[x] \quad \text{for} \quad x \in \mathcal{R}_x, y \in \mathcal{S}_x,
\]
and the maximal upper witness \( \mathcal{P}_+ \subseteq \mathcal{S}_x \times \mathcal{R}_x \) by
\[
\mathcal{P}_+(y,x) :\iff \mathcal{R}[x] \subseteq \mathcal{P}[y] \quad \text{for} \quad x \in \mathcal{R}_x, y \in \mathcal{S}_x.
\]
\end{itemize}

\begin{itemize}
\item \textbf{Theorem 3.3 (Fundamental equivalence).} \textit{The categories \( J\text{SL}_f \) and \( \text{Dep} \) are equivalent.}
1. \textit{The equivalence functor \( \text{Pirr} : J\text{SL}_f \rightarrow \text{Dep} \) maps a finite semilattice \( S \) to the \( \text{Dep} \)-object}
\[
\text{Pirr}(S) := \notin S \subseteq J(S) \times M(S),
\]
\textit{and a \( J\text{SL}_f \)-morphism \( f : S \rightarrow T \) to the \( \text{Dep} \)-morphism}
\[
\text{Pirr}(f) : \text{Pirr}(S) \rightarrow \text{Pirr}(T), \quad \text{Pirr}(f)(j,m) :\iff f(j) \notin T m \quad \text{for} \ j \in J(S), m \in M(T).
\]

2. \textit{The inverse Open: \( \text{Dep} \rightarrow J\text{SL}_f \) maps a \( \text{Dep} \)-object \( \mathcal{R} \) to its semilattice of open sets}
\[
\text{Open}(\mathcal{R}) := (\{ \mathcal{R}[X] : X \subseteq \mathcal{R}_x \}, \subseteq),
\]
\textit{and a \( \text{Dep} \)-morphism \( \mathcal{P} : \mathcal{R} \rightarrow \mathcal{S} \) to the \( J\text{SL}_f \)-morphism}
\[
\text{Open}(\mathcal{P}) : \text{Open}(\mathcal{R}) \rightarrow \text{Open}(\mathcal{S}), \quad \text{Open}(\mathcal{P})(O) := \mathcal{P}_+[O] \quad \text{for} \ O \in \text{Open}(\mathcal{R}),
\]
where \( \mathcal{P}_+ \subseteq \mathcal{S}_x \times \mathcal{R}_x \) is the maximal upper witness of \( \mathcal{P} \).
\end{itemize}

\begin{itemize}
\item \textbf{Remark 3.4.} In the definition of \( \text{Pirr}(S) \) one may replace \( J(S) \) and \( M(S) \) by any two
sets \( J,M \subseteq S \) of join- and meet-generators modulo \( \text{Dep} \)-isomorphism. Indeed, since the
equivalence functor \text{Open} reflects isomorphisms, this follows immediately from the \( J\text{SL}_f \)-
isomorphism \( \text{Open}(\notin S \cap J \times M) \cong \text{Open}(\notin S \cap J(S) \times M(S)) \) given by \( O \mapsto O \cap M(S) \).
\end{itemize}

\begin{itemize}
\item \textbf{Remark 3.5.} Bijectively relabeling the domain and codomain of a relation defines a
\( \text{Dep} \)-isomorphism, the witnesses being the relabelings.
\end{itemize}
We now show that for every regular language \( L \), the semilattices \( \text{SLD}(L) \), \( \text{BLD}(L) \) and \( \text{BLRD}(L) \) equipped with their canonical \( \text{JSL-dfa} \) structure (see Section 2) translate under the equivalence functor \( \text{Pirr} \) 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 \( \text{JSL-dfa} \)s and their corresponding \( \text{Dep} \)-structures. |
|-----------------|-----------------|
| \( \text{JSL}_L \) | \( \text{Dep} \) |
| \( 2 \xrightarrow{1} \text{SLD}(L) \xrightarrow{a} \text{SLD}(L) \xrightarrow{1} 2 \) | \( \text{id}_1 \xrightarrow{1} \text{DR}_L \xrightarrow{\text{DR}_{L,a}} \text{DR}_L \xrightarrow{a} \text{id}_1 \) |
| \( 2 \xrightarrow{1} \text{BLD}(L) \xrightarrow{a} \text{BLD}(L) \xrightarrow{1} 2 \) | \( \text{id}_1 \xrightarrow{1} \text{id}_{\Sigma^* \sim_L} \xrightarrow{a} \text{id}_{\Sigma^* \sim_L} \xrightarrow{a} \text{id}_1 \) |
| \( 2 \xrightarrow{1} \text{BLRD}(L) \xrightarrow{a} \text{BLRD}(L) \xrightarrow{1} 2 \) | \( \text{id}_1 \xrightarrow{1} \text{id}_{\text{syn}(L)} \xrightarrow{a} \text{id}_{\text{syn}(L)} \xrightarrow{1} \text{id}_1 \) |

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

\[
\text{Pirr}(\text{SLD}(L))(u^{-1}L, \text{dr}_L(v^{-1}L')) \overset{\text{def.}}{=} u^{-1}L \not\subset \text{dr}_L(v^{-1}L') \overset{(2.3)}{=} \text{DR}_L(u^{-1}L, v^{-1}L')
\]

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

\[
\text{Pirr}(\text{SLD}(L)) \text{ is a bijective relabeling of } \text{DR}_L \text{ restricted to } J(\text{SLD}(L)) \times J(\text{SLD}(L')).
\]

By Remark 3.4 we know \( \text{Pirr}(\text{SLD}(L)) \) is isomorphic to the domain-codomain extension \( \not\subset \subseteq \text{LD}(L) \times \text{dr}_L[\text{LD}(L')] \) and thus also to the dependency relation \( \text{DR}_L \) by Remark 3.5. Then the \( \text{JSL-dfa} \) structure of the semilattice \( \text{SLD}(L) \) translates into the category of dependency relations as shown in Table 1, where \( \text{id}_1 \) is the identity relation on \( 1 = \{ \ast \} \) and

\[
I \subseteq 1 \times \text{LD}(L'), \quad \text{DR}_L \subseteq \text{LD}(L) \times \text{LD}(L'), \quad F \subseteq \text{LD}(L) \times 1,
\]

\[
I(\ast, v^{-1}L') \Leftrightarrow v \in L', \quad \text{DR}_L(u^{-1}L, v^{-1}L') \Leftrightarrow uv' \in L, \quad F(u^{-1}L, \ast) \Leftrightarrow u \in L.
\]

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

\[
\text{Pirr}(\text{BLD}(L))([u]_{\sim_L}, [v]_{\sim_L}) \overset{\text{def.}}{=} [u]_{\sim_L} \not\subset [v]_{\sim_L} \iff [u]_{\sim_L} = [v]_{\sim_L}.
\]

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

\[
I' \subseteq 1 \times \Sigma^*/\sim_L, \quad \text{DR}'_L \subseteq \Sigma^*/\sim_L \times \Sigma^*/\sim_L,
\]

\[
I'(\ast, [u]_{\sim_L}) \Leftrightarrow u \in L, \quad \text{DR}'_L([u]_{\sim_L}, [v]_{\sim_L}) \Leftrightarrow [v]_{\sim_L} \subseteq a^{-1}[u]_{\sim_L}, \quad F'([u]_{\sim_L}, \ast) \Leftrightarrow u \sim_L \varepsilon.
\]

We note that the above relations induce an nfa

\[
(\Sigma^*/\sim_L, (\text{DR}'_L)_{a \in \Sigma}, I'[\ast], F'[\ast]) \quad \text{known as the } \text{\d{a}tomaton} \text{ for the language } L \ [4].
\]

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

\[
I'' \subseteq 1 \times \text{syn}(L), \quad \text{DR}''_L \subseteq \text{syn}(L) \times \text{syn}(L),
\]

\[
I''(\ast, [u]_{\equiv_L}) \Leftrightarrow u \in L, \quad \text{DR}''_L([u]_{\equiv_L}, [v]_{\equiv_L}) \Leftrightarrow [v]_{\equiv_L} \subseteq a^{-1}[u]_{\equiv_L}, \quad F''([u]_{\equiv_L}, \ast) \Leftrightarrow u \equiv_L \varepsilon.
\]
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. $\dim(\mathcal{R})$ is the least $|J(S)|$ of any injective $\text{JSL}_f$-morphism $\text{Open}(\mathcal{R}) \to S$.
2. $\dim(\mathcal{R})$ is invariant under isomorphism, i.e. $\mathcal{R} \cong S$ in $\text{Dep}$ implies $\dim(\mathcal{R}) = \dim(S)$.

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

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

$$\text{in}_\mathcal{R} : \mathcal{P}(\mathcal{R}_t) \to \mathcal{P}(\mathcal{R}_s) \quad Y \mapsto \bigcup \{\mathcal{R}[X] : X \subseteq \mathcal{R}_s \text{ and } \mathcal{R}[X] \subseteq Y\}.$$  

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

▶ **Lemma 3.11.** Let $\mathcal{R} \subseteq \mathcal{R}_s \times \mathcal{R}_t$ be a relation between finite sets.
1. $J(\text{Open}(\mathcal{R}))$ consists of all sets $\mathcal{R}[x]$ ($x \in \mathcal{R}_s$) 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. $M(\text{Open}(\mathcal{R}))$ consists of all sets $\text{in}_\mathcal{R}(\mathcal{R}_t \setminus \{y\})$ such that $\mathcal{R}[y]$ lies in $J(\text{Open}(\mathcal{R}))$.

### 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 NF-completeness proofs in Section 5. Their definition rests on the notion of a nuclear morphism in $\text{JSL}_f$, 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 $\text{JSL}_f$-morphism $f : S \rightarrow T$ is nuclear if it factorizes through a finite distributive lattice, i.e. $f = (S \xrightarrow{\delta} D \xrightarrow{\sim} T)$ for some finite distributive lattice $D$ and $\text{JSL}_f$-morphisms $g,h$. A regular language $L \subseteq \Sigma^*$ is nuclear if the transition morphisms $\delta_a = a^{-1}(-) : \text{SLD}(L) \to \text{SLD}(L) (a \in \Sigma)$ of its minimal $\text{JSL}$-dfa are nuclear.

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

$$\Sigma := \{\langle j \rangle : j \in J(S)\} \cup \{|m| : m \in M(S)\} \quad \text{and} \quad L(S) := \bigcap_{j \leq s \leq m} \Sigma^j(j||m)||\Sigma^s.$$  

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

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

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 $\text{ns}(L) = \dim(\text{DR}_L)$.
2. If $L = L(S)$ is a lattice language then $\text{natm}(L) = \text{nsyn}(L) = \text{ns}(L) = \dim(\text{DR}_L)$.

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:

$\text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}}$

**Input:** Two dfas $A$ and $B$ such that $L(A) = L(B)'$ and a natural number $k$.

**Task:** Decide whether there exists a $k$-state atomic nfa equivalent to $A$, i.e. $\text{natm}(L(A)) \leq k$.

**Remark 5.1.** Taking mutually reverse dfas $(A, B)$ as input permits an efficient computation of the dependency relation $\mathcal{DR}_L \subseteq \text{LD}(L) \times \text{LD}(L')$ 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 $\text{LD}(L)$ and $\text{LD}(L')$. For $p \in Q_A$ choose some $w_A(p) \in \Sigma^* \text{ sending the initial state to } p$; analogously choose $w_B(q) \in \Sigma^*$ for $q \in Q_B$. Then $\mathcal{DR}_L$ is a bijective relabeling of

$$\widetilde{\mathcal{DR}}_L \subseteq Q_A \times Q_B \quad \text{where } \widetilde{\mathcal{DR}}_L(p, q) :\Longleftrightarrow A \text{ accepts } w_A(p)w_B(q)'$$

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

**Theorem 5.2.** The problem $\text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}}$ 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 $\text{JSL}_f$ and $\text{Dep}$.

**Proposition 5.3.** The problem $\text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}}$ 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 $L(A) = L(B)'$. In fact, this condition is equivalent to $\overline{L}(A) \cap L(B)' = \overline{L}(B) \cap L(A)' = \emptyset$. Using the standard methods for complementing dfas and reversing and intersecting nfas, one can construct nfas for $\overline{L}(A) \cap L(B)'$ and $\overline{L}(B) \cap L(A)'$ of size polynomial in $|A|$ and $|B|$, the number of states of $A$ and $B$, and check for emptiness by verifying that no final state is reachable from the initial states.

2. Let $A$ and $B$ be dfas accepting the languages $L$ and $L'$, 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 $\text{JSL}_f$-morphisms $p, q$ and $\tau_a$ $(a \in \Sigma)$ making the left diagram below commute.
   c. There exists a $\text{Dep}$-object $S \subseteq S \times S$ with $|S| \leq k$ and $|S| \leq |B|$ and $\text{Dep}$-morphisms $P, Q$ and $T_a$ $(a \in \Sigma)$ making the right diagram below commute (cf. Example 3.6/3.7).

\[
\begin{array}{ccc}
B(L) & \xrightarrow{\delta_a} & B(L) \\
\uparrow & & \uparrow \\
\downarrow \left\{ \begin{array}{l}
\tau_a \\
p \\
\end{array} \right\} & \quad \quad & \downarrow \left\{ \begin{array}{l}
p \\
\tau_a \\
\end{array} \right\} \\
2 & \xrightarrow{\delta_a} & 2 \\
\end{array}
\]
In fact, (a) ⇔ (b) was shown in Theorem 2.1(1), and (b) ⇔ (c) follows from the equivalence between \( \text{JSL}_\tau \) and \( \text{Dep} \). 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 \( \text{JSL}_\tau \), dualizing \( q \) yields a surjective morphism from \( \text{BLD}(L) \cong \text{BLD}(L)^{op} \) to \( S^{op} \). Thus,\[ |M(S)| = |J(S^{op})| \leq |J(\text{BLD}(L))| = |\Sigma^*/\sim_L| = |\text{LD}(L')| \leq |B|.

In the two last steps, we use that the congruence classes of \( \sim_L \) correspond bijectively to left derivatives of \( L' \) by (2.4), and that \( \text{LD}(L') \) is the set of states of the minimal dfa for \( L' \). By Example 3.6 and 3.7 the upper and lower path of the left diagram in \( \text{JSL}_\tau \) correspond under the equivalence functor \( \text{Pirr} \) to the upper and lower path of the right diagram in \( \text{Dep} \). Therefore, Theorem 3.3 shows the two diagrams to be equivalent.

3. From (a) ⇔ (c) we deduce that the relations \( S, P, Q \) and \( T_a \) \( (a \in \Sigma) \) 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 \( L' \), using that \( \Sigma^*/\sim_L \cong \text{LD}(L') \).

\[ \textbf{Remark 5.4.} \text{ An alternative proof that } \text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}} \text{ is in NP uses the following characterization of atomic dfas. Given an nfa } N, \text{ let } \text{rsc}(N') \text{ denote the dfa obtained by determinizing the reverse nfa } N' \text{ via the subset construction and restricting to its reachable part. Then } N \text{ is atomic iff } \text{rsc}(N') \text{ is a minimal dfa [4, Corollary 2]. Thus, given a pair } (A, B) \text{ of mutually reversed dfas, to decide whether } \text{natm}(L(A)) \leq k \text{ one may guess a } k\text{-state nfa } N \text{ and verify that } \text{rsc}(N') \text{ is a minimal dfa equivalent to } B. \text{ One advantage of our above categorical argument is that it yields simple certificates in the form of } \text{Dep}-\text{morphisms subject to certain commutative diagrams, which amount to solutions of equations in } \text{Rel}. \text{ 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 dfas rather than just deciding their existence: every certificate } S, P, Q, T_a \text{ } (a \in \Sigma) \text{ yields an atomic nfa with states } S_a, \text{ transitions given by } (T_a)_- \subseteq S_a \times S_a \text{ for } a \in \Sigma, \text{ initial states } (I \downarrow P)_- \subseteq S_a \text{ and final states } (Q \downarrow F)_- \subseteq S_a. \text{ (Recall that } \downarrow \text{ denotes composition in } \text{Dep} \text{ and } (-)_- \text{ denotes the maximum lower witness of a } \text{Dep}-\text{morphism, see Remark 3.2.) In fact, this precisely is the nfa of join-irreducibles of the } \text{JSL}-\text{dfa } (S, \tau, p \circ i, f' \circ q) \text{ induced by the left diagram in (5.1). Analogous reasoning also applies to the computation of small subatomic dfas treated in Theorem 5.7 below.} \]

\[ \textbf{Proposition 5.5.} \text{ The problem } \text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{atm}} \text{ is NP-hard.} \]

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

For any \( (R, k) \), let \( S = \text{Open}(R) \) be the finite semilattice of open sets corresponding to the \( \text{Dep} \)-object \( R \), cf. Theorem 3.3, and let \( L = L(S) \) be its lattice language. We claim that the desired reduction is given by \[ (R, k) \mapsto (\text{dfa}(L), \text{dfa}(L'), k), \]
where \( \text{dfa}(L) \) and \( \text{dfa}(L') \) are the minimal dfas for \( L \) and \( L' \). Thus, we need to prove that (a) \( \dim(\mathcal{R}) = \text{natm}(L) \), and (b) the two dfas can be computed in polynomial time from \( \mathcal{R} \).

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

\[
\mathcal{R} \xrightarrow{\text{Thm 3.3}} \text{Pirr}(\text{Open}(\mathcal{R})) = \text{Pirr}(\mathcal{S}) \xrightarrow{\text{Lem 4.4}} \text{Pirr}(\text{SLD}(\mathcal{L}(\mathcal{S}))) = \text{Pirr}(\text{SLD}(\mathcal{L})) \xrightarrow{\text{Ex 3.6}} \text{DR}_L.
\]

Lemma 3.2(2) and Proposition 4.5 then imply \( \dim(\mathcal{R}) = \dim(\text{DR}_L) = \text{natm}(L) \).

Ad (b). Let \( J(\text{Open}(\mathcal{R})) = \{j_1, \ldots, j_n\} \) and \( M(\text{Open}(\mathcal{R})) = \{m_1, \ldots, m_p\} \). Then \( \text{dfa}(L) \) and \( \text{dfa}(L') \) are the automata depicted below, where \( L \) and \( L' \) 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:

\textbf{Definition 5.6.} A monoid recognizer is a triple \((M, h, F)\) of a finite monoid \( M \), a map \( h: \Sigma \to M \) and a subset \( F \subseteq M \). The language recognized by \((M, h, F)\) is given by \( L(M, h, F) = h^{-1}[F] \), where \( h: \Sigma^* \to M \) 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 \((\text{syn}(L), \mu_L, F_L)\) where \( \mu_L: \Sigma \to \text{syn}(L) \) is the domain restriction of the syntactic morphism and \( F_L = \{|w|_{\text{syn}} : w \in L\} \). It satisfies \( |\text{syn}(L)| \leq |M| \) for every recognizer \((M, h, F)\) of \( L \). Consider the following decision problem:

\textbf{MON} \to \text{NFA}_{\text{syn}}

\textbf{Input:} A monoid recognizer \((M, h, F)\) and a natural number \( k \).

\textbf{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.

\textbf{Theorem 5.7.} The problem MON \to NFA_{\text{syn}} is NP-complete.

\textbf{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 \( \text{syn}(L) \) in Theorem 2.1(2) and translates the ensuing JSL\textsubscript{\text{f}}-diagram into Dep. To show NP-hardness, one reduces from BICLIQUE COVER via

\[
(\mathcal{R}, k) \mapsto (\text{syn}(L), \mu_L, F_L, k),
\]

where again \( L = L(\text{Open}(\mathcal{R})) \).

Our complexity results indicate a trade-off, i.e. computing small subatomic nfas requires a less succinct representation of the input language. Generally, \(|\text{dfa}(L)|, |\text{dfa}(L')| \leq |\text{syn}(L)|\) and the syntactic monoid can be far larger – even for nuclear languages.
We conclude this paper by outlining some useful consequences of our
As shown above, nuclear languages form a natural common generalization of bideterministic,
6.2 Unary languages
For unary regular languages $L \subseteq \{a\}^*$, every two-sided derivative $(a^*)^{-1}L(a^*)^{-1}$ is equal to
6 Applications
We finally further justify the inputs of DFA + DFA' $\rightarrow$ NFA_{atm} and MON $\rightarrow$ NFA_{syn}; the two modified problems DFA $\rightarrow$ NFA_{atm} and DFA $\rightarrow$ NFA_{syn} where only a (single) dfa is given are computationally much harder.

$\triangleright$ Theorem 6.1. For nuclear languages, the problem DFA + DFA' $\rightarrow$ NFA is NP-complete.
In fact, by Proposition 4.5(1) we have $\text{ns}(L) = \dim(D_L)$ 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.

$\triangleright$ 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 $\delta_n(i) = i + 1 \mod n$ for $i = 0, \ldots, n-1$, and $\delta_n(0) = 1, \delta_n(1) = 0, \delta(0) = 1, \delta(1) = 0$ 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, $|\text{dfa}(L_n)| = |\text{dfa}(L_n^r)| = n$.
2. We have $|\text{syn}(L_n)| = n!$. To see this, recall that $\text{syn}(L_n)$ is the transition monoid of $A_n \cong \text{dfa}(L_n)$. It is generated by the n-cycle $\delta_n = (0 \ 1 \ \cdots \ n-1)$ and the transposition $\delta_n(0) = (0 \ 1)$; 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 SLD($L_n$) is a boolean algebra. In particular, $L_n$ is a nuclear language.
We finally further justify the inputs of DFA + DFA' $\rightarrow$ NFA_{atm} and MON $\rightarrow$ NFA_{syn}; the two modified problems DFA $\rightarrow$ NFA_{atm} and DFA $\rightarrow$ NFA_{syn} where only a (single) dfa is given are computationally much harder.

$\triangleright$ Theorem 5.9. DFA $\rightarrow$ NFA_{atm} and DFA $\rightarrow$ NFA_{syn} are PSPACE-complete.
Proof. This follows by inspecting Jiang and Ravikumar’s [15] argument that DFA $\rightarrow$ 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 $\Sigma$ satisfies $\bigcup_i L(A_i) = \Sigma^*$. For any $A_1, \ldots, A_n$ they construct a dfa $A$ over some alphabet $\Gamma$ and a natural number $k$ such that:
1. If $\bigcup_i L(A_i) \neq \Sigma^*$, then every dfa accepting $L(A)$ requires at least $k + 1$ states.
2. If $\bigcup_i L(A_i) = \Sigma^*$, then there exists an dfa accepting $L(A)$ with $k$ states.
In the proof of (2), an explicit k-state dfa $N = (Q, \delta, \{q_0\}, F)$ with $L(N) = L(A)$ is given, see [15, Fig. 1]. It has the property that, after $\varepsilon$-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 dfa [7]. In particular, $N$ is both atomic and subatomic. Consequently, $(A_1, \ldots, A_n) \rightarrow (A, k)$ is also a reduction to both DFA $\rightarrow$ NFA_{atm} and DFA $\rightarrow$ NFA_{syn}.  

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

6.1 Nuclear Languages
As above, nuclear languages form a natural common generalization of bideterministic, biRFA, and lattice languages. Let DFA + DFA' $\rightarrow$ NFA be the variant of DFA + DFA' $\rightarrow$ NFA_{atm} where the target dfas are arbitrary, i.e. the task is to decide $\text{ns}(L(A)) \leq k$. Then:

$\triangleright$ Theorem 6.1. For nuclear languages, the problem DFA + DFA' $\rightarrow$ NFA is NP-complete.
In fact, by Proposition 4.5(1) we have $\text{ns}(L) = \dim(D_L)$ 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 $(a^*)^{-1}L(a^*)^{-1}$ is equal to

$\boldsymbol{MFCS \ 2021}$
Theorem 6.2. For unary languages, the problem \( \text{DFA} \rightarrow \text{NFA}_{\text{syn}} \) is in NP.

This theorem generalizes the best-known complexity result for unary nfas, which asserts that the problem \( \text{DFA} \rightarrow \text{NFA} \) 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 \( \text{nsyn}(L) = n(L) \), hence \( \text{DFA} \rightarrow \text{NFA} \) coincides with \( \text{DFA} \rightarrow \text{NFA}_{\text{syn}} \).

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 \( \text{nsyn}(L) = \text{natm}(L) \).

Therefore, Theorem 5.2 implies

Theorem 6.4. For group languages, \( \text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{syn}} \) is in NP.

The complexity of the general \( \text{DFA} + \text{DFA}' \rightarrow \text{NFA}_{\text{syn}} \) 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 \( \text{DFA} + \text{DFA}' \rightarrow \text{NFA} \) and \( \text{MON} \rightarrow \text{NFA} \) 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 \( \text{DR}_L \). Since our base equivalence \( \text{JSL}_f \cong \text{Dep} \) 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

78:16  Syntactic Minimization of Nondeterministic Finite Automata