Nominal Automata with Name Binding

Lutz Schröder[1], Dexter Kozen[2], Stefan Milius[1], and Thorsten Wißmann[1]

1 **Friedrich-Alexander-Universität Erlangen-Nürnberg**
2 **Cornell University**

Abstract

Automata models for data languages (i.e. languages over infinite alphabets) often feature either global or local freshness operators. We show that Bollig et al.'s *session automata*, which focus on global freshness, are equivalent to *regular nondeterministic nominal automata (RNNA)*, a natural nominal automaton model with explicit name binding that has appeared implicitly in the semantics of *nominal Kleene algebra (NKA)*, an extension of Kleene algebra with name binding. The expected Kleene theorem for NKA is known to fail in one direction, i.e. there are nominal languages that can be accepted by an RNNA but are not definable in NKA; via session automata, we obtain a full Kleene theorem for RNNAs for an expression language that extends NKA with unscoped name binding. Based on the equivalence with RNNAs, we then slightly rephrase the known equivalence checking algorithm for session automata. Reinterpreting the data language semantics of name binding by unrestricted instead of clean α-equivalence, we obtain a *local* freshness semantics as a quotient of the global freshness semantics. Under local freshness semantics, RNNAs turn out to be equivalent to a natural subclass of Bojanczyk et al.'s *nondeterministic orbit-finite automata*. We establish decidability of inclusion under local freshness by modifying the RNNA-based algorithm; in summary, we obtain a formalism for local freshness in data languages that is reasonably expressive and has a decidable inclusion problem.

1998 ACM Subject Classification F.1.1 Models of Computation, F.4.3 Formal Languages

Keywords and phrases Nominal sets, automata, data languages, name binding

1 Introduction

Data languages are languages over infinite alphabets, regarded as modeling the communication of values from infinite data types such as nonces [17], channel names [11], process identifiers [4], URL's [1], or data values in XML documents (see [21] for a summary). There is by now a plethora of automata models for data languages, which can be classified along several axes. One line of division in terms of presentation is between models that use explicit registers, thus have a finite-state description (generating infinite configuration spaces) on the one hand, and more abstract models phrased as automata over nominal sets [22] on the other hand. The latter have infinitely many states but are typically required to be *orbit-finite*, i.e. there are only finitely many distinct states up to renaming implicitly stored letters. There are known correspondences between the two styles; e.g. Bojanczyk et al.'s *nondeterministic orbit-finite automata (NOFA)* [3] are equivalent to Kaminski and Francez' *finite memory automata (FMA)* [12] (also called register automata), or more precisely to their extension with nondeterministic reassignment [14]. A second distinction concerns the semantics of constructs for reading "fresh" names: *global freshness* requires that the next letter to be consumed has not been seen before, while *local freshness* postulates only that the next letter is distinct from the (boundedly many) letters currently stored in the registers.

Although local freshness looks computationally more natural, nondeterministic automata models (typically more expressive than deterministic ones [15]) featuring local freshness tend to have undecidable inclusion problems. This includes FMAs and NOFAs [21, 3] (unless the putatively larger FMA has at most two registers [12]) as well as *variable automata* [10]. *Finite-state unification-based automata (FSUBAs)* [13] have a decidable inclusion problem but do not support freshness other than in the sense of distinctness from a fixed finite set Θ of *read-only* symbols. Contrastingly, *session automata*, which give up local freshness in favor of global freshness, have a decidable inclusion problem [4].

Another formalism for global freshness is *nominal Kleene algebra (NKA)* [8]. It has been shown that a slight variant of the original NKA semantics satisfies one half of a Kleene theorem [15], which states that NKA expressions can be converted into a species of nondeterministic nominal automata with explicit *name binding* transitions (the exact definition of these automata being left implicit in op. cit.); the converse direction of the Kleene theorem fails even for deterministic nominal automata.

In the present work, we give an explicit definition of a nondeterministic nominal automaton model with name binding that we call *regular nondeterministic nominal automata (RNNA)*. The key point in the definition is to impose finite branching modulo α-equivalence of transitions: a state in an RNNA with (orbit-finite) state set Q has a finite set of transitions that are either *free*, i.e. elements of $\mathbb{A} \times Q$ where $\mathbb{A}$ is the infinite alphabet of *names*, or *bound*, i.e. elements $\langle a \rangle q \in [\mathbb{A}]Q$ where $[\mathbb{A}](-)$ denotes the *abstraction* functor [22] and $\langle a \rangle q$ is read as "bind the name a in q," taken modulo α-equivalence. We show first that RNNAs are equivalent to a mild generalization of session automata that we call *nondeterministic finite bar automata* (bar NFAs). Immediate consequences are

- a full Kleene theorem for RNNAs and a language of regular expressions with unscoped name binding called *regular bar expressions*;
- a translation of NKA into regular bar expressions, hence, for closed expressions, into session automata;
- decidability in parametrized PSPACE of inclusion for RNNAs, implying the known EXPSPACE decidability result for NKA [15].

We then go on to modify the semantics of RNNAs: as for NKA [15], their semantics is most naturally given in terms of strings with name binding, which can be converted into an essentially equivalent data language by α-renaming bound variables in all possible ways to be mutually distinct, then removing all binders. By giving up the distinctness requirement, which enforces global freshness, we obtain a semantics that is essentially a restricted form of *local* freshness: Following the usual rules of α-equivalence, a bound name can now stand for any name except those previously bound names that still appear later in the word. We thus obtain a local freshness semantics as a *quotient* of global freshness.

We show that under local freshness, RNNAs correspond to a natural subclass of NOFAs (equivalently, FMAs) defined by excluding nondeterministic reassignment and by enforcing a policy of *name dropping*, which in terms of registers can be phrased as "the automaton may keep a letter in a register only if that letter is going to be used later" (much like teaching your five-year-old not to monopolize toys that she does not actually want to play with). For example, one cannot accept the language $\{ab \mid a \neq b\}$ but one *can* accept $\{aba \mid a \neq b\}$. Unsurprisingly, RNNAs with local freshness semantics are strictly more expressive than FSUBAs (with empty Θ); the relationships of the various models are summarized in Figure 1. We show that RNNAs nevertheless retain a decidable inclusion problem, again in parametrized PSPACE, using an algorithm that we obtain by varying the one for global freshness. We are not aware of any other nondeterministic automata model for local freshness (with more than two registers) that has a decidable inclusion problem.

Further Related Work A Kleene theorem for *deterministic* nominal automata and an expression language with explicit recursion is mentioned in the conclusion of [15]. Kurz et al. [18] introduce regular expressions for languages over words with explicit scoped binding, which differ technically from those used in the semantics of NKA and regular bar expressions in that they are taken only modulo α-equivalence, not the other equations of NKA concerning the extension of the scope of binders. They satisfy a Kleene theorem for a species of automata that incorporate an explicit bound on the depth of nesting of bindings, rejecting words that exceed this depth.

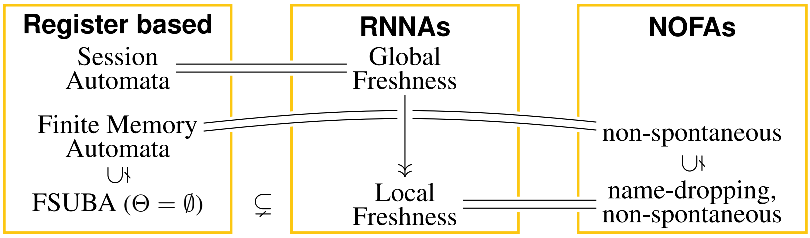


Figure 1 Expressivity of selected data language formalisms (restricted to empty initial register assignment)

Surveys on automata for data languages can be found in [2, 10, 23]. Data languages are often represented as products of a classical finite alphabet and an infinite alphabet; for simplicity, we use just the set of names as the alphabet (as for example in [21]). Our unscoped name binding construct is, under local semantics, similar to the binders in *regular expressions with memory* as introduced by Libkin et al., who also observe that while such binders have a more imperative than declarative flavor, they are necessary to obtain equivalence results with automata (in this case, register automata) [19].

2 Preliminaries

G-sets Recall that a *group action* of a group G on a set X is a map $G \times X \to X$, denoted by juxtaposition or infix $\cdot$, such that $\pi(\rho x) = (\pi\rho)x$ and $1x = x$ for $\pi, \rho \in G, x \in X$. A *G-set* is a set X equipped with an action of G. The *orbit* of $x \in X$ is the set $\{\pi x \mid \pi \in G\}$. A function $f : X \to Y$ between G-sets X, Y is *equivariant* if $f(\pi x) = \pi(f x)$ for all $\pi \in G, x \in X$. Given a G-set X, G acts on subsets $A \subseteq X$ by $\pi A = \{\pi x \mid x \in A\}$. For $A \subseteq X$ and $x \in X$, we put

$$\mathsf{fix}\, x = \{\pi \in G \mid \pi x = x\} \qquad \text{and} \qquad \mathsf{Fix}\, A = \textstyle\bigcap_{x \in A} \mathsf{fix}\, x.$$

Note that elements of $\mathsf{fix}\, A$ and $\mathsf{Fix}\, A$ fix A setwise and pointwise, respectively.

Nominal sets Fix a countably infinite set $\mathbb{A}$ of *names*. We fix G to be the group of finite permutations on $\mathbb{A}$. Putting $\pi a = \pi(a)$ makes $\mathbb{A}$ into a G-set. Given a G-set X and $x \in X$, a set $A \subseteq \mathbb{A}$ *supports* x if $\mathsf{Fix}\, A \subseteq \mathsf{fix}\, x$, and $x \in X$ *has finite support* if some finite $A \subseteq \mathbb{A}$ supports x. In this case, there is a smallest set supporting x, denoted $\mathsf{supp}(x)$. For $a \in \mathbb{A}$, we say that a is *fresh for* x and write $a \# x$ if $a \notin \mathsf{supp}(x)$. A *nominal set* is a G-set all whose elements have finite support. For every equivariant function f between nominal sets, we have $\mathsf{supp}(fx) \subseteq \mathsf{supp}(x)$. The function supp itself is equivariant, i.e. $\mathsf{supp}(\pi x) = \pi(\mathsf{supp}(x))$ for $\pi \in G$. It follows that if x_1, x_2 are in the same orbit of a nominal set, then $\sharp\mathsf{supp}(x_1) = \sharp\mathsf{supp}(x_2)$ (we use $\sharp$ for cardinalities, avoiding overuse of '$|$'). A subset $S \subseteq X$ is *finitely supported (fs)* if S has finite support with respect to the above-mentioned action of G on subsets; *equivariant* if $\pi x \in S$ for all $\pi \in G$ and $x \in S$ (which implies $\mathsf{supp}(S) = \emptyset$); and *uniformly finitely supported (ufs)* if $\bigcup_{x \in S} \mathsf{supp}(x)$ is finite [26].

▶ **Lemma 2.1** ([7], Theorem 2.29). *If S is ufs, then* $\mathsf{supp}(S) = \bigcup_{x \in S} \mathsf{supp}(x)$.

For a nominal set X, we denote by $\mathcal{P}_{\mathsf{fs}}(X)$ and $\mathcal{P}_{\mathsf{ufs}}(X)$ the sets of fs and ufs subsets of X, respectively. Note that any ufs set is fs but not conversely; e.g. the set $\mathbb{A}$ is fs but not ufs. Moreoever, any finite subset of X is ufs but not conversely; e.g. the set of words a^n for fixed $a \in \mathbb{A}$ is ufs but not finite. A nominal set X is *orbit-finite* if the action of G on it has only finitely many orbits.

▶ **Lemma 2.2.** *Ever ufs subset of an orbit-finite set X is finite.*

On the category Nom of nominal sets and equivariant maps, we have the *abstraction functor* $[\mathbb{A}](-)$ defined on objects by $[\mathbb{A}]X = (\mathbb{A} \times X)/{\sim}$, where the relation $\sim$ abstracts α-equivalence: $(a, x) \sim (b, y)$ iff $(c\,a) \cdot x = (c\,b) \cdot y$ for any fresh c. We write $\langle a \rangle x$ for the $\sim$-equivalence class of (a, x).

Coalgebra An F-*coalgebra* (C, γ) for an endofunctor $F : \mathbf{C} \to \mathbf{C}$ on a category $\mathbf{C}$ consists of a $\mathbf{C}$-object C of *states* and a morphism $\gamma : C \to FC$; here, we are interested in the case $\mathbf{C} = \mathsf{Nom}$. A *coalgebra morphism* $f : (C, \gamma) \to (D, \delta)$ is a morphism $f : C \to D$ such that $Ff\gamma = \delta f$. An F-coalgebra (C, γ) is *final* if for each F-coalgebra (D, δ), there exists a unique coalgebra morphism $(D, \delta) \to (C, \gamma)$. A *pointed* coalgebra is a coalgebra with a distinguished *initial* state.

For example, F-coalgebras for the functor $FX = \mathbb{A} \times X$ on Nom consist of equivariant maps $X \to \mathbb{A}$ (output) and $X \to X$ (next state), thus produce a stream of names at each state x; equivariance and the finite support of x imply that this stream has finite support, i.e. contains only finitely many distinct names. Consequently, the final F-coalgebra in this case is the set of finitely supported streams over $\mathbb{A}$.

3 Nominal Automata and Global Freshness

We next recall the basic definitions in the theory of nominal automata [3] (developed in op. cit. in the setting of arbitrary symmetries and for general orbit-finite alphabets), and introduce our model of regular nondeterministic nominal automata. *Nondeterministic orbit-finite automata (NOFAs)* are succinctly defined as orbit-finite coalgebras for the functor F on Nom given by

$$FX = 2 \times \mathcal{P}_{\mathsf{fs}}(\mathbb{A} \times X)$$

(where $2 = \{\top, \bot\}$), equipped with an equivariant subset of *initial* states. That is, a NOFA A consists of an orbit-finite set Q of states, an equivariant set $F \subseteq Q$ of *final* states (those that map to $\top$ under the first component of the F-coalgebra structure), and an equivariant transition relation $\to \subseteq Q \times \mathbb{A} \times Q$, where we write $q \xrightarrow{a} p$ for $(q, a, p) \in \to$. We refer to a NOFA whose transition relation is deterministic as a *DOFA*. An $\mathbb{A}$-*language* is a subset of $\mathbb{A}^*$. The $\mathbb{A}$-language $L(A)$ *accepted* by A is defined in the standard way: First, we inductively extend the transition relation to words $w \in \mathbb{A}$ by putting $q \xrightarrow{\epsilon} q$, and $q \xrightarrow{aw} p$ whenever $q \xrightarrow{a} q'$ and $q' \xrightarrow{w} p$ for some state q'. Then, A *accepts* $w \in \mathbb{A}$ if there exist an initial state q and a final state p such that $q \xrightarrow{w} p$, and $L(A) = \{w \mid A \text{ accepts } w\}$.

NOFAs are equivalent to *finite-memory automata (FMA) with nondeterministic reassignment* [3, 14]. These are roughly described as having a finite set of registers in which names from the current word can be stored if they are *locally fresh*, i.e. not currently stored in any register; transitions are labeled with register indices k, meaning that the transition accepts the next letter if it equals the content of register k. In the equivalence with NOFAs, the names currently stored in the registers correspond to the support of states. Summing up, NOFAs are an automata model for local freshness.

So far, transitions of a state in Q are elements of $\mathbb{A} \times Q$. Given the central role that the abstraction functor $[\mathbb{A}](-)$ (Section 2) plays in nominal sets, it is natural to extend the model to allow *bound* transitions also, i.e. elements of $[\mathbb{A}]Q$, and indeed this is what happens in automata models for nominal Kleene algebra [15]. We can combine this extension with a restriction on branching: while it does not make much sense to restrict a NOFA to be finitely branching (this would imply that any initial state could accept only words consisting of the names in its support, i.e. such a NOFA could never read fresh names), it will turn out that finite branching is technically convenient and still retains a reasonable level of expressivity in the presence of bound transitions.

▶ **Definition 3.1.** A *regular nondeterministic nominal automaton (RNNA)* is a pointed orbit-finite coalgebra $A = (Q, \xi : Q \to NQ)$ for the functor N on Nom given by

$$NX = 2 \times \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times X) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]X).$$

The *degree* $\mathsf{deg}(A) = \max\{\sharp\mathsf{supp}(q) \mid q \in Q\}$ of A is the maximum size of supports of states in A.

The functor N is a nondeterministic variant of the functor $KX = 2 \times X^{\mathbb{A}} \times [\mathbb{A}]X$ whose coalgebras are *deterministic nominal automata* [15]. Explicitly, an RNNA can be described as a tuple $A = (Q, \to, s, F)$ consisting of

- an orbit-finite set Q of states;
- an equivariant subset $\to$ of $Q \times \bar{\mathbb{A}} \times Q$, the *transition relation*, where $\bar{\mathbb{A}} = \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$ and we write $q \xrightarrow{\alpha} q'$ for $(q, \alpha, q') \in \to$; transitions of type $q \xrightarrow{a} q'$ are called *free*, and those of type $q \xrightarrow{|a} q'$ *bound*;
- an equivariant subset $F \subseteq Q$ of *final* states; and
- an *initial* state $s \in Q$.

These data are required to satisfy the following conditions:

- The relation $\to$ is α-*invariant*, i.e. closed under α-equivalence of transitions, where transitions $q \xrightarrow{|a} q'$ and $p \xrightarrow{|b} p'$ are α-*equivalent* if $q = p$ and $\langle a\rangle q' = \langle b\rangle p'$.
- The relation $\to$ is *finitely branching up to α-equivalence*, i.e. for each state q the sets $\{(a, q') \mid q \xrightarrow{a} q'\}$ and $\{\langle a\rangle q' \mid q \xrightarrow{|a} q'\}$ are finite (equivalently ufs, by Lemma 2.2).

We proceed to define the language semantics of RNNAs.

▶ **Definition 3.2.** A *bar string* is a word over $\bar{\mathbb{A}}$, i.e. an element of $\bar{\mathbb{A}}^*$. The set $\bar{\mathbb{A}}^*$ is made into a nominal set by the letter-wise action of G. The *free names* occurring in a bar string w are those names a that occur in w to the left of any occurrence of $|a$. We write $\mathsf{FN}(w)$ for the set of free names of w, and say that w is *closed* if $\mathsf{FN}(w) = \emptyset$. We define α-*equivalence* $\equiv_\alpha$ on bar strings as the equivalence (*not*: congruence) generated by $w|av \equiv_\alpha w|bu$ if $\langle a\rangle w = \langle b\rangle u$ (for $w, v, w \in \bar{\mathbb{A}}^*$). We write $[w]_\alpha$ for the α-equivalence class of w. The set $\mathsf{FN}(w)$ is clearly invariant under α-equivalence, so we have a well-defined notion of free names of bar strings modulo $\equiv_\alpha$. We say that a bar string is *clean* if its bound variables are mutually distinct and distinct from all its free variables. Clearly, every bar string is α-equivalent to a clean one. For a bar string w, we denote by $\mathsf{ub}(w) \in \mathbb{A}^*$ (for *unbind*) the word arising from w by replacing all bound names $|a$ with the corresponding free name a.

A *literal language* is a set of bar strings, and a *bar language* is an fs set of bar strings modulo α-equivalence, i.e. an fs subset of

$$\bar{M} := \bar{\mathbb{A}}^*/\equiv_\alpha.$$

An RNNA A, with data as above, *(literally) accepts* a bar string $w \in \bar{\mathbb{A}}^*$ if $s \xrightarrow{w} q$ for some $q \in F$, where we extend the transition notation $\xrightarrow{w}$ to bar strings in the usual way. The *literal language accepted by* A is the set $L_0(A)$ of bar strings accepted by A, and the *bar language accepted by* A is the quotient $L_\alpha(A)$ of $L_0(A)$ modulo α-equivalence.

▶ **Remark 3.3.** In *dynamic sequences* [9], there are two dynamically scoped constructs $\langle a$ and $a\rangle$ for dynamic allocation and deallocation, respectively, of a name a; in this notation, our $|a$ corresponds to $\langle aa$. As we discuss later in this section, the bar language model is isomorphic to the ν-*string*-based model of NKA [15]. In particular, the bar languages form the final coalgebra for the endofunctor $KX = 2 \times X^{\mathbb{A}} \times [\mathbb{A}]X$ on Nom for deterministic nominal automata mentioned before, with free and bound transitions understood in the same way as for RNNA. (There is however an expressivity gap between deterministic and nondeterministic nominal automata [15, Example 4.13]). The ν-string-based model is equivalent to an *alternative language mode* AL [16], which essentially implements *global freshness*. That is, AL is defined on closed expressions in terms of $\mathbb{A}$-languages, with bound names required to be globally fresh, i.e. not previously seen in the current word. Formally, AL is given by applying to a bar language L the operator N given by

$$N(L) = \{\mathsf{ub}(w) \mid w \text{ clean}, [w]_\alpha \in L\} \subseteq \mathbb{A}^*.$$

Summing up, under bar language semantics, RNNAs are a formalism for global freshness, so we also refer to bar language semantics as *global freshness semantics*. Since RNNAs will turn out to be essentially equivalent to session automata under this semantics, we defer examples to Section 4.

A key property of RNNAs is that supports of states evolve in the expected way along transitions (cf. [15, Lemma 4.6] for the deterministic case):

▶ **Lemma 3.4.** *Let A be an RNNA. Then the following hold.*
1. *If $q \xrightarrow{a} q'$ in A then $\mathsf{supp}(q') \cup \{a\} \subseteq \mathsf{supp}(q)$.*
2. *If $q \xrightarrow{|a} q'$ in A then $\mathsf{supp}(q') \subseteq \mathsf{supp}(q) \cup \{a\}$.*

In fact, the properties in the lemma are clearly also sufficient for ufs branching. From Lemma 3.4, an easy induction shows that for any state q in an RNNA and any $w \in L_0(q)$, we have $\mathsf{FN}(w) \subseteq \mathsf{supp}(q)$. Furthermore, we immediately have

▶ **Corollary 3.5.** *Let A be a RNNA. Then $L_\alpha(A)$ is ufs; specifically, if s is the initial state of A and $w \in L_\alpha(A)$, then $\mathsf{supp}(w) \subseteq \mathsf{supp}(s)$.*

We have an evident notion of α-equivalence of paths in RNNAs, defined analogously as for bar strings (see Remark A.5 in the appendix). Of course, α-equivalent paths always start in the same state. The set of paths of an RNNA A is closed under α-equivalence (see Lemma A.15 in the appendix). However, this does not in general imply that $L_0(A)$ is closed under α-equivalence; e.g. for A being

$$s() \xrightarrow{|a} t(a) \xrightarrow{|b} u(a,b) \tag{1}$$

(with a, b ranging over distinct names in $\mathbb{A}$), where $s()$ is initial and the states $u(-,-)$ are final, we have $|a|b \in L_0(A)$ but the α-equivalent $|a|a$ is not in $L_0(A)$. Crucially, assuming closure of $L_0(A)$ under α-equivalence is nevertheless without loss of generality, as we show next.

▶ **Definition 3.6.** An RNNA A is *name-dropping* if for every state q in A and every subset $N \subseteq \mathsf{supp}(q)$ there exists a state $q|_N$ in A that *restricts q to N*; that is, $\mathsf{supp}(q|_N) = N$, $q|_N$ is final if q is final, and $q|_N$ has at least the same incoming transitions as q (i.e. for all states p in A, if $p \xrightarrow{a} q$ then $p \xrightarrow{a} q|_N$ and if $p \xrightarrow{|a} q$ then $p \xrightarrow{|a} q|_N$), and as many of the outgoing transitions of q as possible; i.e. $q|_N \xrightarrow{a} q'$ whenever $q \xrightarrow{a} q'$ and $\mathsf{supp}(q') \cup \{a\} \subseteq N$, and $q|_N \xrightarrow{|a} q'$ whenever $q \xrightarrow{|a} q'$ and $\mathsf{supp}(q') \subseteq N \cup \{a\}$.

The counterexample shown in (1) fails to be name-dropping, as no state restricts $q = u(a,b)$ to $N = \{b\}$.

The following lemma shows that closure under α-equivalence is restored under name-dropping:

▶ **Lemma 3.7.** *Let A be a name-dropping RNNA. Then $L_0(A)$ is closed under α-equivalence, i.e. $L_0(A) = \{w \mid [w]_\alpha \in L_\alpha(A)\}$.*

Finally, we can close a given RNNA under name dropping, preserving the bar language:

▶ **Lemma 3.8.** *Given an RNNA of degree k with n orbits, there exists a bar language-equivalent name-dropping RNNA of degree k with at most $n2^k$ orbits.*

Proof (Sketch). From an RNNA A, construct a name-dropping RNNA with states of the form

$$q|_N := \mathsf{Fix}(N)q$$

where q is a state in A, $N \subseteq \mathsf{supp}(q)$, and $\mathsf{Fix}(N)q$ denotes the orbit of q under $\mathsf{Fix}(N)$. The final states are the $q|_N$ with q final in A, and the initial state is $s|_{\mathsf{supp}(s)}$, where s is the initial state of A. As transitions, we take

- $q|_N \xrightarrow{a} q'|_{N'}$ whenever $q \xrightarrow{a} q'$, $N' \subseteq N$, and $a \in N$, and
- $q|_N \xrightarrow{|a} q'|_{N'}$ whenever $q \xrightarrow{|b} q''$, $N'' \subseteq \mathsf{supp}(q'') \cap (N \cup \{b\})$, and $\langle a\rangle(q'|_{N'}) = \langle b\rangle(q''|_{N''})$.

One can show that this yields a name-dropping RNNA that is equivalent to A. ◀

▶ **Example 3.9.** Closing the RNNA from (1) under name dropping as per Lemma 3.8 yields additional states that we may denote $u(\bot, b)$ (among others), with transitions $t(a) \xrightarrow{|b} u(\bot, b)$; now, $\langle b\rangle u(\bot, b) = \langle a\rangle u(\bot, a)$, so $|a a$ is accepted.

Relation to Nominal Kleene Algebra We recall that expressions r, s of nominal Kleene algebra (NKA) [8], briefly *NKA expressions*, are defined by the grammar

$$r, s ::= 0 \mid 1 \mid a \mid r + s \mid rs \mid r^* \mid \nu a.\, r \qquad (a \in \mathbb{A}).$$

Kozen et al. [16, 15] give a semantics of NKA in terms of languages over words with binding, so called *ν-strings*, which are either 1 or ν-regular expressions formed using only names $a \in \mathbb{A}$, sequential composition, and name binding ν, taken modulo the laws of NKA [8], including α-equivalence and laws for scope extension of binding. It is easy to see that the nominal set of ν-strings modulo these laws is isomorphic to $\bar{M}$; one converts bar strings into ν-strings by replacing any occurrence of $|a$ with $\nu a.a$, with the scope of the binder extending to the end of the string. In this semantics, a binder νa is just interpreted as itself, and all other clauses are standard. Kozen et al. show that on closed expressions, their semantics is equivalent to the one originally defined by Gabbay and Ciancia [8].

▶ **Remark 3.10.** On open expressions, the semantics of [8] and [15, 16] differ. For purposes of expressivity comparisons, we will generally restrict to closed expressions as well as "closed" automata and languages in the sequel. For automata, this typically amounts to the initial register assignment being empty, and for languages to being equivariant.

It has been shown by Kozen et al. [15] that a given NKA expression r can be translated into a nondeterministic nominal automaton whose states are the so-called *spines* of r, which amounts to one direction of a Kleene theorem. One can show that the spines in fact form an RNNA. The other direction of the Kleene theorem is known to fail even for orbit-finite deterministic nominal automata, i.e. RNNAs are strictly more expressive than NKA:

▶ **Example 3.11.** [15] It is easy to construct an orbit-finite deterministic nominal automaton (or an RNNA, shown explicitly in Example 4.6) accepting the ν-language

$$\{\epsilon, \nu b.ba, \nu b.ba(\nu a.ab), \nu b.ba(\nu a.ab(\nu b.ba)), \nu b.ba(\nu a.ab(\nu b.ba(\nu a.ab))), \dots\},$$

which requires unbounded nesting depth of ν, hence cannot be defined in NKA.

4 Nondeterministic Finite Bar Automata

We next provide a finite representation of RNNAs by proving their equivalence with ordinary nondeterministic finite automata (NFAs) over $\bar{\mathbb{A}}$. These are a mild generalization of session automata [4] and are equivalent to the latter on closed languages (session automata accept only closed languages); that is, on closed languages *RNNA under global freshness semantics are equivalent to session automata*.

▶ **Definition 4.1.** A *nondeterministic finite bar automaton*, or *bar NFA* for short, over $\mathbb{A}$ is an NFA A over $\bar{\mathbb{A}}$. We call transitions of type $q \xrightarrow{a} q$ in A *free transitions* and transitions of type $q \xrightarrow{|a} q$ *bound transitions*. The *literal language* $L_0(A)$ of A is the language accepted by A qua NFA over $\bar{\mathbb{A}}$. The *bar language* $L_\alpha(A) \subseteq \bar{M}$ accepted by A is defined as

$$L_\alpha(A) = L_0(A)/\!\equiv_\alpha.$$

Generally, we denote by $L_0(q)$ the $\bar{\mathbb{A}}$-language accepted by the state q in A and by $L_\alpha(q)$ the quotient of $L_0(q)$ by α-equivalence. The *degree* $\mathsf{deg}(A)$ of A is the number of names $a \in \mathbb{A}$ that occur in transitions $q \xrightarrow{a} q'$ or $q \xrightarrow{|a} q'$ in A.

Similarly, a *regular bar expression* is a regular expression r over $\bar{\mathbb{A}}$; the *literal language* $L_0(r) \subseteq \bar{\mathbb{A}}^*$ *defined by* r is the language defined by r qua regular expression, and the *bar language defined by* r is $L_\alpha(r) = L_0(r)/\equiv_\alpha$. The *degree* $\mathsf{deg}(r)$ *of* r is the number of free or bound names occurring in r.

▶ **Remark 4.2.** Disregarding an additional finite component of the alphabet, a *session automaton* [4] is essentially a bar NFA (where free names a are denoted as $a^\uparrow$, and bound names $|a$ as $a^{\circledast}$). It defines an $\mathbb{A}$-language and interprets bound transitions for $|a$ as binding a to some globally fresh name. In the light of the equivalence of global freshness semantics and bar language semantics as discussed in Section 3, session automata are thus essentially the same as bar NFAs; again, the only difference concerns the treatment of open bar strings: While session automata explicitly reject bar strings that fail to be closed (*well-formed* [4]), a bar NFA will happily accept open bar strings. Part of the motivation for this permissiveness is that we now do not need to insist on regular bar expressions to be closed; in particular, regular bar expressions are closed under subexpressions. Moreover, standard regular expressions over $\mathbb{A}$ are now (open) regular bar expressions.

▶ **Example 4.3.** Phrased in terms of $\mathbb{A}$-languages, bar NFAs, being equivalent to session automata, can express the language “all letters are distinct” but not the universal language $\mathbb{A}^*$.

▶ **Construction 4.4.** We construct an RNNA $\bar{A}$ from a given bar NFA A with set Q of states. For brevity, we already incorporate closure under name dropping as per Lemma 3.8. For a state $q \in Q$, we put $N_q = \mathsf{supp}(L_\alpha(q))$. The set $\bar{Q}$ of states of $\bar{A}$ consists of pairs

$$(q, \pi F_N) \qquad (q \in Q, N \subseteq N_q)$$

where F_N abbreviates $\mathsf{Fix}(N)$ and πF_N denotes a left coset. Note that left cosets for F_N can be identified with injective renamings $N \to \mathbb{A}$; intuitively, $(q, \pi F_N)$ restricts q to N and renames N according to π. (A slightly similar construction with explicit *total* injections has been used to convert history-dependent automata into NOFAs [6]). We let G act on states by $\pi_1 \cdot (q, \pi_2 F_N) = (q, \pi_1\pi_2 F_N)$. The initial state of $\bar{A}$ is (s, F_{N_s}), where s is the initial state of A; a state $(q, \pi F_N)$ is final in $\bar{A}$ iff q is final in A. Free transitions in $\bar{A}$ are given by

$$(q, \pi F_N) \xrightarrow{\pi(a)} (q', \pi F_{N'}) \quad \text{whenever} \quad q \xrightarrow{a} q' \text{ and } N' \cup \{a\} \subseteq N$$

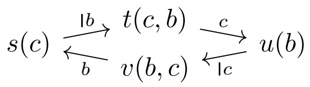
and bound transitions by

$$(q, \pi F_N) \xrightarrow{|a} (q', \pi' F_{N'}) \quad \text{whenever} \quad q \xrightarrow{|b} q', N' \subseteq N \cup \{b\}, \langle a\rangle \pi' F_{N'} = \langle \pi(b)\rangle \pi F_{N'}.$$

▶ **Theorem 4.5.** *$\bar{A}$ is a name-dropping RNNA with at most $|Q|2^{\mathsf{deg}(A)}$ orbits,* $\mathsf{deg}(\bar{A}) = \mathsf{deg}(A)$, *and* $L_\alpha(\bar{A}) = L_\alpha(A)$.

▶ **Example 4.6.** The language from Example 3.11 is equivalent to the bar language $L = \{\epsilon, |ba, |ba|ab, |ba|ab|ba, |ba|ab|ba|ab \dots\}$. Translating the closed bar language $|aL$ equivalently into an $\mathbb{A}$-language under global freshness, we obtain the language of odd-length words in $\mathbb{A}^*$ with identical letters in positions 0 and 2, and with every letter in an odd position being globally fresh and repeated three positions later. The language L is defined by the regular bar expression $(|ba|ab)^*(1 + |ba)$ and accepted by the bar NFA A with four states s, t, u, v, where s is initial and s and u are final, and transitions $s \xrightarrow{|b} t \xrightarrow{a} u \xrightarrow{|a} v \xrightarrow{b} s$. The above construction then produces an

RNNA that is similar to the one shown for this example in [15]: By the above description of left cosets for F_N, we annotate every state q with a list of $\sharp\mathsf{supp}(L_\alpha(q))$ entries that are either (pairwise distinct) names or $\bot$, indicating that the corresponding name from $\mathsf{supp}(L_\alpha(q))$ has been dropped. We can draw those orbits of the resulting RNNA that have the form $(q, \pi N_q)$, i.e. do not drop any names, as



for $b \neq c$, with $s(c)$, $u(b)$ final for all $b, c \in \mathbb{A}$ and $s(c)$ initial.

Additional states and transitions then arise from name dropping; e.g. for t we have additional states $t(\bot, b)$, $t(c, \bot)$, and $t(\bot, \bot)$, all with a $\mathsf{|}b$-transition from $s(c)$. The states $t(\bot, \bot)$ and $t(\bot, b)$ have no outgoing transitions, while $t(c, \bot)$ has a c-transition to $u(\bot)$.

We next present the reverse construction, i.e. given an RNNA A we extract a bar NFA A_0 (a subautomaton of A) such that $L_\alpha(A_0) = L_\alpha(A)$.

Put $k = \mathsf{deg}(A)$. We fix a set $\mathbb{A}_0 \subseteq \mathbb{A}$ of size $\sharp\mathbb{A}_0 = k$ such that $\mathsf{supp}(s) \subseteq \mathbb{A}_0$ for the initial state s of A, and a name $* \in \mathbb{A} - \mathbb{A}_0$. The states of A_0 are those states q in A such that $\mathsf{supp}(q) \subseteq \mathbb{A}_0$. As this implies that the set Q_0 of states in A_0 is ufs, Q_0 is finite by Lemma 2.2. Note that $s \in Q_0$. For $q, q' \in Q_0$, the free transitions $q \xrightarrow{a} q'$ in A_0 are the same as in A (hence have $a \in \mathbb{A}_0$ by Lemma 3.4.1). The bound transitions $q \xrightarrow{\mathsf{|}a} q'$ in A_0 are those bound transitions $q \xrightarrow{\mathsf{|}a} q'$ in A such that $a \in \mathbb{A}_0 \cup \{*\}$. A state is final in A_0 iff it is final in A. The initial state of A_0 is s.

▶ **Theorem 4.7.** *The number of states in the bar NFA A_0 is linear in the number of orbits of A and exponential in* $\mathsf{deg}(A)$. *Moreover,* $\mathsf{deg}(A_0) \leq \mathsf{deg}(A) + 1$, *and* $L_\alpha(A_0) = L_\alpha(A)$.

In combination with the previous construction, we obtain the announced equivalence result:

▶ **Corollary 4.8.** *RNNAs are expressively equivalent to bar NFAs, hence to regular bar expressions.*

This amounts to a Kleene theorem for RNNAs. In combination with the discussion in Section 3, this shows that regular bar expressions are strictly more expressive than NKA. While it might seem that we can now just give up nominal automata and use bar NFAs instead, it turns out that our decision procedure for inclusion (Section 6) will actually use both concepts, essentially running a bar NFA in synchrony with an RNNA.

5 Local Freshness

Recall that the global freshness semantics of RNNA is defined by removing bars from the *clean* representatives of the α-equivalence classes of bar strings in the bar language. Alternatively, we can extract from a bar language L the $\mathbb{A}$-language

$$D(L) = \{\mathsf{ub}(w) \mid [w]_\alpha \in L\}.$$

That is, $D(L)$ is obtained by taking *all* representatives of α-equivalence classes in L and removing all bars. As we show below, RNNAs correspond to a class of NOFAs under the semantics $D(L_\alpha(\cdot))$, which we therefore call the *local freshness* semantics. Note that local freshness is coarser than global freshness; e.g., $L_\alpha(\mathsf{|}a\mathsf{|}b + \mathsf{|}aa) = \{[\mathsf{|}a\mathsf{|}b]_\alpha, [\mathsf{|}aa]_\alpha\} \neq \{[\mathsf{|}a\mathsf{|}b]_\alpha\} = L_\alpha(\mathsf{|}a\mathsf{|}b)$, but $D(L_\alpha(\mathsf{|}a\mathsf{|}b + \mathsf{|}aa)) = \mathbb{A}^2 = D(\mathsf{|}a\mathsf{|}b)$. The semantics $D(L_\alpha(\cdot))$ enforces local freshness by blocking α-renamings of bound names into names that have free occurrences later in the bar string. For example, $\{ab \in \mathbb{A}^2 \mid a \neq b\}$ cannot be accepted by an RNNA under local freshness semantics (e.g. the regular bar expression $\mathsf{|}a\mathsf{|}b$ defines $D(\mathsf{|}a\mathsf{|}b) = \mathbb{A}^2$, as $\mathsf{|}a\mathsf{|}b \equiv_\alpha \mathsf{|}a\mathsf{|}a$). Contrastingly, the language $\{aba \in \mathbb{A}^3 \mid a \neq b\}$ *can* be accepted by an RNNA under local freshness semantics, being defined by the regular bar expression $\mathsf{|}a\mathsf{|}ba$.

▶ **Example 5.1.** Under $D(L_\alpha(\cdot))$, the regular expression $\mathsf{l}a(\mathsf{l}ba\mathsf{l}ab)^*(1+\mathsf{l}ba)$ (Example 4.6) defines the $\mathbb{A}$-language consisting of all odd-length words over $\mathbb{A}$ that contain the same letters in positions 0 and 2 (if any) and repeat every letter in an odd position three positions later (if any) *but no earlier*; that is, the bound names are interpreted as being locally fresh. The reason for this is that, e.g., in the bar string $\mathsf{l}a\mathsf{l}ba\mathsf{l}ab$, α-renaming of the bound name $\mathsf{l}b$ into $\mathsf{l}a$ is blocked by the occurrence of a after $\mathsf{l}b$; similarly, the second occurrence of $\mathsf{l}a$ cannot be renamed into $\mathsf{l}b$.

Relationship to NOFAs To enable a comparison of RNNAs with NOFAs over $\mathbb{A}$ (Section 3), we restrict our attention in the following discussion to RNNAs that are *closed*, i.e. whose initial state has empty support, therefore accept equivariant $\mathbb{A}$-languages. We can convert a closed RNNA A into a NOFA $D(A)$ accepting $D(L_\alpha(A))$ by simply replacing every transition $q \xrightarrow{\mathsf{l}a} q'$ with a transition $q \xrightarrow{a} q'$. We show that the image of this translation is a natural class of NOFAs:

▶ **Definition 5.2.** A NOFA A is *non-spontaneous* if $\mathsf{supp}(s) = \emptyset$ for every initial state s, and

$$\mathsf{supp}(q') \subseteq \mathsf{supp}(q) \cup \{a\} \quad \text{whenever} \quad q \xrightarrow{a} q'.$$

Moreover, A is *α-invariant* if $q \xrightarrow{a} q''$ whenever $q \xrightarrow{b} q'$, $b \# q$, and $\langle a\rangle q'' = \langle b\rangle q'$ (this condition is automatic if $a \# q$). Finally, A is *name-dropping* if for each state q and each set $N \subseteq \mathsf{supp}(q)$ of names, there exists a state $q|_N$ that *restricts q to N*, i.e. $\mathsf{supp}(q|_N) = N$, $q|_N$ is final if q is final, and

- $q|_N$ has at least the same incoming transitions as q;
- whenever $q \xrightarrow{a} q'$, $a \in \mathsf{supp}(q)$, and $\mathsf{supp}(q') \cup \{a\} \subseteq N$, then $q|_N \xrightarrow{a} q'$;
- whenever $q \xrightarrow{a} q'$, $a \# q$, and $\mathsf{supp}(q') \subseteq N \cup \{a\}$, then $q|_N \xrightarrow{a} q'$.

In words, A is non-spontaneous if transitions $q \xrightarrow{a} q'$ in A create no new names other than a in q'.

▶ **Proposition 5.3.** *A NOFA is of the form $D(B)$ for some (name-dropping) RNNA B iff it is (name-dropping and) non-spontaneous and α-invariant.*

▶ **Proposition 5.4.** *For every non-spontaneous and name-dropping NOFA, there is an equivalent non-spontaneous, name-dropping, and α-invariant NOFA.*

In combination with Lemma 3.7, these facts imply

▶ **Corollary 5.5.** *Under local freshness semantics, RNNAs are expressively equivalent to non-spontaneous name-dropping NOFAs.*

▶ **Corollary 5.6.** *The class of languages accepted by RNNAs under local freshness semantics is closed under finite intersections.*

Proof (Sketch). Non-spontaneous name-dropping NOFAs are closed under the standard product construction. ◀

▶ **Remark 5.7.** Non-spontaneity is prevalent in automata models for data languages. Every DOFA is non-spontaneous. Moreover, finite memory automata and register automata are morally non-spontaneous according to their original definitions, i.e. they can read names from the current word into the registers but cannot guess names nondeterministically [12, 21]; the variant of finite memory automata that is proved equivalent to NOFAs in [3] in fact allows such *nondeterministic reassignment* [14]. This makes unrestricted NOFAs strictly more expressive than non-spontaneous ones: the language "the last letter has not been seen before" can be accepted by an unrestricted NOFA (by guessing the last name at the beginning) but not by a non-spontaneous NOFA [12, 27].

Name-dropping restricts expressivity further, with the mentioned language $\{ab \mid a \neq b\}$ being a separating example. In return, it buys decidability of inclusion (Section 6), while for non-spontaneous NOFAs even universality is undecidable [3, 21]. DOFAs are incomparable to RNNAs under local freshness semantics—the language "the last letter has been seen before" is defined by the regular bar expression $(\mathsf{l}b)^*\mathsf{l}a(\mathsf{l}b)^*a$ but not accepted by any DOFA.

Relationship to FSUBAs We now compare RNNAs to *finite-state unification-based automata (FSUBAs)* [13, 25]. A particular feature of FSUBAs is that they distinguish a finite subset Θ of the alphabet that is *read-only*, i.e. cannot be written into the registers. We have no corresponding feature, therefore restrict to $\Theta = \emptyset$ in the following discussion. An FSUBA then consists of finite sets Q and r of states and registers, respectively, a transition relation $\mu \subseteq Q \times r \times \mathcal{P}_\omega(r) \times Q$, an initial state $q_0 \in Q$, a set $F \subseteq Q$ of final states, and an initial register assignment u. Register assignments are partial maps $v : r \rightharpoonup \mathbb{A}$, which means a register $k \in r$ can be empty ($v(k) = \bot$) or hold a name from $\mathbb{A}$. An *FSUBA configuration* is a pair (q, v), where $q \in Q$ and v is a register assignment. The initial configuration is (q_0, u). A transition $(q, k, S, p) \in \mu$ applies to a configuration with state q for an input symbol $a \in \mathbb{A}$ if register k is empty or holds a; the resulting configuration has state p, with the input a first written into register k and the register contents from S cleared afterwards. A word is accepted if there is a sequence of transitions from (q_0, u) to a configuration with a final state.

As the name *unification-based* suggests, FSUBAs can check equality of input symbols, but not inequality (except with respect to the read-only letters); in other words, they have no notion of freshness. Thus the above-mentioned language $\{aba \mid a \neq b\}$ cannot be accepted by an FSUBA [13].

The configurations of an FSUBA A are a nominal set C under the group action $\pi\cdot(q, v) = (q, \pi\cdot v)$. We show in the appendix that one can equip C with the structure of an RNNA that accepts the same $\mathbb{A}$-language as A under local freshness semantics; that is, RNNAs are strictly more expressive than FSUBAs with empty read-only alphabet.

6 Deciding Inclusion under Global and Local Freshness

We next show that under both global and local freshness, the inclusion problem for regular bar expressions (equivalently bar NFAs) is in ExpSpace. In view of Remark 4.2, for global freshness, this just reproves the known decidability of inclusion for session automata [4] in a marginally more general setting (the complexity bound is not stated in [4] but can be extracted from the decidability proof), while the result for local freshness appears to be new. Our algorithm is mildly different from the one suggested in [4] in that it exploits name dropping; we describe it explicitly, as we will modify it for local freshness.

▶ **Theorem 6.1.** *The inclusion problem for bar NFAs (or regular bar expressions) is in* ExpSpace*; more precisely, the inclusion $L_\alpha(A_1) \subseteq L_\alpha(A_2)$ can be checked using space polynomial in the size of A_1 and A_2 and exponential in* $\mathsf{deg}(A_2)\log(\mathsf{deg}(A_1) + \mathsf{deg}(A_2) + 1)$.

The theorem can be rephrased as saying that bar language inclusion of NFA is in parameterized polynomial space (para-PSpace) [24], the parameter being the degree.

Proof (Sketch). Let A_1, A_2 be bar NFAs with initial states s_1, s_2. We exhibit an NExpSpace procedure to check that $L_\alpha(A_1)$ is *not* a subset of $L_\alpha(A_2)$, which implies the claimed bound by Savitch's theorem. It maintains a state q of A_1 and a set Ξ of states in the name-dropping RNNA $\bar{A}_2$ generated by A_2 as described in Construction 4.4, with q initialized to s_1 and Ξ to $\{(s_2, \mathsf{id}F_{N_{s_2}})\}$. It then iterates the following:

1. Guess a transition $q \xrightarrow{\alpha} q'$ in A_1 and update q to q'.
2. Compute the set Ξ' of all states of $\bar{A}_2$ reachable from states in Ξ via α-transitions (literally, i.e. not up to α-equivalence) and update Ξ to Ξ'.

The algorithm terminates successfully and reports that $L_\alpha(A_1) \not\subseteq L_\alpha(A_2)$ if it reaches a final state q of A_1 while Ξ contains only non-final states.

Correctness of the algorithm follows from Theorem 4.5 and Lemma 3.7. To analyze space usage, first recall that cosets πF_N can be represented as injective renamings $N \to \mathbb{A}$. Note that Ξ will only ever contain states $(q, \pi F_N)$ such that the image πN of the corresponding injective renaming is contained in the set P of names occurring literally in either A_1 or A_2. In fact, at the beginning, $\mathsf{id}N_{s_2}$ consists only of names literally occurring in A_2, and the only names that are added are those occurring in transitions guessed in Step 1, i.e. occurring literally in A_1. So states $(q, \pi F_N)$ in Ξ can be coded using partial functions $N_q \rightharpoonup P$. There are only exponentially many such states; noting that $\sharp P \leq \mathsf{deg}(A_1) + \mathsf{deg}(A_2)$, there are at most $k \cdot (\mathsf{deg}(A_1) + \mathsf{deg}(A_2) + 1)^{\mathsf{deg}(A_2)} = k \cdot 2^{\mathsf{deg}(A_2) \log(\mathsf{deg}(A_1)+\mathsf{deg}(A_2)+1)}$ such states, where k is the number of states of A_2. ◀

▶ **Remark 6.2.** The translation from NKA expressions to regular bar expressions from Section 3 increases expression size exponentially but the degree only linearly. Therefore, the EXPSPACE upper bound on inclusion for NKA expressions [15] follows from Theorem 6.1.

We now adapt the inclusion algorithm to local freshness semantics.

▶ **Definition 6.3.** We denote by $\sqsubseteq$ the preorder (in fact: order) on $\bar{\mathbb{A}}^*$ generated by $wav \sqsubseteq w|av$.

▶ **Lemma 6.4.** *Let L_1, L_2 be bar languages accepted by RNNA. Then $D(L_1) \subseteq D(L_2)$ iff for each $[w]_\alpha \in L_1$ there exists $w' \sqsupseteq w$ such that $[w']_\alpha \in L_2$.*

▶ **Corollary 6.5.** *Inclusion $D(L_\alpha(A_1)) \subseteq D(L_\alpha(A_2))$ of bar NFAs (or regular bar expressions) under local freshness semantics is in para-*PSPACE*, with parameter* $\mathsf{deg}(A_2) \log(\mathsf{deg}(A_1) + \mathsf{deg}(A_2) + 1)$.

Proof. By Lemma 6.4, we can use a modification of the above algorithm, where Ξ' additionally contains states of $\bar{A}_2$ reachable from states in Ξ via $|a$-transitions in case α is a free name a. ◀

7 Conclusions and Future Work

We have studied the global and local freshness semantics of *regular nondeterministic nominal automata*, which feature explicit name-binding transitions. We have shown that RNNAs are equivalent to session automata [4] under global freshness and to *non-spontaneous* and *name-dropping* nondeterministic orbit-finite automata [3] under local freshness. Under both semantics, RNNAs are comparatively well-behaved computationally, and in particular admit inclusion checking in parameterized polynomial space. While this reproves known results on session automata under global freshness, decidability of inclusion under local freshness appears to be new, and in fact other nondeterministic automata models for local freshness tend to have undecidable inclusion problems (e.g. finite memory automata (FMAs) with more than two registers [12], nondeterministic orbit-finite automata [3], and variable automata [10]). In terms of expressivity, RNNAs lie strictly between finite unification-based automata without read-only symbols [13] and FMAs.

We leave the implementation of our calculus, possibly transferring efficient methods for equivalence checking of NFAs using bisimulation up to congruence [5] to the nominal setting, as future work. Another challenge is to add support for deallocation operators in the spirit of dynamic sequences [9] to the framework.

Acknowledgements We wish to thank Charles Paperman for useful discussions. We also thank the anonymous reviewers of an earlier version of this paper for their insightful comments that helped us improve the paper, and in particular led us to discover the crucial notion of name dropping.

References

1 M. Bielecki, J. Hidders, J. Paredaens, J. Tyszkiewicz, and J. V. den Bussche. Navigating with a browser. In *Automata, Languages and Programming, ICALP 2002*, vol. 2380 of *LNCS*, pp. 764–775. Springer, 2002.

2 M. Bojańczyk. Automata for data words and data trees. In *Rewriting Techniques and Applications, RTA 2010*, vol. 6 of *LIPIcs*, pp. 1–4. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

3 M. Bojanczyk, B. Klin, and S. Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10, 2014.

4 B. Bollig, P. Habermehl, M. Leucker, and B. Monmege. A robust class of data languages and an application to learning. *Log. Meth. Comput. Sci.*, 10, 2014.

5 F. Bonchi and D. Pous. Checking NFA equivalence with bisimulations up to congruence. In *Principles of Programming Languages, POPL 2013*, pp. 457–468. ACM, 2013.

6 V. Ciancia and E. Tuosto. A novel class of automata for languages on infinite alphabets. Technical report, University of Leicester, 2009. CS-09-003.

7 M. J. Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bull. Symbolic Logic*, 17(2):161–229, 2011.

8 M. J. Gabbay and V. Ciancia. Freshness and name-restriction in sets of traces with names. In *Foundations of Software Science and Computational Structures, FOSSACS 2011*, vol. 6604 of *LNCS*, pp. 365–380. Springer, 2011.

9 M. J. Gabbay, D. R. Ghica, and D. Petrisan. Leaving the nest: Nominal techniques for variables with interleaving scopes. In *Computer Science Logic, CSL 2015*, vol. 41 of *LIPIcs*, pp. 374–389. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

10 O. Grumberg, O. Kupferman, and S. Sheinvald. Variable automata over infinite alphabets. In *Language and Automata Theory and Applications, LATA 2010*, vol. 6031 of *LNCS*, pp. 561–572. Springer, 2010.

11 M. Hennessy. A fully abstract denotational semantics for the pi-calculus. *Theor. Comput. Sci.*, 278:53–89, 2002.

12 M. Kaminski and N. Francez. Finite-memory automata. *Theor. Comput. Sci.*, 134:329–363, 1994.

13 M. Kaminski and T. Tan. Regular expressions for languages over infinite alphabets. *Fund. Inform.*, 69:301–318, 2006.

14 M. Kaminski and D. Zeitlin. Finite-memory automata with non-deterministic reassignment. *Int. J. Found. Comput. Sci.*, 21:741–760, 2010.

15 D. Kozen, K. Mamouras, D. Petrisan, and A. Silva. Nominal Kleene coalgebra. In *Automata, Languages, and Programming, ICALP 2015*, vol. 9135 of *LNCS*, pp. 286–298. Springer, 2015.

16 D. Kozen, K. Mamouras, and A. Silva. Completeness and incompleteness in nominal kleene algebra. In *Relational and Algebraic Methods in Computer Science, RAMiCS 2015*, vol. 9348 of *LNCS*, pp. 51–66. Springer, 2015.

17 K. Kürtz, R. Küsters, and T. Wilke. Selecting theories and nonce generation for recursive protocols. In *Formal methods in security engineering, FMSE 2007*, pp. 61–70. ACM, 2007.

18 A. Kurz, T. Suzuki, and E. Tuosto. On nominal regular languages with binders. In *Foundations of Software Science and Computational Structures, FOSSACS 2012*, vol. 7213 of *LNCS*, pp. 255–269. Springer, 2012.

19 L. Libkin, T. Tan, and D. Vrgoc. Regular expressions for data words. *J. Comput. Syst. Sci.*, 81:1278–1297, 2015.

20 S. Milius, L. Schröder, and T. Wißmann. Regular behaviours with names: On rational fixpoints of endofunctors on nominal sets. submitted; available at `http://www8.cs.fau.de/ext/thorsten/nomliftings.pdf`.

21 F. Neven, T. Schwentick, and V. Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Log.*, 5:403–435, 2004.

22 A. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press, 2013.

23 L. Segoufin. Automata and logics for words and trees over an infinite alphabet. In *Computer Science Logic, CSL 2006*, vol. 4207 of *LNCS*, pp. 41–57. Springer, 2006.

24 C. Stockhusen and T. Tantau. Completeness results for parameterized space classes. In *Parameterized and Exact Computation, IPEC 2013*, vol. 8246 of *LNCS*, pp. 335–347. Springer, 2013.

25 A. Tal. Decidability of inclusion for unification based automata. Master's thesis, Technion, 1999.

26 D. Turner and G. Winskel. Nominal domain theory for concurrency. In *Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings*, pp. 546–560, 2009.

27 T. Wysocki. Alternating register automata on finite words. Master's thesis, University of Warsaw, 2013. (In Polish).

A Omitted Proofs

A.1 Abstraction in Nominal Sets

We occasionally use, without express mention, the following alternative description of equality in the abstraction $[\mathbb{A}]X$, which formalizes the usual intuitions about α-equivalence:

▶ **Lemma A.1.** *Let $a, b \in \mathbb{A}$ and $x, y \in X$. Then $\langle a\rangle x = \langle b\rangle y$ in $[\mathbb{A}]X$ iff either*

(i) $(a, x) = (b, y)$, *or*
(ii) $b \neq a$, $b \# x$, *and* $(ab) \cdot x = y$.

Proof. *'If':* the case where (i) holds is trivial, so assume (ii). Let c be fresh; we have to show $(ca) \cdot x = (cb) \cdot y$. But $(cb) \cdot y = (cb) \cdot (ab) \cdot x = (acb) \cdot x = (ca) \cdot x$, where we use in the last step that b, c are both fresh for x so that $(ca)^{-1}(acb) = (ca)(acb) = (bc)$ fixes x.

'Only if': We assume $(a, x) \neq (b, y)$ and prove (ii). We first show $a \neq b$: Assume the contrary. Let c be fresh; by the definition of abstraction, we then have $(ca) \cdot x = (cb) \cdot y$, so $y = (cb)(ca) \cdot x = (ca)(ca) \cdot x = x$, contradiction. We have $\mathsf{supp}(x) \subseteq \{a\} \cup \mathsf{supp}(\langle a\rangle x) = \{a\} \cup \mathsf{supp}(\langle b\rangle y)$, whence $b \# x$ since $a \neq b$ and $b \# \langle b\rangle y$. Finally, with c as above $y = (cb)^{-1}(ca) \cdot x = (cb)(ca) \cdot x = (acb) \cdot x = (ab) \cdot x$, again because $(ab)^{-1}(abc) = (ab)(abc) = (bc)$ and b, c are fresh for x. ◀

As an easy consequence we obtain:

▶ **Corollary A.2.** *Let X be a nominal set, $a \in \mathbb{A}$ and $x \in X$. Then* $\mathsf{supp}(\langle a\rangle x) = \mathsf{supp}(x) - \{a\}$.

Proof of Lemma 2.2 Firstly, any finite set $S \subseteq X$ is ufs, because $\bigcup_{y\in S} \mathsf{supp}(y)$ is a finite union of finite sets. Secondly, for any ufs $S \subseteq X$, we have $\mathsf{supp}(S) = \bigcup_{y\in S} \mathsf{supp}(y)$, which is a finite union (because X is orbit-finite) of again finite sets. ◀

A.2 Proofs and Lemmas for Section 3

NOFAs as coalgebras We show that the standard description of NOFAs as repeated at the beginning of Section 3 is equivalent to the one as F-coalgebras for $FX = 2 \times \mathcal{P}_{\mathsf{fs}}(\mathbb{A} \times X)$. For the direction from the standard description to F-coalgbras, recall that the transition relation is assumed to be equivariant; therefore, the map taking a state q to $\{(a, q') \mid q \xrightarrow{a} q'\}$ is equivariant, hence preserves supports and therefore ends up in FQ where Q is the set of states. Conversely, let $\xi : Q \to FQ$ be an F-coalgebra with components $f : Q \to 2$, $g : Q \to \mathcal{P}_{\mathsf{fs}}(\mathbb{A} \times Q)$. Define the transition relation on Q by $q \xrightarrow{a} q'$ iff $(a, q') \in g(Q)$, and make q final iff $f(q) = \top$. Then finality is equivariant by equivariance of f. To see that the transition relation is equivariant let $q \xrightarrow{a} q'$ and $\pi \in G$. Then $(\pi a, \pi q') \in \pi(g(q)) = g(\pi(q))$ by equivariance of g, i.e. $\pi q \xrightarrow{\pi a} \pi q'$. ◀

▶ **Definition A.3.** Given a state q in A we write $L_0(q)$ and $L_\alpha(q)$ for the literal language and the bar language, respectively, accepted by the automaton obtained by making q the initial state of A.

▶ **Lemma A.4.** *In an RNNA, the map $q \mapsto L_\alpha(q)$ is equivariant.*

Proof. Note first that the set of bar strings $\bar{\mathbb{A}}^*$ is the initial algebra for the functor $SX = 1 + \mathbb{A} \times X + \mathbb{A} \times X$ on Nom. And the set $\bar{M}$ of bar strings modulo α-equivalence is the intial algebra for the functor $S_\alpha X = 1 + \mathbb{A} \times X + [\mathbb{A}]X$ on Nom. The functor S_α is a quotient of the functor S via the natural transformation $q : S \twoheadrightarrow S_\alpha$ given by the canonical quotient maps $\mathbb{A} \times X \twoheadrightarrow [\mathbb{A}]X$. The canonical quotient map $[-]_\alpha : \bar{\mathbb{A}}^* \twoheadrightarrow \bar{M}$ that maps every bar string to its α-equivalence class is obtained inductively, i.e. $[-]_\alpha$ is the unique equivariant map such that the following square commutes:

$$\begin{array}{ccccc} S\bar{\mathbb{A}}^* & & \xrightarrow{\iota} & & \bar{\mathbb{A}}^* \\ \downarrow{\scriptstyle S[-]_\alpha} & & & & \downarrow{\scriptstyle [-]_\alpha} \\ S\bar{M} & \xrightarrow[q_{\bar{M}}]{} & S_\alpha \bar{M} & \xrightarrow[\iota_\alpha]{} & \bar{M} \end{array}$$

where $\iota : S\bar{\mathbb{A}}^* \to \bar{\mathbb{A}}^*$ and $\iota_\alpha : S_\alpha\bar{M} \to \bar{M}$ are the structures of the initial algebras, respectively. Since the map $[-]_\alpha$ is equivariant we thus have $\pi[w]_\alpha = [\pi w]_\alpha$ for every $w \in \bar{\mathbb{A}}^*$.

Now we prove the statement of the lemma. Since both free and bound transitions are equivariant, the literal language $L_0(-)$ is equivariant. It follows that the bar language $L_\alpha(-)$ is equivariant: If $m \in L_\alpha(q)$ then there is $w \in L_0(q)$ such that $[w]_\alpha = m$. For $\pi \in G$, it follows that $\pi \cdot w \in L_0(\pi q)$, and hence $[\pi \cdot w]_\alpha \in L_\alpha(\pi q)$. But $[\pi \cdot w]_\alpha = \pi[w]_\alpha$, so $\pi m \in L_\alpha(\pi q)$. ◀

Proof of Lemma 3.4

1. Consider the ufs set $Z = \{(a, q') \mid q \xrightarrow{a} q'\}$. Then we have $\mathsf{supp}(q') \cup \{a\} = \mathsf{supp}(a, q') \subseteq \mathsf{supp}(Z) \subseteq \mathsf{supp}(q)$ where the second inclusion holds because Z is ufs, and the third because Z depends equivariantly on q.
2. Consider the ufs set $Z = \{[a]q' \mid q \xrightarrow{|a} q'\}$. Then we have $\mathsf{supp}(q') \subseteq \mathsf{supp}([a]q') \cup \{a\} \subseteq \mathsf{supp}(Z) \cup \{a\} \subseteq \mathsf{supp}(q) \cup \{a\}$ where the second inclusion holds because Z is ufs, and the third because Z depends equivariantly on q. ◀

▶ **Remark A.5.** Given an RNNA A with the state set Q the paths in A form the initial algebra for the functor $Q \times S(-)$, where S is the functor in the proof of Lemma A.4. Paths in A modulo α-equivalence then form the initial algebra for $Q \times S_\alpha(-)$ and the canonical quotient map $[-]_\alpha$ mapping a path to its α-equivalence class is obtained by initiality similarly as the canonical quotient map in Lemma A.4.

Proof of Lemma 3.7

▶ **Lemma A.6.** *Let A be a name-dropping RNNA, and let $q|_N$ restrict a state q in A to $N \subseteq \mathsf{supp}(q)$. Then $\{w \in L_0(q) \mid \mathsf{FN}(w) \subseteq N\} \subseteq L_0(q|_N)$.*

Proof. Induction on the length of $w \in L_0(q)$ with $\mathsf{FN}(w) \subseteq N$, with the base case immediate from the finality condition in Definition 3.6. So let $w = \alpha v$ with $\alpha \in \bar{\mathbb{A}}$, accepted via a path $q \xrightarrow{\alpha} q' \xrightarrow{v} p$, and let $q'|_{N_v}$ restrict q' to $N_v := \mathsf{FN}(v)$. By the induction hypothesis, $v \in L_0(q'|_{N_v})$. Moreover, $q \xrightarrow{\alpha} q'|_{N_v}$. We are done once we show that $q|_N \xrightarrow{\alpha} q'|_{N_v}$. If α is free, then we have to show $N_v \cup \{\alpha\} \subseteq N$, and if $\alpha = |a$ is bound, we have to show $N_v \subseteq N \cup \{a\}$. In both cases, the requisite inclusion is immediate from $\mathsf{FN}(\alpha v) \subseteq N$. ◀

Proof (Lemma 3.7). We use induction on the word length. It suffices to show that $L_0(A)$ is closed under single α-conversion steps. So let $v|aw \in \bar{\mathbb{A}}^* \in L_0(A)$, via a path $s = q_0 \xrightarrow{v} q_1 \xrightarrow{|a} q_2 \xrightarrow{w} q_3$ (with q_3 final), let $b \neq a$ with $b \notin \mathsf{FN}(w)$, and let w' be obtained from w by replacing free occurrences of a with b. We have to show that $v|bw' \in L_0(A)$; it suffices to show that $|bw' \in L_0(q_1)$. Put $N = \mathsf{supp}(q_2) - \{b\}$, and let $q_2|_N$ restrict q_2 to N. By Lemma A.6, $w \in L_0(q_2|_N)$, and hence $[w]_\alpha \in L_\alpha(q_2|_N)$. By Lemma A.4, it follows that $[w']_\alpha \in L_\alpha((ab)(q_2|_N))$, so by the induction hypothesis, $w' \in L_0((ab)(q_2|_N))$. We clearly have $q_1 \xrightarrow{|a} q_2|_N$, and by α-invariance, $q_1 \xrightarrow{|b} (ab)(q_2|_N)$ because $b \# (q_2|_N)$.

Thus, $|bw' \in L_0(q_1)$ as required. ◀

Proof of Lemma 3.8

▶ **Definition A.7.** Given the transition data of an RNNA A (not necessarily assuming any finiteness and invariance conditions) and a state q in A, we denote by $\mathsf{fsuc}(q)$ the set

$$\mathsf{fsuc}(q) = \{(a, q') \mid q \xrightarrow{a} q'\}$$

of free transitions of q, and by $\mathsf{bsuc}(q)$ the set

$$\mathsf{bsuc}(q) = \{\langle a\rangle q' \mid q \xrightarrow{\mathsf{l}a} q'\}$$

of bound transitions of q modulo α-equivalence.

Note that under α-invariance of transitions we have $\langle a\rangle q' \in \mathsf{bsuc}(q)$ if and only if $q \xrightarrow{\mathsf{l}a} q'$.

Before we proceed to the proof of the lemma we note the following general fact about nominal sets: for the value of $\pi \cdot x$, it matters only what π does on the atoms in $\mathsf{supp}(x)$:

▶ **Lemma A.8.** *For $x \in (X, \cdot)$ and any $\pi, \sigma \in G$ with $\pi(v) = \sigma(v)$ for all $v \in \mathsf{supp}(x)$, we have $\pi \cdot x = \sigma \cdot x$.*

Proof. Under the given assumptions, $\pi^{-1}\sigma \in \mathsf{Fix}(\mathsf{supp}(x)) \subseteq \mathsf{fix}(x)$. ◀

Proof (Lemma 3.8). Let A be an RNNA with set Q of states.

(1) We construct an equivalent name-dropping RNNA A' as follows. As states, we take pairs

$$q|_N := \mathsf{Fix}(N)q$$

where $q \in Q$, $N \subseteq \mathsf{supp}(q)$, and $\mathsf{Fix}(N)q$ denotes the orbit of q under $\mathsf{Fix}(N)$. We define an action of G on states by $\pi \cdot (q|_N) = (\pi q)|_{\pi N}$. To see well-definedness, let $\pi' \in \mathsf{Fix}(N)$ (i.e. $(\pi' q)|_N = q|_N$); we have to show $(\pi\pi' q)|_{\pi N} = (\pi q)|_{\pi N}$. Since $(\pi\pi'\pi^{-1})\pi q = \pi\pi' q$, this follows from $\pi\pi'\pi^{-1} \in \mathsf{Fix}(\pi N)$. The map $(q, N) \mapsto q|_N$ is equivariant, which proves the bound on the number of orbits in A'. A state $q|_N$ is final if q is final in A; this clearly yields an equivariant subset of states of A'. The initial state of A' is $s|_{\mathsf{supp}(s)}$ where s is the initial state of A. We have

$$\mathsf{supp}(q|_N) = N; \tag{2}$$

in particular, the states of A' form a nominal set. To see '$\subseteq$' in (2), it suffices to show that N supports $q|_N$. So let $\pi \in \mathsf{Fix}(N)$. Then $\pi \cdot (q|_N) = (\pi q)|_{\pi N} = q|_N$, as required. For '$\supseteq$', let $a \in N$; we have to show that $N - \{a\}$ does not support $q|_N$. Assume the contrary. Pick $b \# q$. Then $(ab) \in \mathsf{Fix}(N - \{a\})$, so $(ab) \cdot (q|_N) = \mathsf{Fix}((ab) \cdot N)(ab) \cdot q = \mathsf{Fix}(N)q = q|_N$. In particular, $q \in \mathsf{Fix}(ab) \cdot N)(ab) \cdot q$, i.e. there is $\rho \in \mathsf{Fix}((ab) \cdot N)$ such that $\rho(ab) \cdot q = q$. By equivariance of supp, it follows that $\rho(ab) \cdot \mathsf{supp}(q) = \mathsf{supp}(q)$. Now $b \in (ab) \cdot N$, so $\rho(b) = b$. Since $a \in \mathsf{supp}(q)$, it follows that $b \in \rho(ab) \cdot \mathsf{supp}(q)$; but $b \notin \mathsf{supp}(q)$, contradiction.

As transitions of A', we take

- $q|_N \xrightarrow{a} q'|_{N'}$ whenever $q \xrightarrow{a} q'$, $N' \subseteq N$, and $a \in N$, and
- $q|_N \xrightarrow{\mathsf{l}a} q'|_{N'}$ whenever $q \xrightarrow{\mathsf{l}b} q''$, $N'' \subseteq \mathsf{supp}(q'') \cap (N \cup \{b\})$, and $\langle a\rangle(q'|_{N'}) = \langle b\rangle(q''|_{N''})$.

(We do not require the converse implications. E.g. $q|_N \xrightarrow{a} q'|_{N'}$ need not imply that $q \xrightarrow{a} q'$, only that $\pi q \xrightarrow{a} q'$ for some $\pi \in \mathsf{Fix}(N)$; see also (3) below.) Transitions are clearly equivariant. Moreover, bound transitions are, by construction, α-invariant.

▶ **Fact A.9.** *By construction, every bound transition in A' is α-equivalent*[1] *to one of the form $q|_N \xrightarrow{\mathsf{l}a} q'|_{N'}$ where $q \xrightarrow{\mathsf{l}a} q'$ and $N' \subseteq \mathsf{supp}(q') \cap (N \cup \{a\})$.*

---

[1] Recall that a transition $q \xrightarrow{\mathsf{l}a} q'$ is α-equivalent to a transition $r \xrightarrow{\mathsf{l}b} r'$ if $q = r$ and $\langle a\rangle q' = \langle b\rangle r'$.

(2) To see ufs branching, let $q|_N$ be a state in A'. For free transitions, we have to show that the set

$$\mathsf{fsuc}(q|_N) = \{(a, q'|_{N'}) \mid N' \subseteq N, a \in N, \pi q \xrightarrow{a} q' \text{ for some } \pi \in \mathsf{Fix}(N)\}$$

of free successors of $q|_N$ is ufs. But for $\pi \in \mathsf{Fix}(N)$, $N' \subseteq N$, and $a \in N$, we have $\pi q \xrightarrow{a} q'$ iff $q \xrightarrow{a} \pi^{-1}q'$, and then moreover $\pi^{-1} \in \mathsf{Fix}(N')$ so $\mathsf{Fix}(N')\pi^{-1}q' = \mathsf{Fix}(N')q'$, i.e. $q'|_{N'} = (\pi^{-1}q')|_{N'}$. We thus have

$$\begin{aligned}\mathsf{fsuc}(q|_N) &= \{(a, (\pi^{-1}q')|_{N'} \mid N' \subseteq N, a \in N, q \xrightarrow{a} \pi^{-1}q'\} \\ &= \{(a, q'|_{N'}) \mid N' \subseteq N, a \in N, q \xrightarrow{a} q'\},\end{aligned} \tag{3}$$

which is ufs.

We proceed similarly for the bound transitions: We need to show that the set $\mathsf{bsuc}(q|_N)$ of bound successors of $q|_N$ is ufs. By Fact A.9, a bound transition $q|_N \xrightarrow{|a} q'|_{N'}$ arises from $\pi \in \mathsf{Fix}(N)$ and $N' \subseteq \mathsf{supp}(q') \cap (N \cup \{a\})$ such that $\pi q \xrightarrow{|a} q'$. Then $q \xrightarrow{|(\pi^{-1}a)} \pi^{-1}q'$. Moreover, we claim that

$$\langle a\rangle(q'|_{N'}) = \langle \pi^{-1}a\rangle(\pi^{-1}(q'|_{N'})). \tag{4}$$

To see (4), we distinguish two cases: If $\pi^{-1}(a) = a$ then the two sides are equal because π^{-1} fixes the support of $q'|_{N'}$. If $\pi^{-1}(a) \neq a$ then $\pi^{-1}a \notin N$ because π^{-1} fixes N, so $\pi^{-1}a \notin N \cup \{a\}$ and therefore $\pi^{-1}a \notin N' = \mathsf{supp}(q'|_{N'})$. This means that we can α-equivalently rename a into $\pi^{-1}a$ in $(|a, q'|_{N'})$; since π^{-1} fixes N, the result of this renaming equals $(|\pi^{-1}a, \pi^{-1}q'|_{N'})$. Since $\pi^{-1}(q'|_{N'}) = (\pi^{-1}q)|_{\pi^{-1}N'}$ and $\pi^{-1}N' \subseteq \mathsf{supp}(\pi^{-1}q') \cap (N \cup \{\pi^{-1}(a)\})$ (recall $\pi \in \mathsf{Fix}(N)$), (4) proves

$$\begin{aligned}\mathsf{bsuc}(q|_N) &= \{\langle \pi^{-1}a\rangle((\pi^{-1}q')|_{\pi^{-1}N'}) \mid \\ &\qquad \pi \in \mathsf{Fix}(N), \pi^{-1}N' \subseteq \mathsf{supp}(\pi^{-1}q') \cap (N \cup \{\pi^{-1}a\}), q \xrightarrow{|\pi^{-1}a} \pi^{-1}q'\} \\ &= \{\langle a\rangle(q'|_{N'}) \mid N' \subseteq \mathsf{supp}(q') \cap (N \cup \{a\}), q \xrightarrow{|a} q'\}.\end{aligned} \tag{5}$$

By (5), $\mathsf{bsuc}(q|_N)$ is ufs; indeed, we have $\mathsf{supp}(\langle a\rangle(q'_{N'})) = N' - \{a\}$ so the support of every element of $\mathsf{bsuc}(q|_N)$ is a subset of N. (Note that (5) is not the same as Fact A.9, as in (5) we use a fixed representative q of $\mathsf{Fix}(N)q$.)

(3) We show next that A' is name-dropping. So let $q|_N$ be a state in A', and let $N' \subseteq \mathsf{supp}(q|_N) = N$. We show that $q|_{N'}$ restricts $q|_N$ to N'. We first establish that $q|_{N'}$ has at least the same incoming transitions as $q|_N$. For the free transitions, let $\pi \in \mathsf{Fix}(N)$, $q' \xrightarrow{a} \pi q$ and $a \in N'' \supseteq N$, so that $q'|_{N''} \xrightarrow{a} (\pi q)|_N = q|_N$. Then also $\pi \in \mathsf{Fix}(N')$ and $N' \subseteq N''$, so $q'|_{N''} \xrightarrow{a} \pi q|_{N'} = q|_{N'}$ as required. For the bound transitions, let $\pi \in \mathsf{Fix}(N)$, let $\langle a\rangle(q|_N) = \langle b\rangle((ab) \cdot (q|_N))$, let $q' \xrightarrow{|b} \pi(ab) \cdot q$ where $\pi \in \mathsf{Fix}((ab)N)$, and let $(ab) \cdot N \subseteq N'' \cup \{b\}$, so that $q'|_{N''} \xrightarrow{|a} q|_N$. We have to show that $q'|_{N''} \xrightarrow{|a} q|_{N'}$. From $q' \xrightarrow{|b} \pi(ab) \cdot q$ we have $q|_{N''} \xrightarrow{|b} ((\pi(ab)q)|_{(ab)N'}) = (((ab)q)|_{(ab)N'})$, because $(ab)N' \subseteq (ab)N \subseteq N'' \cup \{b\}$ and $\pi \in \mathsf{Fix}((ab)N) \subseteq \mathsf{Fix}((ab)N')$. If $b = a$, we are done. So assume $b \neq a$. Since A' is α-invariant, it remains only to show that

$$\langle b\rangle((ab)q|_{(ab)N'}) = \langle a\rangle(q|_{N'}),$$

i.e. that $b \notin \mathsf{supp}(q|_{N'})$; but since $b \neq a$ and $\langle a\rangle(q|_N) = \langle b\rangle((ab) \cdot (q|_N))$, we even have $b \notin \mathsf{supp}(q|_N) \supseteq N' \supseteq \mathsf{supp}(q|_{N'})$.

Next, we show that $q|_{N'}$ has the requisite outgoing transitions. For the free transitions, let $q|_N \xrightarrow{a} q'|_M$ where $\mathsf{supp}(q'|_M) \cup \{a\} = M \cup \{a\} \subseteq N'$. We have to show $q|_{N'} \xrightarrow{a} q'|_M$. By (3), we have $q \xrightarrow{a} \pi q'$ for some $\pi \in \mathsf{Fix}\, M$. By construction of A', $q|_{N'} \xrightarrow{a} (\pi q')|_M = q'|_M$, as required.

For the bound transitions, we proceed as follows. By (5), a given outgoing bound transition of $q|_N \xrightarrow{|a} q'|_M$ yields a state $r|_S$ of A' and $b \in \mathbb{A}$ such that $\langle a\rangle q'|_M = \langle b\rangle r|_S$, $S \subseteq \mathsf{supp}(r) \cap (N \cup \{a\})$ and $q \xrightarrow{|b} r$.

Now if $M \subseteq N' \cup \{a\}$ this yields a transition $q|_{N'} \xrightarrow{|a} q'|_M$ by construction of A'; indeed, we already have $q \xrightarrow{|b} r$, $S \subseteq \mathsf{supp}(r)$ and $\langle a\rangle q'|_M = \langle b\rangle r|_S$, so it remains to show that $S \subseteq N' \cup \{b\}$. By Lemma A.1, $\langle a\rangle q'|_M = \langle b\rangle r|_S$ iff either $a = b$ and $q'|_M = r|_S$ (and the latter yields $M = S$, thus we are done), or $a \neq b$, $a \,\#\, r|_S$ and $(ba)(r|_S) = q'|_M$. It follows that $a \notin S$ and $M = (ba)S$. Since $(ba)S \subseteq N' \cup \{a\}$ we have equivalently $S \subseteq (ba)N' \cup \{b\}$. This implies $S \subseteq N' \cup \{b\}$ using that $a \notin S$.

(4) It remains to show that $L_\alpha(A') = L_\alpha(A)$. To show '$\subseteq$', we show that $[w]_\alpha \in L_\alpha(q)$ for every state $q|_N$ in A' and every $w \in L_0(q|_N)$, by induction on w: for the empty word, the claim follows from the definition of final states in A'. For $w = \alpha v$, let $q|_N \xrightarrow{\alpha} q'|_{N'} \xrightarrow{v} t$ be an accepting path in A'. Then we have $[v]_\alpha v \in L_\alpha(q')$ by induction hypothesis and $\mathsf{FN}(v) \subseteq N'$ by Corollary 3.5. By (3) and (5), we have $q \xrightarrow{\alpha} \pi q'$ for some $\pi \in \mathsf{Fix}\, N'$. It follows that $\pi \cdot v = v$ and therefore $[v]_\alpha = \pi[v]_\alpha \in L_\alpha(\pi q')$ by the equivariance of L_α (see Lemma A.4). Hence $[\alpha v]_\alpha \in L_\alpha(q)$.

To see $L_\alpha(A') \supseteq L_\alpha(A)$, it suffices to note that A is included as a subautomaton in A' via the map that takes q to $q|_{\mathsf{supp}(q)}$, i.e. $q \xrightarrow{\alpha} q'$ in A implies $q|_{\mathsf{supp}(q)} \xrightarrow{\alpha} q'|_{\mathsf{supp}(q')}$ in A'. ◀

A.3 Proofs and Lemmas for Section 4

▶ **Lemma A.10.** *Let q be a state in a bar NFA; then $L_\alpha(q)$ is ufs.*

Proof. The finitely many transitions of A only mention letters from a finite subset of $\bar{\mathbb{A}}$, and $\bigcup_{w \in L_\alpha(q)} \mathsf{supp}(w)$ is contained in that finite subset. ◀

Proof of Theorem 4.5

As indicated in the text, we split the construction into two parts, and first construct a plain RNNA $\tilde{A}$. The states of $\tilde{A}$ are pairs

$$(q, \pi H_q) \quad \text{where } H_q = \mathsf{Fix}(\mathsf{supp}(L_\alpha(q)))$$

consisting of a state q in A and a left coset πH_q, where the action of G is as on $\bar{A}$:

$$\pi_1 \cdot (q, \pi_2 H_q) = (q, \pi_1\pi_2 H_q).$$

We continue to write $N_q = \mathsf{supp}(L_\alpha(q))$ (note $H_q = F_{N_q}$ in the notation used in the construction of $\bar{A}$). The initial state of $\tilde{A}$ is (s, H_s) where s is the initial state of A; a state $(q, \pi H_q)$ is final in $\tilde{A}$ iff q is final in A. Free transitions in $\tilde{A}$ are of the form

$$(q, \pi H_q) \xrightarrow{\pi(a)} (q', \pi H_{q'}) \quad \text{where } q \xrightarrow{a} q' \text{ and } a \in N_q,$$

(where the condition $a \in N_q$ is automatic unless $L_\alpha(q') = \emptyset$) and bound transitions are of the form

$$(q, \pi H_q) \xrightarrow{|a} (q', \pi' H_{q'}) \;\text{ where }\; q \xrightarrow{|b} q' \;\text{ and }\; \langle a\rangle \pi' H_{q'} = \langle \pi(b)\rangle \pi H_{q'}.$$

▶ **Remark A.11.** (1) Note that by Lemma 2.1, $N_q = \mathsf{supp}(L_\alpha(q)) = \bigcup_{w \in L_\alpha(q)} \mathsf{supp}(w)$, i.e. N_q is the set of names that appear free in some word $w \in L_\alpha(q)$.

(2) Observe that $\pi H_q = \pi' H_q$ iff $\pi'(v) = \pi(v)$ for all $v \in N_q$: $\pi H_q = \pi' H_q$ iff $\pi^{-1}\pi' \in H_q$ iff $\pi^{-1}\pi'(v) = v$ for all $v \in N_q$ iff $\pi'(v) = \pi(v)$ for all $v \in N_q$.

(3) For a coset πH_q, we have

$$\mathsf{supp}(q, \pi H_q) = \mathsf{supp}(\pi H_q) = \pi N_q$$

so the set $\tilde{Q}$ of states of $\tilde{A}$ is a nominal set. This is by Item (2): for $\pi' \in G$, we have $\pi'\pi H_q = \pi H_q$ iff $\pi'\pi(a) = \pi(a)$ for all $a \in N_q$ iff $\pi' \in \mathsf{Fix}(\pi N_q)$.

(4) Note that $\langle a\rangle(\pi H_q) = \langle b\rangle(\pi' H_q)$ implies $\langle a\rangle(q, \pi H_q) = \langle b\rangle(q, \pi' H_q)$ since the action of G on states of $\tilde{A}$ is trivial in the first component.

▶ **Remark A.12.** Left cosets for H_q are in one-to-one correspondence with injections $N_q \to \mathbb{A}$. Indeed, in the light of Remark A.11(2) it suffices to prove that every injection $i : N_q \to \mathbb{A}$ can be extended to a finite permutation. Define π by

$$\pi(a) = \begin{cases} i(a) & a \in N_q \\ i^{-n}(a) & \text{else, for } n \geq 0 \text{ minimal s.t. } i^{-n}(a) \notin i[N_q] \end{cases}$$

For the proof that π is a indeed a finite permutation see [20, Corollary 2.4].

Transitions from a given state $(q, \pi H_q)$ can be characterized as follows.

▶ **Lemma A.13.** *Let $(q, \pi H_q)$ be a state in $\tilde{A}$. Then*

$$\mathsf{fsuc}(q, \pi H_q) = \{(\pi(a), (q', \pi H_{q'})) \mid q \xrightarrow{a} q', a \in N_q\} \tag{6}$$

and

$$\mathsf{bsuc}(q, \pi H_q) = \{\langle \pi(a)\rangle(q', \pi H_{q'}) \mid q \xrightarrow{|a} q'\}. \tag{7}$$

Proof. For the free transitions, we have by definition

$$\mathsf{fsuc}(q, \pi H_q) = \{(\pi'(a), (q', \pi' H_{q'})) \mid \pi' H_q = \pi H_q, q \xrightarrow{a} q', a \in N_q\}.$$

Now if $\pi' H_q = \pi H_q$ and $q \xrightarrow{a} q'$, then π and π' agree on N_q and hence on $N_{q'} \cup \{a\}$ (as $N_{q'} \cup \{a\} \subseteq N_q$), so $(\pi'(a), (q', \pi' H_{q'})) = (\pi(a), (q', \pi H_{q'}))$. This shows (6).

For the bound transitions, we have by definition and using Remark A.11(4)

$$\mathsf{bsuc}(q, \pi H_q) = \{\langle \pi'(a)\rangle(q', \pi' H_{q'}) \mid \pi' H_q = \pi H_q, q \xrightarrow{|a} q'\}.$$

So let $\pi' H_q = \pi H_q$ and $q \xrightarrow{|a} q'$. The claim (7) follows from

$$\langle \pi'(a)\rangle(q', \pi' H_{q'}) = \langle \pi(a)\rangle(q', \pi H_{q'}), \tag{8}$$

which we now prove. By Remark A.11(2) we know that π and π' agree on N_q. In order to prove (8), we distinguish two cases: if $\pi(a) = \pi'(a)$ then π and π' agree on $N_{q'} \subseteq N_q \cup \{a\}$, i.e. $\pi' H_{q'} = \pi H_{q'}$, so the two sides of (8) are literally equal. Otherwise, $a \notin N_q$, and π', π differ on $N_{q'}$ only w.r.t. their value on a. It follows that $(\pi(a)\, \pi'(a))\pi$ and π' agree on $N_{q'} = \mathsf{supp}(H_{q'})$. Therefore $(\pi(a)\, \pi'(a))\pi H_{q'} = \pi' H_{q'}$ by Lemma A.8. So, by Lemma A.1, to show (8) it suffices to show that $\pi'(a) \notin \mathsf{supp}(q', \pi H_{q'})$. But $\mathsf{supp}(q', \pi H_{q'}) = \pi N_{q'} \subseteq \pi N_q \cup \pi(a) = \pi' N_q \cup \pi(a)$, and $\pi'(a) \notin \pi' N_q \cup \pi(a)$ because $a \notin N_q$ and $\pi'(a) \neq \pi(a)$. ◀

The key ingredient in the proof that $\tilde{A}$ accepts the same bar language as A will be a normalization result on paths that uses an obvious notion of α-equivalence on paths in an RNNA (see Remark A.5); explicitly:

▶ **Definition A.14.** α-equivalence of paths in an RNNA is defined inductively by

$$q_0 \xrightarrow{a} q_1 \xrightarrow{\alpha_2} q_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_n} q_n \text{ is } \alpha\text{-equivalent to } q_0 \xrightarrow{a} q_1' \xrightarrow{\alpha_2'} q_2' \xrightarrow{\alpha_3'} \cdots \xrightarrow{\alpha_n'} q_n'$$

if $q_1 \xrightarrow{\alpha_2} q_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_n} q_n$ is α-equivalent to $q_1' \xrightarrow{\alpha_2'} q_2' \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_n'} q_n'$, and

$$q_0 \xrightarrow{|a} q_1 \xrightarrow{\alpha_2} q_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_n} q_n \text{ is } \alpha\text{-equivalent to } q_0 \xrightarrow{|b} q_1' \xrightarrow{\alpha_2'} q_2' \xrightarrow{\alpha_3'} \cdots \xrightarrow{\alpha_n'} q_n'$$

if $\langle a\rangle[q_1 \xrightarrow{\alpha_2} q_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_n} q_n]_\alpha = \langle b\rangle[q_1' \xrightarrow{\alpha_2'} q_2' \xrightarrow{\alpha_3'} \cdots \xrightarrow{\alpha_4'} q_n']_\alpha$, where we use $[-]_\alpha$ to denote α-equivalence classes of paths.

▶ **Lemma A.15.** *The set of paths of an RNNA is closed under α-equivalence.*

Proof. Observe that by equivariance, G acts pointwise on paths. It suffices to show closure under single α-conversion steps. So let $q_0 \xrightarrow{|a} q_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} q_n$ be path in an RNNA A, denote the path from q_1 onwards by P, and let $\langle a\rangle P = \langle b\rangle P'$, so $P' = (ab)\cdot P$. Then by α-invariance of $\to$, we have $q_0 \xrightarrow{|b} (ab)q_1$, and by equivariance, $(ab)\cdot P$ is a path from $(ab)q_1$. ◀

▶ **Lemma A.16.** *Let $P = q_0 \xrightarrow{|a} q_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} q_n$ be a path in an RNNA A, and let $\langle a\rangle q_1 = \langle b\rangle q_1'$. Then there exists a path in A of the form $q_0 \xrightarrow{|b} q_1' \xrightarrow{\alpha_2'} \ldots \xrightarrow{\alpha_n'} q_n'$ that is α-equivalent to P.*

Proof. Since A is an RNNA, the support of the α-equivalence class of $q_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} q_n$ is $\mathsf{supp}(q_1)$ (Remark A.17), so we obtain an α-equivalent path $q_0 \xrightarrow{|b} q_1' \xrightarrow{\alpha_2'} \ldots \xrightarrow{\alpha_n'} q_n'$ by renaming a into b in $q_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} q_n$. ◀

▶ **Remark A.17.** Note that the support of the α-equivalence class of a path in an RNNA is the support of its starting state. Indeed, let $[P]_\alpha$ be such an equivalence class and let q be the starting state of P. The inclusion $\mathsf{supp}(q) \subseteq \mathsf{supp}([P]_\alpha)$ holds because we have a well-defined equivariant projection from paths to their initial states. The converse inclusion is shown by induction, using Lemma 3.4.

In the proof that $\tilde{A}$ accepts $L_\alpha(A)$, the following normalization result for paths is crucial.

▶ **Definition A.18.** A path in $\tilde{A}$ is *π-literal* for $\pi \in G$ if all transitions in it are of the form $(q, \pi H_q) \xrightarrow{\pi\alpha} (q', \pi H_{q'})$ where $\alpha \in \bar{\mathbb{A}}$ and $q \xrightarrow{\alpha} q'$.

Intuitively, a π-literal path is one that uses the same pattern of name reusage for free and bound names as the underlying path in A, up to a joint renaming π of the free and bound names.

▶ **Lemma A.19.** *Let P be a path in $\tilde{A}$ beginning at $(q_0, \pi_0 H_{q_0})$. Then P is α-equivalent to a π_0-literal path.*

Proof. We prove the statement by induction over the path length. The base case is trivial. For the inductive step, let $P = (q_0, \pi_0 H_{q_0}) \xrightarrow{\alpha_1} (q_1, \pi_1 H_{q_1}) \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} (q_n, \pi_n H_{q_n})$ be a path of length $n > 0$. If α_1 is a free name then $\pi_1 H_{q_1} = \pi_0 H_{q_1}$ by (6); by induction, we can assume that the length-$(n-1)$ path from $(q_1, \pi_0 H_{q_1})$ onward is π_0-literal, and hence the whole path is π_0-literal. If $\alpha_1 = |a$ then by (7) we have $\langle a\rangle(q_1, \pi_1 H_{q_1}) = \langle \pi_0(b)\rangle(q_1, \pi_0 H_{q_1})$ for some transition $q_0 \xrightarrow{|b} q_1$ in A. By Lemma A.16, this induces an α-equivalence of P with a path $(q_0, \pi_0 H_{q_0}) \xrightarrow{|\pi_0(b)} (q_1, \pi_0 H_{q_1}) \to \ldots$; by the induction hypothesis, we can transform the length-$(n-1)$ path from $(q_1, \pi_0 H_{q_1})$ onward into a π_0-literal one, so that the whole path becomes π_0-literal as desired. ◀

▶ **Lemma A.20.** *$\tilde{A}$ is an RNNA, with as many orbits as A has states, and accepts the bar language $L_\alpha(A)$.*

Proof. The free and bound transitions of $\tilde{A}$ are equivariant, and the bound transitions are α-invariant by construction of the transition relation on $\tilde{A}$ (note that all states in the orbit of $(q, \pi H_q)$ have the form $(q, \pi' H_q)$). Every orbit of $\tilde{A}$ contains a state of the form $(q, \mathsf{id} H_q)$. This proves the claim on the number of orbits, which implies that $\tilde{A}$ is orbit-finite. Finite branching is immediate from Lemma A.13. Thus, $\tilde{A}$ is an RNNA.

It remains to show that $L_\alpha(\tilde{A}) = L_\alpha(A)$. The inclusion '$\supseteq$' is clear because A is a subautomaton in $\tilde{A}$ via the inclusion map f taking a state q to $(q, \mathsf{id} H_q)$; i.e. $q \xrightarrow{\alpha} q'$ in A implies $(q, \mathsf{id} H_q) \xrightarrow{\alpha} (q', \mathsf{id} H_{q'})$ in $\tilde{A}$. For the reverse inclusion, note that by Lemma A.19, every accepting path of $\tilde{A}$ is α-equivalent to an id-literal accepting path starting at the initial state $(s, \mathsf{id} H_s)$ of $\tilde{A}$; such a path comes from an accepting path in A for the same bar string via the map f. ◀

We are now set to prove Theorem 4.5 by combining the above construction with that of Lemma 3.8; i.e. we show that $(\tilde{A})'$ is isomorphic, and in fact equal, to $\bar{A}$: A state in $(\tilde{A})'$ has the form

$$(q, \pi H_q)|_N = (\mathsf{Fix}\, N) \cdot (q, \pi H_q) = (q, (\mathsf{Fix}\, N)\pi H_q)$$

for $N \subseteq \mathsf{supp}(q, \pi H_q) = \pi N_q$ (hence $\pi^{-1} N \subseteq N_q$). We claim that

$$(\mathsf{Fix}\, N)\pi H_q = \pi\, \mathsf{Fix}(\pi^{-1} N)(= \pi F_{\pi^{-1}N}). \tag{9}$$

To see '$\subseteq$', let $\rho \in \mathsf{Fix}\, N$ and $\sigma \in H_q$. Then $\pi^{-1}\rho\pi \in \mathsf{Fix}(\pi^{-1}N)$ and, since $\pi^{-1}N \subseteq N_q$, $\sigma \in \mathsf{Fix}(\pi^{-1}N)$, so $\pi^{-1}\rho\pi\sigma \in \mathsf{Fix}(\pi^{-1}N)$ and therefore $\rho\pi\sigma = \pi\pi^{-1}\rho\pi\sigma \in \pi\, \mathsf{Fix}(\pi^{-1}N)$.

For '$\supseteq$', let $\rho \in \mathsf{Fix}(\pi^{-1}N)$. Then $\pi\rho\pi^{-1} \in \mathsf{Fix}(N)$, so to show $\pi\rho \in \mathsf{Fix}(N)\pi H_q$ it suffices to show $(\pi\rho\pi^{-1})^{-1}\pi\rho \in \pi H_q$. But $(\pi\rho\pi^{-1})^{-1}\pi\rho = \pi \in \pi H_q$.

This proves equality of the state sets. It remains to show that the transitions in $(\tilde{A})'$ and $\bar{A}$ are the same. The free transitions in $(\tilde{A})'$ are of the form $(q, \pi H_q)|_N \xrightarrow{\pi(a)} (q', \pi H_{q'})|_{N'}$ where $q \xrightarrow{a} q'$, $N' \subseteq \pi N_{q'}$, and $N' \cup \{a\} \subseteq N \subseteq \pi N_q$; by (9), they thus have, up to α-equivalence, the form $(q, \pi F_{\pi^{-1}N}) \xrightarrow{\pi(a)} (q', \pi F_{\pi^{-1}N'})$ where $\pi^{-1}N' \subseteq N_{q'}$ $\pi^{-1}N' \cup \{a\} \subseteq \pi^{-1}N \subseteq N_q$, and hence are the same as in $\bar{A}$.

The bound transitions in $(\tilde{A})'$ are, up to α-equivalence, those of the form $(q, \pi H_q)|_N \xrightarrow{|\pi(a)} (q', \pi H_{q'})|_{N'}$ where $q \xrightarrow{|a} q'$, $N' \subseteq \{a\} \cup N$, $N' \subseteq \pi N_{q'}$; and $N \subseteq \pi N_q$; by (9), they thus have the form $(q, \pi F_{\pi^{-1}N}) \xrightarrow{|\pi(a)} (q', \pi F_{\pi^{-1}N'})$ where $\pi^{-1}N' \subseteq \{a\} \cup \pi^{-1}N$, $\pi^{-1}N' \subseteq N_{q'}$, and $\pi^{-1}N \subseteq N_q$, and hence again are the same as in $\bar{A}$. ◀

Proof of Theorem 4.7

We have to show that every accepting path in A is α-equivalent to an accepting path in A_0. Note that Q_0 is closed under free transitions in A, so by Lemma A.16, it suffices to show that for every bound transition $q \xrightarrow{|b} q'$ in A with $q \in Q_0$ we find an α-equivalent transition $q \xrightarrow{|a} q''$ in A_0. We distinguish the following cases.

- If already $b \in \mathbb{A}_0$ then $\mathsf{supp}(q') \subseteq \mathsf{supp}(q) \cup \{b\} \subseteq \mathbb{A}_0$, so $q' \in Q_0$ and we are done.
- If $b \notin \mathbb{A}_0$ and $b \notin \mathsf{supp}(q')$ then $\mathsf{supp}(q') \subseteq \mathsf{supp}(q) \subseteq \mathbb{A}_0$. In particular, q' is already in Q_0 and $*$ is fresh for q', so we can rename b into $*$ and obtain an α-equivalent transition $q \xrightarrow{|*} q'$ in A_0.
- If $b \notin \mathbb{A}_0$ and $b \in \mathsf{supp}(q')$ then $|\mathsf{supp}(q') \cap \mathbb{A}_0| < k$, so that we can pick a name $a \in \mathbb{A}_0$ that is fresh for q'. We put $q'' = (ab)q'$; then $\langle b \rangle q' = \langle a \rangle q''$, and $q'' \in Q_0$ because $\mathsf{supp}(q'') = \{a\} \cup (\mathsf{supp}(q') - \{b\}) \subseteq \{a\} \cup \mathsf{supp}(q) \subseteq \mathbb{A}_0$; thus, $q \xrightarrow{|a} q''$ is a transition in A_0.

A.4 Proofs and Lemmas for Section 5

▶ **Lemma A.21.** *If $\pi \in G$ and $q|_N$ restricts a state q in a NOFA to $N \subseteq \mathsf{supp}(q)$, then $\pi(q|_N)$ restricts πq to πN.*

Proof. We have $\pi N \subseteq \pi\mathsf{supp}(q) = \mathsf{supp}(\pi q)$, so the claim is well-formed.

For the support of $\pi(q|_N)$, we have $\mathsf{supp}(\pi(q|_N)) = \pi\mathsf{supp}(q|_N) = \pi N$ as required.

By the equivariance of final states we have that $\pi(q|_N)$ is final if q is final.

For incoming transitions, let $p \xrightarrow{a} \pi q$. Then $\pi^{-1}p \xrightarrow{\pi^{-1}a} q$ by equivariance, hence $\pi^{-1}p \xrightarrow{\pi^{-1}a} q|_N$ so that $p \xrightarrow{a} \pi(q|_N)$.

For outgoing transitions, let $\pi q \xrightarrow{a} q'$ where $\mathsf{supp}(q') \subseteq \pi N \cup \{a\}$. Then $q \xrightarrow{\pi^{-1}a} \pi^{-1}q'$ by equivariance, and $\mathsf{supp}(\pi^{-1}q') \subseteq N \cup \pi^{-1}a$, so $q|_N \xrightarrow{\pi^{-1}a} \pi^{-1}q'$ and hence, by equivariance, $\pi(q|_N) \xrightarrow{a} q'$. ◀

Proof of Proposition 5.3. In the first claim, 'only if' is immediate by Lemma 3.4. To see 'if', let A be a non-spontaneous and α-invariant NOFA. We construct an RNNA B with the same states as A, as follows.

- $q \xrightarrow{a} q'$ in B iff $q \xrightarrow{a} q'$ in A and $a \in \mathsf{supp}(q)$.
- $q \xrightarrow{|a} q'$ in B iff $q \xrightarrow{b} q''$ in A for some b, q'' such that $b \# q$ and $\langle b\rangle q'' = \langle a\rangle q'$.

The transition relation thus defined is clearly equivariant and α-invariant. That for every q the sets $\{(a, q') \mid q \xrightarrow{a} q'\}$ and $\{\langle a\rangle q' \mid q \xrightarrow{|a} q'\}$ are ufs (whence finite) easily follows from non-spontaneity.

It remains to verify that $D(B) = A$, i.e. that

$$q \xrightarrow{a} q' \text{ in } A \quad \text{iff} \quad (q \xrightarrow{a} q' \text{ or } q \xrightarrow{|a} q' \text{ in } B).$$

To see the 'only if' direction, let $q \xrightarrow{a} q'$ in A. If $a \in \mathsf{supp}(q)$ then $q \xrightarrow{a} q'$ in B. Otherwise, $a \# q$ and hence $q \xrightarrow{|a} q'$. For the 'if' direction, we have two cases; the case where $q \xrightarrow{a} q'$ in B is immediate by construction of B. So let $q \xrightarrow{|a} q'$ in B, that is, we have $q \xrightarrow{b} q''$ in A for some b, q'' such that $\langle b\rangle q'' = \langle a\rangle q'$ and $b \# q$. Then by α-invariance of A, $q \xrightarrow{a} q'$.

We proceed to prove the second claim, beginning with 'only if'. So let C be a name-dropping RNNA, let q be a state, let $N \subseteq \mathsf{supp}(q)$, and let $q|_N$ restrict q to N in C. We show that $q|_N$ restricts q to N in $D(C)$. The condition $\mathsf{supp}(q|_N) \subseteq N$ is clear, as the nominal set of states is not changed by D. Since $q|_N$ has at least the same incoming transitions as q in C, the same holds in $D(C)$. For the outgoing transitions, first let $q \xrightarrow{a} q'$ in $D(C)$ where $a \in \mathsf{supp}(q)$ and $\mathsf{supp}(q') \cup \{a\} \subseteq N$. Then either $q \xrightarrow{a} q'$ or $q \xrightarrow{|a} q'$ in C. In the first case, $q|_N \xrightarrow{a} q'$ in C and hence also in $D(C)$. In the second case, we have $\mathsf{supp}(q') \subseteq N \subseteq \{a\} \cup N$ and therefore $q|_N \xrightarrow{|a} q'$ in C, so $q|_N \xrightarrow{a} q'$ in $D(C)$. Second, let $q \xrightarrow{a} q'$ in $D(C)$ where $a \# q$ and $\mathsf{supp}(q') \subseteq N \cup \{a\}$. By Lemma 3.4.1 we know that $q \xrightarrow{|a} q'$ in C, so $q|_N \xrightarrow{|a} q'$ in C and hence $q|_N \xrightarrow{a} q'$ in $D(C)$.

For the 'if' direction of the second claim, let A be a non-spontaneous, name-dropping, and α-invariant NOFA. We construct B such that $D(B) = A$ as for the first claim, and show additionally that B is name-dropping. Let q be a state, let $N \subseteq \mathsf{supp}(q)$, and let $q|_N$ restrict q to N in A. We claim that $q|_N$ also restricts q to N in B. We first show that $q|_N$ has at least the same incoming transitions as q in B. For the free transitions, let $p \xrightarrow{a} q$ in B. Then by construction of B, $p \xrightarrow{a} q$ in A and $a \in \mathsf{supp}(p)$, so since A is name-dropping, $p \xrightarrow{a} q|_N$ in A and hence $p \xrightarrow{a} q|_N$ in B. For the bound transitions, let $p \xrightarrow{|a} q$ in B, i.e. we have $p \xrightarrow{b} q'$ in A with $b \# p$ and $\langle a\rangle q = \langle b\rangle q'$, in particular $q' = (ab)q$. If $a = b$ then $p \xrightarrow{a} q$ in A, and since A is name-dropping $p \xrightarrow{a} q|_N$ in A whence in B. Otherwise, $b \# q$. By Lemma A.21, $(ab)(q|_N)$ restricts q' to $(ab)N$ in A, so $p \xrightarrow{b} (ab)(q|_N)$ in A.

Since $\mathsf{supp}(q|_N) \subseteq \mathsf{supp}(q)$, we have $b \# q|_N$, so $\langle b\rangle((ab)(q|_N)) = \langle a\rangle(q|_N)$. By construction of B we have $q \xrightarrow{|a} q|_N$ in B, as required.

For the outgoing transitions, first let $q \xrightarrow{a} q'$ in B where $\mathsf{supp}(q') \cup \{a\} \subseteq N$. Then $q \xrightarrow{a} q'$ in $A = D(B)$, so $q|_N \xrightarrow{a} q'$ in A; since $a \in N = \mathsf{supp}(q|_N)$, it follows by construction of B that $q|_N \xrightarrow{a} q'$ in B. Second, let $q \xrightarrow{|a} q'$ in B where $\mathsf{supp}(q') \subseteq N \cup \{a\}$. Pick $b \# (q, a)$ (so $b \notin N$); then $\langle b\rangle((ab)q') = \langle a\rangle q'$ and therefore $q \xrightarrow{|b} (ab)q'$ in B by α-invariance. Thus, $q \xrightarrow{b} (ab)q'$ in $A = D(B)$, and $\mathsf{supp}((ab)q') = (ab)\mathsf{supp}(q') \subseteq (ab)(N \cup \{a\}) = N - \{a\} \cup \{b\} \subseteq N \cup \{b\}$, where the last but one equation holds since $b \notin N$. Therefore, $q|_N \xrightarrow{b} (ab)q'$ in A since A is name-dropping. By construction of B, it follows that $q|_N \xrightarrow{|a} q'$ in B. ◀

Proof of Proposition 5.4. Let A be a non-spontaneous and name-dropping NOFA. We construct a NOFA $\bar{A}$ by closing A under α-equivalence of transitions; that is, $\bar{A}$ has the same states as A (in particular is orbit-finite), and its transitions are given by

$q \xrightarrow{a} q'$ in $\bar{A}$ iff $q \xrightarrow{a} q'$ in A or

there exist b, q'' such that $q \xrightarrow{b} q''$ in $A, b \# q,$ and $\langle a\rangle q' = \langle b\rangle q''$.

We say that a transition $q \xrightarrow{a} q'$ in $\bar{A}$ is *new* if it is not in A.

► **Fact A.22.** *If $q \xrightarrow{a} q'$ is new then $a \in \mathsf{supp}(q)$ and there exist b, q'' such that $q \xrightarrow{b} q''$ in A, $b \# q$ (so $a \neq b$), and $\langle a\rangle q' = \langle b\rangle q''$.*

We check that $\bar{A}$ has the requisite properties. First, the transition relation is clearly equivariant. Moreover, $\bar{A}$ is α-invariant by construction.

$\bar{A}$ is non-spontaneous: It suffices to check new transitions $q \xrightarrow{a} q'$. By Fact A.22, we have $a \in \mathsf{supp}(q)$ and b, q'' such that $q \xrightarrow{b} q''$ in A, $b \# q$, and $\langle a\rangle q' = \langle b\rangle q''$. Since A is non-spontaneous, $\mathsf{supp}(q'') \subseteq \mathsf{supp}(q) \cup \{b\}$. Let $c \in \mathsf{supp}(q')$ and $c \neq a$; we have to show $c \in \mathsf{supp}(q)$. Now $q' = (ab) \cdot q''$, so $\mathsf{supp}(q') = (ab) \cdot \mathsf{supp}(q'') \subseteq (ab) \cdot (\mathsf{supp}(q) \cup \{b\})$. Since $\langle a\rangle q' = \langle b\rangle q''$, we have $b \# q'$, so $c \notin \{a, b\}$; thus, $c \in (ab) \cdot (\mathsf{supp}(q) \cup \{b\})$ implies $c \in \mathsf{supp}(q) \cup \{a\}$, hence $c \in \mathsf{supp}(q)$.

$\bar{A}$ is name-dropping: Let $N \subseteq \mathsf{supp}(q)$ for a state q, and let $q|_N$ restrict q to N in A; we show that $q|_N$ also restricts q to N in $\bar{A}$. The support of $q|_N$ stays unchanged in $\bar{A}$, so we only have to check that $q|_N$ retains the requisite transitions. Throughout, it suffices to check new transitions.

For incoming transitions, let $p \xrightarrow{a} q$ in $\bar{A}$ be new, i.e. by Fact A.22 we have $a \in \mathsf{supp}(p)$, $p \xrightarrow{b} q'$ in A, $b \# p$ (hence $a \neq b$), and $\langle b\rangle q' = \langle a\rangle q$. Then $(ab) \cdot q = q'$. Therefore, $(ab) \cdot (q|_N)$ restricts q' to $(ab) \cdot N$ in A by Lemma A.21. It follows that $p \xrightarrow{b} (ab) \cdot (q|_N)$ in A. Since $a \neq b$ and $\langle b\rangle q' = \langle a\rangle q$, we have $a \# q'$ and therefore $a \# ((ab) \cdot (q|_N))$, so $\langle b\rangle((ab) \cdot (q|_N)) = \langle a\rangle(q|_N)$ and therefore $p \xrightarrow{a} q|_N$ by construction of $\bar{A}$.

For outgoing transitions, let $q \xrightarrow{a} q'$ be new in $\bar{A}$; i.e. by Fact A.22 we have $a \in \mathsf{supp}(q)$, $q \xrightarrow{b} q''$ in A, $b \# q$ (hence $a \neq b$) and $\langle b\rangle q'' = \langle a\rangle q'$. Since $a \in \mathsf{supp}(q)$, we have to show that $q|_N \xrightarrow{a} q'$ in $\bar{A}$, assuming $\mathsf{supp}(q') \cup \{a\} \subseteq N$. From $b \# q$ we have $b \# q|_N$, so by construction of $\bar{A}$, it suffices to show $q|_N \xrightarrow{b} q''$ in A, which will follow once we show $\mathsf{supp}(q'') \subseteq N \cup \{b\}$. So let $b \neq c \in \mathsf{supp}(q'')$; we have to show $b \in N$. Now $a \neq b$ and $\langle b\rangle q'' = \langle a\rangle q'$ imply $a \# q''$, so $c \neq a$ and hence $c \notin \{a, b\}$. Therefore $c \in (ab) \cdot \mathsf{supp}(q'') = \mathsf{supp}((ab) \cdot q'') = \mathsf{supp}(q') \subseteq N$, as required.

$\bar{A}$ is equivalent to A: $L(A) \subseteq L(\bar{A})$ is immediate as $A \subseteq \bar{A}$ by construction. For the reverse inclusion, we show that[2]

---

[2] For greater clarity we write $L(A, q)$ for $L(q)$ where q is a state in A.

($*$) whenever $w \in L(\bar{A}, q)$ then there exists $N \subseteq \mathsf{supp}(q)$ such that if $q|_N$ restricts q to N in A then $w \in L(A, q|_N)$

(in fact, N will be such that $|\mathsf{supp}(q) - N| \leq 1$). Since $\mathsf{supp}(s) = \emptyset$ for the initial state s, this implies that $L(\bar{A}) \subseteq L(A)$.

We prove ($*$) by induction on w, with trivial induction base. So let $w = av$ and $q \xrightarrow{a} q'$ in $\bar{A}$ where $v \in L(\bar{A}, q')$. By induction, there is $N \subseteq \mathsf{supp}(q')$ such that $v \in L(A, q'|_N)$ whenever $q'|_N$ restricts q' to N in A. If $q \xrightarrow{a} q'$ in A then $q \xrightarrow{a} q'|_N$ in A, so that $av \in L(A, q)$. The remaining case is that $q \xrightarrow{a} q'$ is new. By Fact A.22, we have $a \in \mathsf{supp}(q)$ and b, q'' such that $q \xrightarrow{b} q''$ in A, $b \# q$ (so $a \neq b$), and $\langle a \rangle q' = \langle b \rangle q''$. We claim that whenever $q|_{N_a}$ restricts q to $N_a := \mathsf{supp}(q) - \{a\}$ in A then $av \in L(A, q|_{N_a})$. It suffices to show

$$q|_{N_a} \xrightarrow{a} q'|_N \text{ in } A. \tag{10}$$

Since $a \neq b$ and $\langle a \rangle q' = \langle b \rangle q''$, we have $a \# q''$ so from $q \xrightarrow{b} q''$ in A we obtain $\mathsf{supp}(q'') \subseteq \{b\} \cup N_a$ by non-spontaneity of A. By the definition of restriction, it follows that $q|_{N_a} \xrightarrow{b} q''$ in A (recall that $b \# q$). Since $a \notin \mathsf{supp}(q|_{N_a}) = N_a$, we obtain by equivariance of transitions that $q|_{N_a} \xrightarrow{a} q'$, which implies (10) by the definition of restriction: we have $q' = (ab) \cdot q''$ which implies

$$\mathsf{supp}(q') = (ab) \cdot \mathsf{supp}(q'') \subseteq \{a\} \cup (ab) \cdot N_a = \{a\} \cup N_a,$$

where the last step holds since $a, b \notin N_a$. ◀

Additional proof details for Corollary 5.6. It is straightforward to verify that non-spontaneous name-dropping NOFAs are closed under the standard product construction; specifically, given a state (q_1, q_2) in a product automaton and $N \subseteq \mathsf{supp}(q_1, q_2) = \mathsf{supp}(q_1) \cup \mathsf{supp}(q_2)$, one checks readily that if $q_i|_{N_i}$ restricts q_i to $N_i := N \cap \mathsf{supp}(q_i)$ for $i = 1, 2$, then $(q_1|_{N_1}, q_2|_{N_2})$ restricts (q_1, q_2) to N. ◀

Translation of FSUBAs into RNNAs

Let A be an FSUBA with set Q of state, set r of registers, initial state q_0, set F of final states, transition relation $\mu \subseteq Q \times r \times \mathcal{P}_\omega(r) \times Q$, and initial register assignment u; as indicated in Section 5, we restrict the read-only alphabet Θ to be empty. We denote the $\mathbb{A}$-language accepted by A by $L(A)$. We construct an equivalent RNNA $R(A)$ as follows. The states of $R(A)$ are the configurations of A, which form a nominal set C under the group action $\pi \cdot (q, v) = (q, \pi \cdot v)$. The transitions of $R(A)$ are given by

$$\mathsf{fsuc}(q, v) = \{(v(k), (p, \mathsf{erase}_S(v))) \mid (q, k, S, p) \in \mu\} \tag{11}$$
$$\cup\ \{(a, (p, \mathsf{erase}_S(v[k \mapsto a]))) \mid (q, k, S, p) \in \mu, a \in \mathsf{supp}(v), v(k) = \bot\} \tag{12}$$
$$\mathsf{bsuc}(q, v) = \{\langle a \rangle (p, \mathsf{erase}_S(v[k \mapsto a])) \mid (q, k, S, p) \in \mu, a \# v, v(k) = \bot\} \tag{13}$$

where erase_S clears the contents of the registers in S.

This RNNA $R(A)$ behaves, under local freshness semantics, like the FSUBA A:

▶ **Lemma A.23.** *The transitions between configurations of A are precisely given by $(q, v) \xrightarrow{\mathsf{ub}(\alpha)} (p, w)$, where $(q, v) \xrightarrow{\alpha} (p, w)$, $\alpha \in \bar{\mathbb{A}}$, is a transition in $R(A)$.*

Proof. Let $(q, v) \xrightarrow{\alpha} (p, w)$ be a transition in the RNNA $R(A)$. We distinguish cases:

- For (11), we have an FSUBA transition $(q, k, S, p) \in \mu$ with $\alpha = v(k) \in \mathbb{A}$, and $w = \mathsf{erase}_S(v)$. Hence we have a transition $(q, v) \xrightarrow{\mathsf{ub}(\alpha)} (p, w)$ between FSUBA configurations.

- For (12), we have an FSUBA transition $(q, k, S, p) \in \mu$ and $v(k) = \bot$, $\alpha = v(i) \in \mathbb{A}$ for some $i \in r$, and $w = \mathsf{erase}_S(v[k \mapsto v(i)])$. Hence, from the FSUBA configuration (q, v) the input $v(i)$ is read into register k and then the registers in S are cleared, i.e. $(q, v) \xrightarrow{\alpha} (p, w)$ is a transition of FSUBA configurations.
- For (13), i.e. for $\alpha = |a$, we have an FSUBA transition $(q, k, S, p') \in \mu$ and $v(k) = \bot$ and some $b \# v$ with $\langle a\rangle(p, w) = \langle b\rangle(p', \mathsf{erase}_S(v[k \mapsto b]))$. It follows that $(ab)(p, w) = (p', \mathsf{erase}_S(v[k \mapsto b]))$, and equivalently, $p = p'$ and $(ab)w = \mathsf{erase}_S(v[k \mapsto b])$. The latter implies that $w = \mathsf{erase}_S(v[k \mapsto a])$. Thus, we obtain a transition of FSUBA configurations $(q, v) \xrightarrow{a} (p, w)$ as desired.

Conversely, consider a transition $(q, v) \xrightarrow{a} (p, w)$ of FSUBA configurations admitted by $(q, k, S, p) \in \mu$.

- If $v(k) \neq \bot$, then $v(k) = a$. Hence $(q, v) \xrightarrow{a} (p, w)$ is a transition in $R(A)$ by (11).
- If $v(k) = \bot$ and $a \in \mathsf{supp}(v)$, then $(q, v) \xrightarrow{a} (p, w)$ is a transition in $R(A)$ by (12).
- If $v(k) = \bot$ and $a \# v$, then $w = \mathsf{erase}_S(v[k \mapsto a])$ and $\langle a\rangle(p, w) \in \mathsf{bsuc}(q, v)$. By α-invariance, this implies $(q, v) \xrightarrow{|a} (p, w)$ in $R(A)$. ◀

Using Lemma A.23, one shows by induction on w that $L(A) = \{\mathsf{ub}(w) \mid w \in L_0(R(A))\}$. The RNNA $R(A)$ in general fails to be name-dropping, but for any $[|aw]_\alpha \in L_\alpha(q, v)$, $w \in \bar{\mathbb{A}}^*$, we have

$$(q, v) \xrightarrow{|a} (p, v'), w \in L_\alpha(p, v') \quad \text{or} \quad (q, v) \xrightarrow{a} (p, v'), w \in L_\alpha(p, v') : \tag{14}$$

Since $[|aw]_\alpha \in L_\alpha(q, v)$, we have some transition $(q, v) \xrightarrow{|b} (p', v'')$ in $R(A)$ such that $\langle a\rangle w = \langle b\rangle w'$ for some $w' \in L_\alpha(p', v'')$; if we cannot α-equivalently rename the $|b$-transition into an $|a$-transition to obtain the left alternative in (14), then $b \neq a \in \mathsf{supp}(v'')$ and hence $a \in \mathsf{supp}(v)$, so by construction of $R(A)$ we obtain the right alternative in (14). By induction on w, it follows that $\{\mathsf{ub}(w) \mid w \in L_0(q_0, u)\} = D(L_\alpha(q_0, u))$, so that $L(A) = D(L_\alpha(R(A)))$, as claimed. ◀

Details for Remark 5.7 We show that the data language

$$L = \{wava \mid w, v \in \mathbb{A}^*, a \in \mathbb{A}\}$$

is not accepted by any DOFA. Assume for a contradiction that A is a DOFA that accepts L. Let n be the maximal size of a support of a state in A. Let $w = a_1 \dots a_{n+1}$ for distinct a_i, and let q be the state reached by A after consuming w. Then there is $i \in \{1, \dots, n+1\}$ such that $a_i \notin \mathsf{supp}(q)$. Pick a fresh name b. Then $\delta(a_i, q)$ is final and $\delta(b, q)$ is not; but since $\delta(a_i, q) = (a_i b) \cdot \delta(b, q)$, this is in contradiction to equivariance of the set of final states. ◀

A.5 Proofs and Lemmas for Section 6

Additional details for the proof of Theorem 6.1. We have omitted the space analysis of the initialization step. To initialize Ξ we need to compute $N_2 = \mathsf{supp}(L_\alpha(s_2))$. This can be done in nondeterministic logspace: for every free transition $q \xrightarrow{a} q'$ in A_2, in order to decide whether or not $a \in N_{s_2}$, remove from the transition graph of A_2 all transitions with label $|a$ and then check whether there exists a path from s_2 to a final state passing through the given transition. ◀

Details for Remark 6.2 The spines of an NKA expression r arise by α-renaming and subsequent deletion of some binders from expressions that consist of subexpressions of r, prefixed by at most as many binders as occur already in r; therefore, the degree of the RNNA formed by the spines, and hence, by Theorem 4.7 (and the fact that the translation from bar NFA to regular bar expressions is polynomial and preserves the degree), that of the arising regular bar expression, is linear in the degree of r (specifically, at most twice as large). ◀

We shortly write $D(w) = D(L_\alpha(w)) = \{\mathsf{ub}(w') \mid w' \equiv_\alpha w\}$ for $w \in \bar{\mathbb{A}}^*$.

▶ **Lemma A.24.** *If $w \sqsubseteq w'$ then $D(w) \subseteq D(w')$.*

Proof. Induction over w, with trivial base case. The only non-trivial case in the induction step is that $w = av$ and $w' = \mathsf{l}av'$ where $v \sqsubseteq v'$. All bar strings that are α-equivalent to w have the form au where $v \equiv_\alpha u$; we have to show $\mathsf{ub}(au) \in D(\mathsf{l}av')$. We have $\mathsf{ub}(u) \in D(v)$, so $\mathsf{ub}(u) \in D(v')$ by induction; that is, there exists $\bar{v}' \equiv_\alpha v'$ such that $\mathsf{ub}(\bar{v}') = \mathsf{ub}(u)$. Then $\mathsf{ub}(\mathsf{l}a\bar{v}') = \mathsf{ub}(au)$ and $\mathsf{l}a\bar{v}' \equiv_\alpha \mathsf{l}av'$, so $au \in D(\mathsf{l}av')$. ◀

Lemma 6.4 is immediate from the following:

▶ **Lemma A.25.** *Let L be a regular bar language, and let $w \in \bar{\mathbb{A}}^*$. Then $D(w) \subseteq D(L)$ iff there exists $w' \sqsupseteq w$ such that $[w']_\alpha \in L$.*

Proof. *'If':* If $[w']_\alpha \in L$ then $D(w') \subseteq D(L)$, so $D(w) \subseteq D(L)$ by Lemma A.24.

'Only if': We generalize the claim to state that whenever

$$D(w) \subseteq \bigcup_{i \in I} D(L_\alpha(q_i))$$

for states q_i in a name-dropping RNNA A and a finite index set I, then there exist i and $w' \sqsupseteq w$ such that $[w']_\alpha \in L_\alpha(q_i)$.

We prove the generalized claim by induction over w. The base case is trivial.

Induction step for words aw: Let $D(aw) \subseteq \bigcup_{i=1}^n D(L_\alpha(q_i))$. We prove below that

$$D(w) \subseteq \bigcup_{i \in I, q_i \xrightarrow{\alpha} q', \alpha \in \{a, \mathsf{l}a\}} D(L_\alpha(q')). \tag{15}$$

Indeed, let $u \in D(w)$, i.e. there exists $v \equiv_\alpha w$ with $\mathsf{ub}(v) = u$. Then $\mathsf{ub}(av) = au$ and $av \equiv_\alpha aw$ imply $au \in D(aw)$, so by assumption there exists $i \in \{1, \dots, n\}$ such that $au \in D(L_\alpha(q_i))$, i.e. $au = \mathsf{ub}(\alpha\bar{u})$ for $\alpha \in \{a, \mathsf{l}a\}$ and $[\alpha\bar{u}]_\alpha \in L_\alpha(q_i)$. By Lemma 3.7, $\alpha\bar{u} \in L_0(q_i)$. Therefore there exists a transition $q \xrightarrow{\alpha} q'$ and $\bar{u} \in L_0(q')$. We conclude that $u = \mathsf{ub}(\bar{u}) \in D(L_\alpha(q'))$ as desired.

Now, by induction hypothesis, it follows from (15) that we have $i \in I$, $\alpha \in \{a, \mathsf{l}a\}$, $q_i \xrightarrow{\alpha} q'$, and $w' \sqsupseteq w$ such that $[w']_\alpha \in L_\alpha(q')$. Then $\alpha w' \sqsupseteq aw$ and $[\alpha w']_\alpha \in L_\alpha(q_i)$, as required.

Induction step for words $\mathsf{l}aw$: Let $D(\mathsf{l}aw) \subseteq \bigcup_{i=1}^n D(L_\alpha(q_i))$. Notice that

$$D(\mathsf{l}aw) = \bigcup_{b = a \vee b \# [w]_\alpha} bD(\pi_{ab} \cdot w)$$

(where $\cdot$ denotes the permutation group action and $\pi_{ab} = (a\ b)$ the transposition of a and b; also note that $b \# [w]_\alpha$ iff $b \notin \mathsf{FN}(w)$). Now pick $b \in \mathbb{A}$ such that $b \# [w]_\alpha$ and none of the q_i has a b-transition (such a b exists because the set of free transitions of each q_i is finite, as A is an RNNA). We prove below that

$$bD(\pi_{ab} \cdot w) \subseteq \bigcup_{i \in I, q_i \xrightarrow{\mathsf{l}b} q'} bD(L_\alpha(q')),$$

and hence we have

$$D(\pi_{ab} \cdot w) \subseteq \bigcup_{i \in I, q_i \xrightarrow{\mathsf{l}b} q'} D(L_\alpha(q')), \tag{16}$$

again a finite union. In order to see that the above inclusion holds, let $bu \in bD(\pi_{ab} \cdot w)$, i.e., we have $v \equiv_\alpha \pi_{ab} \cdot w$ with $\mathsf{ub}(v) = u$. Then $|bv \equiv_\alpha |b(\pi_{ab} \cdot w) \equiv_\alpha |aw$ and $\mathsf{ub}(|bv) = bu$, which implies that $bu \in D(|aw)$. By our assumption $D(|aw) \subseteq \bigcup_{i=1}^{n} D(L_\alpha(q_i))$ we obtain $i \in \{1, \dots, n\}$ such that $bu \in D(L_\alpha(q_i))$, i.e. $bu = \mathsf{ub}(\beta\bar{u})$ for $\beta \in \{b, |b\}$ and $[\beta\bar{u}]_\alpha \in L_\alpha(q_i)$. By Lemma 3.7, we have $\beta\bar{u} \in L_0(q_i)$, and since q_i has no b-transitions, we therefore know that $\beta = |b$. Hence we have a transition $q_i \xrightarrow{|b} q'$ and $\bar{u} \in L_0(q')$. It follows that $u = \mathsf{ub}(\bar{u}) \in D(L_\alpha(q'))$, whence $bu \in dD(L_\alpha(q'))$ as desired.

Now, by induction hypothesis, we obtain from (16) $i \in I$, $q_i \xrightarrow{|b} q'$, and $w' \sqsupseteq \pi_{ab} \cdot w$ such that $[w']_\alpha \in L_\alpha(q')$. It follows that

$$|bw' \sqsupseteq |b(\pi_{ab} \cdot w) \quad \text{and} \quad [|bw']_\alpha \in L_\alpha(q_i).$$

Now we have $a \# [\pi_{ab} \cdot w]_\alpha$ (because $b \# [w]_\alpha$)), and therefore $a \# [w']_\alpha$ because $\pi_{ab} \cdot w \sqsubseteq w'$; it follows that $|a(\pi_{ab} \cdot w') \equiv_\alpha |bw'$. As $\sqsubseteq$ is clearly equivariant, we have $\pi_{ab} \cdot w' \sqsupseteq w$, so

$$|a(\pi_{ab} \cdot w') \sqsupseteq |aw \quad \text{and} \quad [|a(\pi_{ab} \cdot w')]_\alpha = [|bw']_\alpha \in L_\alpha(q_i),$$

which proves the inductive claim. ◀