Positive Data Languages

Florian Frank
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stefan Milius
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

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

Abstract

Positive data languages are languages over an infinite alphabet closed under possibly non-injective renamings of data values. Informally, they model properties of data words expressible by assertions about equality, but not inequality, of data values occurring in the word. We investigate the class of positive data languages recognizable by nondeterministic orbit-finite nominal automata, an abstract form of register automata introduced by Bojańczyk, Klin, and Lasota. As our main contribution we provide a number of equivalent characterizations of that class in terms of positive register automata, monadic second-order logic with positive equality tests, and finitely presentable nondeterministic automata in the categories of nominal renaming sets and of presheaves over finite sets.

2012 ACM Subject Classification Theory of Computation → Formal languages

Keywords and phrases Data Languages, Register Automata, MSO, Nominal Sets, Presheaves

Digital Object Identifier 10.4230/LIPIcs...

Funding *Florian Frank*: Supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Research and Training Group 2475 "Cybercrime and Forensic Computing" (grant number 393541319/GRK2475/1-2019).
Stefan Milius: Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 419850228.
Henning Urbat: Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 470467389.

1 Introduction

Automata over infinite alphabets provide a simple computational model for reasoning about structures involving *data* such as nonces [23], URLs [4], or values in XML documents [28]. Consider, for instance, the (infinite) set $\mathbb{A}$ of admissible user IDs for a server. The sequence of all user logins within a given time period then forms a finite word $a_1 \cdots a_n \in \mathbb{A}^\star$ over the infinite alphabet $\mathbb{A}$, and behaviour patterns may be modelled as data languages over $\mathbb{A}$, e.g.

$$L_0 = \{\, a_1 \cdots a_n \in \mathbb{A}^\star \mid a_i \neq a_n \text{ for all } i < n \,\} \qquad \text{("last user has not logged in before")},$$
$$L_1 = \{\, a_1 \cdots a_n \in \mathbb{A}^\star \mid a_i = a_j \text{ for some } i \neq j \,\} \qquad \text{("some user has logged in twice")}.$$

Both L_0 and L_1 involve assertions about equality, or inequality, of data values (here, user IDs). However, asserting *inequality* is sometimes considered problematic and thus undesired. For example, since users may have multiple IDs, a logfile $a_1 \dots a_n \in L_0$ does not actually guarantee that the last user has not logged in before. In contrast, if $a_1 \dots a_n \in L_1$, then it is guaranteed that some user has indeed logged in twice. The structural difference between the two languages is that L_1 is closed under arbitrary renamings $\rho\colon \mathbb{A} \to \mathbb{A}$ (i.e. $a_1 \cdots a_n \in L_1$ implies $\rho(a_1) \cdots \rho(a_n) \in L_1$), taking into account possible identification of data values, while L_0 is only closed under injective (equivalently bijective) renamings. We refer to

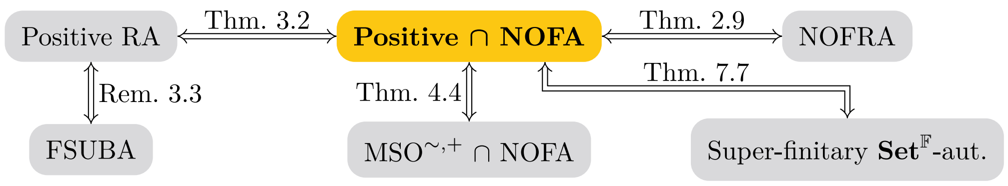

Figure 1 Equivalent characterizations of positive NOFA-recognizable languages

languages with the former, stronger closure property as *positive data languages*. Intuitively, such languages model properties of data words expressible by positive statements about equality of data values. It is one of the goals of our paper to turn this into a theorem.

For that purpose, we build on the abstract account of data languages and their automata based on the theory of nominal sets [15, 30], initiated by the work of Bojańczyk, Klin, and Lasota [7]. Specifically, we investigate *nondeterministic orbit-finite nominal automata* (*NOFA*), the nominal version of classical nondeterministic finite automata. We approach the class of *positive* NOFA-recognizable data languages from several different perspectives, ranging from concrete to more abstract and conceptual, and establish the equivalent characterizations summarized in Figure 1. In more detail, our main contributions are as follows.

Register automata. NOFAs are known to be expressively equivalent to register automata [19, 21], i.e. finite automata that can memorize data values using a fixed number of registers and test the input for (in)equality with previously stored values. Restricting transitions to positive equality tests leads to *positive register automata*, which correspond to *finite-state unification-based automata* (*FSUBA*) [20, 35] and are shown to capture precisely positive NOFA-recognizable languages (Theorem 3.2 and Remark 3.3). On the way, we isolate a remarkable property of this language class: while NOFAs generally require the ability to guess data values during the computation to reach their full expressive strength, guessing and non-guessing NOFA are equivalent for positive data languages (Theorem 2.17).

Monadic second-order logic. As illustrated above, positive data languages model (only) positive assertions about the equality of data values. To substantiate this intuition, we employ monadic second-order logic (MSO$^{\sim}$) over data words [5, 11, 28], an extension of classical MSO with equality tests for data values, and consider its restriction MSO$^{\sim,+}$ to positive equality tests. While this logic is more expressive than NOFA, we show that within the class of NOFA-recognizable languages it models exactly the positive languages (Theorem 4.4).

Categorical perspective. The classical notion of nondeterministic finite automata can be categorified by replacing the finite set of states with a finitely presentable object of a category $\mathscr{C}$. For example, NOFAs are precisely nondeterministic $\mathscr{C}$-automata for $\mathscr{C}$ = nominal sets. Apart from the latter category, several other toposes have been proposed as abstract foundations for reasoning about names (data values), most prominently the category of *nominal renaming sets* [14], the category $\mathbf{Set}^{\mathbb{I}}$ of presheaves over finite sets and injective maps [34], and the category $\mathbf{Set}^{\mathbb{F}}$ of presheaves over finite sets and all maps (equivalently, finitary set functors) [12]. It is thus natural to study nondeterministic automata in the latter three categories, viz. *nondeterministic orbit-finite renaming automata* (*NOFRA*), *nondeterministic super-finitary* $\mathbf{Set}^{\mathbb{I}}$*-automata* and *nondeterministic super-finitary* $\mathbf{Set}^{\mathbb{F}}$*-automata*. Our final contribution is a classification of their expressive power: we show that $\mathbf{Set}^{\mathbb{I}}$-automata are equivalent to NOFAs, while both NOFRAs and $\mathbf{Set}^{\mathbb{F}}$-automata capture positive NOFA-recognizable languages (Theorems 2.9 and 7.7). Hence, both nominal and presheaf-based automata are able to recognize positive and all NOFA-recognizable languages, respectively.

Acknowledgement. The authors wish to thank Bartek Klin for pointing out the example in Remark 2.10.

2 Nominal Automata and Positive Data Languages

For the remainder of the article, we fix a countably infinite set $\mathbb{A}$ of *data values*, a.k.a. *names* or *atoms*. The goal is to study positive data languages, that is, languages of finite words over $\mathbb{A}$ closed under arbitrary renamings. This is achieved via the framework of nominal (renaming) sets [14, 15, 30].

2.1 Nominal Sets and Nominal Renaming Sets

A *renaming* is a finite map $\rho\colon \mathbb{A} \to \mathbb{A}$, that is, $\rho(a) = a$ for all but finitely many $a \in \mathbb{A}$. We let $\mathsf{Fin}(\mathbb{A})$ denote the monoid of renamings, with multiplication given by composition, and $\mathsf{Perm}(\mathbb{A})$ its subgroup given by *finite permutations*, i.e. bijective renamings. For $M \in \{\,\mathsf{Perm}(\mathbb{A}), \mathsf{Fin}(\mathbb{A})\,\}$ an *M-set* is a set X equipped with a monoid action $M \times X \to X$, denoted $(\rho, x) \mapsto \rho \cdot x$. A subset $S \subseteq \mathbb{A}$ is a *support* of $x \in X$ if for every $\rho, \sigma \in M$ such that $\rho|_S = \sigma|_S$ one has $\rho \cdot x = \sigma \cdot x$. Informally, consider X as a set of syntactic objects (e.g. words, trees, λ-terms) whose description may involve free names from S. A *nominal M-set* is an M-set where every element x has a finite support. This implies that x has a least finite support $\mathsf{supp}\, x \subseteq \mathbb{A}$. A name $a \in \mathbb{A}$ is *fresh* for x, denoted $a \,\#\, x$, if $a \notin \mathsf{supp}\, x$.

Nominal $\mathsf{Perm}(\mathbb{A})$-sets are called *nominal sets*, and nominal $\mathsf{Fin}(\mathbb{A})$-sets are called *nominal renaming sets*. A nominal renaming set X can be regarded as a nominal set by restricting its $\mathsf{Fin}(\mathbb{A})$-action to a $\mathsf{Perm}(\mathbb{A})$-action. The least supports of an element $x \in X$ w.r.t. both actions coincide [13, Thm. 4.8], so the notation $\mathsf{supp}\, x$ is unambiguous.

A subset X of a nominal M-set Y is *M-equivariant* if $\rho \cdot x \in X$ for all $x \in X$ and $\rho \in M$. More generally, a map $f\colon X \to Y$ between nominal M-sets is *M-equivariant* if $f(\rho \cdot x) = \rho \cdot f(x)$ for all $x \in X$ and $\rho \in M$. This implies $\mathsf{supp}\, f(x) \subseteq \mathsf{supp}\, x$ for all $x \in X$.

We write $X \times Y$ for the cartesian product of nominal M-sets with componentwise action, and $\coprod_{i \in I} X_i$ for the coproduct (disjoint union) with action inherited from the summands.

Given a nominal set X, the *orbit* of an element $x \in X$ is the set $\{\pi \cdot x : \pi \in \mathsf{Perm}(\mathbb{A})\}$. The orbits form a partition of X. A nominal set is *orbit-finite* if it has only finitely many orbits. A nominal renaming set is *orbit-finite* if it is orbit-finite as a nominal set.

▶ **Example 2.1.** The set $\mathbb{A}$ with the $\mathsf{Fin}(\mathbb{A})$-action $\rho \cdot a = \rho(a)$ is a nominal renaming set, as is the set $\mathbb{A}^\star$ of finite words over $\mathbb{A}$ with $\rho \cdot w = \rho^\star(w) = \rho(a_1) \cdots \rho(a_n)$ for $w = a_1 \cdots a_n$. The least support of $a_1 \cdots a_n \in \mathbb{A}^\star$ is the set $\{a_1, \ldots, a_n\}$. The set $\mathbb{A}^\star$ has infinitely many orbits; its equivariant subsets $\mathbb{A}^n$ (words of a fixed length n) are orbit-finite. For instance, $\mathbb{A}^2$ has the two orbits $\{aa : a \in \mathbb{A}\}$ and $\{ab : a \neq b \in \mathbb{A}\}$. An example of a nominal set that is not a renaming set is $\mathbb{A}^{\#n} = \{\, a_1 \ldots a_n : a_i \neq a_j \text{ for } i \neq j \,\}$ with pointwise $\mathsf{Perm}(\mathbb{A})$-action.

A nominal set X is *strong* if, for every $x \in X$ and $\pi \in \mathsf{Perm}(\mathbb{A})$, one has $\pi \cdot x = x$ if and only if π fixes every element of $\mathsf{supp}(x)$. (The 'if' statement holds in every nominal set.) For instance, the nominal sets $\mathbb{A}^{\#n}$, $\mathbb{A}^n$ and $\mathbb{A}^\star$ are strong. Up to isomorphism, (orbit-finite) strong nominal sets are precisely (finite) coproducts $\coprod_{i \in I} \mathbb{A}^{\#n_i}$ where $n_i \in \mathbb{N}$. For every orbit-finite nominal set X, there exists a surjective $\mathsf{Perm}(\mathbb{A})$-equivariant map $e\colon Y \twoheadrightarrow X$ for some orbit-finite strong nominal set Y (see e.g. [26, Cor. B.27]). In fact, if o is the number of orbits of X, one may take $Y = J \times \mathbb{A}^{\#n}$ where $J = \{1, \ldots, o\}$ and $n = \max_{x \in X} |\mathsf{supp}\, x|$. We refer the reader to [16, Sec. 4.1] and [7, Thm. 10.9] for more details on representing orbit-finite nominal sets.

2.2 Nominal Automata and Nominal Renaming Automata

The object of interest in this paper is data languages $L \subseteq \mathbb{A}^\star$ closed under renamings:

▶ **Definition 2.2.** 1. A data language $L \subseteq \mathbb{A}^\star$ is *positive* if it is $\mathsf{Fin}(\mathbb{A})$-equivariant.
2. The *positive closure* of $L \subseteq \mathbb{A}^\star$ is given by $\overline{L} \;=\; \{\, \rho^\star(w) : w \in L,\ \rho \in \mathsf{Fin}(\mathbb{A}) \,\}$.

A natural automata model for data languages is given by nondeterministic orbit-finite automata [7] over nominal sets and their restriction to nominal renaming sets:

▶ **Definition 2.3.** Let $M \in \{\, \mathsf{Perm}(\mathbb{A}), \mathsf{Fin}(\mathbb{A}) \,\}$.
1. A *nondeterministic orbit-finite M-automaton* $A = (Q, \delta, I, F)$ consists of an orbit-finite nominal M-set Q of states, an M-equivariant transition relation $\delta \subseteq Q \times \mathbb{A} \times Q$, and M-equivariant subsets $I, F \subseteq Q$ of initial and final states. Nominal orbit-finite M-automata are called *nondeterministic orbit-finite automata* (*NOFA*) for $M = \mathsf{Perm}(\mathbb{A})$ and *nondeterministic orbit-finite renaming automata* (*NOFRA*) for $M = \mathsf{Fin}(\mathbb{A})$.
2. Given a nominal orbit-finite M-automaton A, we write $q \xrightarrow{a} q'$ if $q' \in \delta(q, a)$. A *run* of A on input $w = a_1 \cdots a_n \in \mathbb{A}^\star$ is a sequence $(q_0, a_1, q_1, a_2, \ldots, a_n, q_n)$ such that $q_0 \in I$ and $q_r \xrightarrow{a_{r+1}} q_{r+1}$ for $0 \le r < n$. The run is *accepting* if $q_n \in F$. The automaton A *accepts* the word w if A admits an accepting run on input w. The *accepted language* $L(A) \subseteq \mathbb{A}^\star$ is the set of all accepted words. A data language is *NOF(R)A-recognizable* if some NOF(R)A accepts it.

For example, the languages L_0 and L_1 from the Introduction are NOFA-recognizable.

▶ **Remark 2.4.** 1. The restriction to the input alphabet $\mathbb{A}$ is for simplicity: all our results extend to alphabets $\Sigma = \Sigma_0 \times \mathbb{A}$ for a finite set Σ_0, i.e. finite coproducts of copies of $\mathbb{A}$.
2. Another use of nominal renaming sets in automata theory appears in the work by Moerman and Rot [27] on deterministic nominal automata with outputs. The restrictions of their model make it unsuitable for language recognition [27, Rem. 4.1] but allow for a succinct representation of computed maps via *separating automata*.

To relate the expressive power of NOFA and NOFRA, we start with a simple observation:

▶ **Proposition 2.5.** *Every NOFRA accepts a positive language.*

The converse (Theorem 2.9) needs an automata-theoretic construction of the closure of a language. To this end, we first turn the states of a NOFA into a sort of normal form.

▶ **Remark 2.6** (cf. [7]). Every NOFA $A = (Q, \delta, I, F)$ is equivalent to one whose nominal set of states is of the form $J \times \mathbb{A}^{\# m}$ for some finite set J and $m \in \mathbb{N}$. Indeed, choose a nominal set $Q' = J \times \mathbb{A}^{\# m}$ and an equivariant surjection $e\colon Q' \twoheadrightarrow Q$ (see Section 2.1), and consider the NOFA $A' = (Q', \delta', I', F')$ whose structure is given by the preimages

$$\delta' = (e \times \mathsf{id}_{\mathbb{A}} \times e)^{-1}[\delta], \qquad I' = e^{-1}[I], \qquad F' = e^{-1}[F].$$

It is not difficult to verify that $L(A') = L(A)$; see also Proposition 6.9. Note that in a NOFA with states $J \times \mathbb{A}^{\# m}$, the equivariant sets of initial and final states are of the form $I = J_I \times \mathbb{A}^{\# m}$ and $F = J_F \times \mathbb{A}^{\# m}$ for some $J_I, J_F \subseteq J$.

▶ **Construction 2.7** (Positive Closure of a NOFA). Let $A = (Q, \delta, I, F)$ be a NOFA with states $Q = J \times \mathbb{A}^{\# m}$ (cf. Remark 2.6). The NOFRA $\bar{A} = (\bar{Q}, \bar{\delta}, \bar{I}, \bar{F})$ is given by the states $\bar{Q} = J \times \mathbb{A}^m$, initial states $\bar{I} = J_I \times \mathbb{A}^m$, final states $\bar{F} = J_F \times \mathbb{A}^m$, and transitions

$$\bar{\delta} \;=\; \{\, (j, \rho^\star p) \xrightarrow{\rho a} (j', \rho^\star p') : (j, p) \xrightarrow{a} (j', p') \text{ in } A \text{ and } \rho \in \mathsf{Fin}(\mathbb{A}) \,\}.$$

▶ **Proposition 2.8.** *The NOFRA $\bar{A}$ accepts the positive closure of the language of A.*

The proof of $L(\bar{A}) \subseteq \overline{L(A)}$ is slightly subtle since the transitions of a run in $\bar{A}$ may be induced by different ρ's; some bookkeeping and sensible choice of fresh names ensures compatibility.

Now we come to our first characterization of positive NOFA-recognizable languages:

▶ **Theorem 2.9.** *A language is positive and NOFA-recognizable iff it is NOFRA-recognizable.*

Indeed, the "if" direction holds due to Proposition 2.5 and because every NOFRA is a NOFA. The "only if" direction follows from Proposition 2.8, using that $\overline{L} = L$ for positive L.

▶ **Remark 2.10.** A NOF(R)A is *deterministic*, and hence called a *DOF(R)A*, if it admits a single initial state and its transition relation is a function $\delta\colon Q \times \mathbb{A} \to Q$. In contrast to classical finite automata, DOFAs are less expressive that NOFAs [7]. We leave it as an open problem whether Theorem 2.9 restricts to DOF(R)As. In this regard, observe that Construction 2.7 produces a *non*deterministic automaton $\bar{A}$ even if the given automaton A is deterministic. Computing the positive closure of a DOFA-recognizable language necessarily requires the introduction of nondeterminism, as illustrated by the following example due to Bartek Klin (personal communication). Consider the language L consisting of all words whose last letter appears immediately before the last occurrence of a repeated letter; that is, words of the form $vabbwa$ where (i) $v, w \in \mathbb{A}^\star$ and $a, b \in \mathbb{A}$, (ii) any two consecutive letters in w are distinct, (iii) the first letter of w is distinct from b and (iv) the last letter of w is distinct from a. This language is recognizable by a DOFA, in fact by an orbit-finite nominal monoid [5]. Its positive closure $\overline{L}$ consists of all words whose last letter appears immediately before *some* occurrence of a repeated letter, which is not DOFA-recognizable.

2.3 Abstract Transitions and Runs

Sections 3 and 4 will relate positive NOFA-recognizable languages to register automata and monadic second-order logic. This relies on a presentation of transitions of $\bar{A}$ in terms of abstract transitions, given by equations involving register entries and input values.

▶ **Definition 2.11.** Let $A = (Q, \delta, I, F)$ and $\bar{A} = (\bar{Q}, \bar{\delta}, \bar{I}, \bar{F})$ be as in Construction 2.7.

1. An *equation* is an expression of the form $k = \bullet$, $\bullet = k$ or $k = \bar{k}$, where $k, \bar{k} \in \{1, \ldots, m\}$.
2. An *abstract transition* is a triple (j, E, j') where $j, j' \in J$ and E is a set of equations.
3. Every triple $((j,p), a, (j',p')) \in Q \times \mathbb{A} \times Q$ induces an abstract transition (j, E, j') defined as follows for $k, \bar{k} \in \{1, \ldots, m\}$ (we write $(-)_i$ for the i-th letter of a word):
$$k = \bullet \in E \iff p_k = a, \qquad \bullet = k \in E \iff a = p'_k, \qquad k = \bar{k} \in E \iff p_k = p'_{\bar{k}}.$$
We let $\mathsf{abs}(\delta)$ denote the set of abstract transitions induced by transitions in δ, and we write $j \xrightarrow{E} j'$ for $(j, E, j') \in \mathsf{abs}(\delta)$.
4. A triple $((j,q), b, (j',q')) \in \bar{Q} \times \mathbb{A} \times \bar{Q}$ is *consistent* with the abstract transition (j, E, j') if for every $k, \bar{k} \in \{1, \ldots, m\}$ the following conditions hold:
$$k = \bullet \in E \implies q_k = b, \qquad \bullet = k \in E \implies b = q'_k, \qquad k = \bar{k} \in E \implies q_k = q'_{\bar{k}}.$$

▶ **Proposition 2.12.** *For every triple $((j,q), b, (j',q')) \in \bar{Q} \times \mathbb{A} \times \bar{Q}$, we have*

$$(j,q) \xrightarrow{b} (j',q') \text{ in } \bar{A} \qquad \text{iff} \qquad ((j,q), b, (j',q')) \text{ is consistent with some } (j, E, j') \in \mathsf{abs}(\delta).$$

▶ **Definition 2.13.** An *abstract run* in $\bar{A}$ is a sequence $(j_0, E_1, j_1, E_2, j_2, \ldots, E_n, j_n)$ such that $j_0 \in J_I$ and $j_{r-1} \xrightarrow{E_r} j_r$ for $r = 1, \ldots, n$. It is *accepting* if $j_n \in J_F$.

▶ **Notation 2.14.** Given an abstract run $(j_0, E_1, j_1, E_2, j_2, \dots, E_n, j_n)$, we inductively define the predicates $\mathsf{Eq}_k^{(i)}$ ($i \in \{1, \dots, n\}$, $k \in \{1, \dots, m\}$) on the set $\{1, \dots, n\}$:

1. if $\bullet = k$ in E_i then $\mathsf{Eq}_k^{(i)}(i)$;
2. if $r < n$ and $k = \bar{k}$ in E_{r+1} and $\mathsf{Eq}_k^{(i)}(r)$ then $\mathsf{Eq}_{\bar{k}}^{(i)}(r+1)$.

Informally, $\mathsf{Eq}_k^{(i)}(r)$ asserts that $1 \le i \le r \le n$ and that in every run in $\bar{A}$ of length r whose transitions are consistent with $E_1, \dots, E_r$, the i-th input letter equals the content of register k after r steps. The accepted language may be characterized using these predicates:

▶ **Proposition 2.15.** *The NOFRA $\bar{A}$ accepts the word $b_1 \cdots b_n \in \mathbb{A}^\star$ iff there exists an accepting abstract run of length n (with induced predicates $\mathsf{Eq}_k^{(i)}$) such that for $i, r \in \{1, \dots, n\}$,*

$$r < n \text{ and } k = \bullet \text{ in } E_{r+1} \text{ and } \mathsf{Eq}_k^{(i)}(r) \text{ for some } k \quad \Longrightarrow \quad b_i = b_{r+1}. \tag{2.1}$$

As a first application of this result, we identify an important difference between NOFA and NOFRA concerning the power of guessing data values during the computation:

▶ **Definition 2.16.** A NOFA/NOFRA is *non-guessing* if each initial state has empty support and for each transition $q \xrightarrow{a} q'$ one has $\mathsf{supp}\, q' \subseteq \mathsf{supp}\, q \cup \{a\}$.

The NOFA-recognizable language L_0 from the Introduction is not recognizable by any non-guessing NOFA [19, Ex. 1]. Note that L_0 is not positive; in fact, it is necessarily so, since for positive languages guessing does not add to the expressive power of automata:

▶ **Theorem 2.17.** *Every positive NOFA-recognizable language is accepted by some non-guessing NOFRA, hence by some non-guessing NOFA.*

To make a NOFRA non-guessing, one keeps track (via the state) of those registers containing data values forced by abstract transitions. The other registers then may be modified arbitrarily, which allows the elimination of guessing transitions.

3 Positive Register Automata

We now relate positive NOFA-recognizable languages to register automata, a.k.a. finite-memory automata, originally introduced by Kaminski and Francez [19]; we follow the equivalent presentation by Bojańczyk et al. [7]. A *register automaton* is a quintuple $A = (C, m, \delta, I, F)$ where C is a finite set of control states, $m \in \mathbb{N}$ is the number of registers (numbered from 1 to m), $I, F \subseteq C$ are sets of initial and final states, and $\delta \subseteq C \times \mathrm{Bool}(\Phi) \times C$ is the set of transitions. Here, $\mathrm{Bool}(\Phi)$ denotes the set of boolean formulas over the atoms $\Phi = (\{1, \dots, m\} \times \{\text{before}\} \cup \{\bullet\} \cup \{1, \dots, m\} \times \{\text{after}\})^2$. Elements of Φ are called *equations*; we write $x = y$ for $(x, y) \in \Phi$. Moreover, we denote $(c, \varphi, c') \in \delta$ by $c \xrightarrow{\varphi} c'$. A *configuration* of A is a pair (c, r) of a state $c \in C$ and a word $r \in (\mathbb{A} \cup \{\bot\})^m$ corresponding to a partial assignment of data values to the registers. The initial configurations are $(c, \bot^m)$ for $c \in I$. Given an input $a \in \mathbb{A}$ and configurations $(c, r), (c', r')$ we write $(c, r) \xrightarrow{a} (c', r')$ if this move is consistent with some transition $c \xrightarrow{\varphi} c'$, that is, the formula φ is true under the assignment making an atom $x = y \in \Phi$ true iff the corresponding data values are defined and equal. For instance, $(k, \text{before}) = \bullet$ is true iff $r_k \neq \bot$ and $r_k = a$, and $(k, \text{before}) = (\bar{k}, \text{after})$ is true iff $r_k, r'_{\bar{k}} \neq \bot$ and $r_k = r'_{\bar{k}}$. A word $w = a_1 \dots a_n \in \mathbb{A}^\star$ is *accepted* by A if it admits an accepting run, viz. a sequence of moves $(c_0, r_0) \xrightarrow{a_1} (c_1, r_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (c_n, r_n)$ where (c_0, r_0) is initial and $c_n \in F$. The *accepted language* $L(A) \subseteq \mathbb{A}^\star$ is the set of accepted words.

As shown by Bojańczyk et al. [7], register automata accept the same languages as NOFAs. To capture positive languages, we restrict to register automata with positive transitions:

▶ **Definition 3.1.** A register automaton is *positive* if for each transition $c \xrightarrow{\varphi} c'$ the formula φ is positive: $\varphi = \text{true}$ or φ uses the boolean operations $\vee$ and $\wedge$ only.

▶ **Theorem 3.2.** *A data language is positive and NOFA-recognizable iff it is accepted by some positive register automaton.*

Here, the approach is to regard a configuration of a positive register automaton as a state of a NOFRA. Conversely, an abstract transition $j \xrightarrow{E} j'$ of a NOFA can be transformed into a transition $j \xrightarrow{\varphi} j'$ of a register automaton for the conjunction φ of all equations in E, identifying $k = \bullet$, $\bullet = k$, $k = \bar{k}$ with $(k, \text{before}) = \bullet$, $\bullet = (k, \text{after})$, $(k, \text{before}) = (\bar{k}, \text{after})$. A tweak of the initial states accounts for the requirement that registers are initially empty.

▶ **Remark 3.3.** Just like register automata are equivalent to finite-memory automata, positive register automata correspond to a restricted version of finite-memory automata called *finite-state unification-based automata* (*FSUBA*) [20, 35]. The original definition of the latter involves a fixed initial register assignment, which enables acceptance of non-positive languages. However, FSUBA with empty initial registers are equivalent to positive register automata; see Appendix for details. This implies in particular that positive register automata admit a decidable inclusion problem, in contrast to the case of unrestricted register automata [28]. Indeed, FSUBA translate into a more general model called *RNNA* [32, Sec. 6], for which inclusion is decidable. Tal [35] has given a direct decidability proof for FSUBA.

4 Monadic Second-Order Logic with Positive Equality Tests

As motivated in the Introduction, positive data languages are considered as expressing properties of data words involving positive statements about equality of data values. In the following we make this idea precise. For this purpose, we employ monadic second-order logic with equality tests, abbreviated $\mathrm{MSO}^\sim$ [5, 11, 28]. Its formulae are given by the grammar

$$\varphi, \psi \quad := \quad x < y \mid x \sim y \mid X(x) \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \exists x.\, \varphi \mid \exists X.\, \varphi \mid \forall x.\, \varphi \mid \forall X.\, \varphi,$$

where x, y range over first-order variables and X over monadic second-order variables. A formula is interpreted over a fixed data word $w = a_1 \ldots a_n \in \mathbb{A}^\star$. First-order variables represent positions, i.e. elements of the set $\{1, \ldots, n\}$, and second-order variables represent subsets of $\{1, \ldots, n\}$. The atomic formula $x < y$ means "position x comes before position y", and $x \sim y$ means "the same data value occurs at positions x and y". The interpretation of the remaining constructs is standard. A *sentence* is a formula without free variables. We write $L(\varphi) \subseteq \mathbb{A}^\star$ for the set of data words satisfying the sentence φ. For example, the languages L_0 and L_1 from the Introduction are defined by $\varphi_0 = \forall y.\, \text{last}(y) \Rightarrow (\forall x.\, x < y \Rightarrow \neg(x \sim y))$, where $\text{last}(y) = \neg\exists z.\, y < z$ and $\psi \Rightarrow \xi = \neg\psi \vee \xi$, and by $\varphi_1 = \exists x.\, \exists y.\, x < y \wedge x \sim y$.

Recall that by standard rules of negation, every formula is equivalent to one in *negation normal form* (*NNF*), where for each subformula $\neg\varphi$ the formula φ is atomic.

▶ **Definition 4.1.** An $\mathrm{MSO}^\sim$ formula lies in $\mathrm{MSO}^{\sim,+}$ (*monadic second-order logic with positive equality tests*) if it admits an NNF containing no subformula of the form $\neg(x \sim y)$. A data language is *$\mathrm{MSO}^{\sim,+}$-definable* if it is of the form $L(\varphi)$ for an $\mathrm{MSO}^{\sim,+}$ sentence φ.

The above sentence φ_1 lies in $\mathrm{MSO}^{\sim,+}$ but φ_0 does not. The following is immediate:

▶ **Proposition 4.2.** *Every $\mathrm{MSO}^{\sim,+}$-definable language is positive.*

▶ **Remark 4.3.** The logic MSO$^{\sim}$ is more expressive than NOFAs [28], and the same holds for MSO$^{\sim,+}$: the language defined by the MSO$^{\sim,+}$ sentence $\varphi = \forall x.\, \exists y.\, (x < y \vee y < x) \wedge x \sim y$ ("no data value occurs only once") is not NOFA-recognizable. However, within the class of NOFA-recognizable languages, positive and MSO$^{\sim,+}$-definable languages coincide:

▶ **Theorem 4.4.** *A NOFA-recognizable language is positive iff it is MSO$^{\sim,+}$-definable.*

Indeed, one can express the abstract acceptance condition of Proposition 2.15 in MSO$^{\sim,+}$.

5 Toposes for Names

In the remainder, we investigate positive data languages and their automata from a more conceptual perspective. Some familiarity with basic category theory (functors, natural transformations, (co-)limits, adjunctions) is required; see Mac Lane [24] for a gentle introduction.

Nominal sets and nominal renamings sets (Section 2.1) were initially introduced as a convenient abstract framework for reasoning about names, and related issues such as freshness, binding, and substitution. An alternative, and more general, approach uses the presheaf categories $\mathbf{Set}^{\mathbb{I}}$ [34] and $\mathbf{Set}^{\mathbb{F}}$ [12]. The intuition behind each of these categories $\mathscr{C}$ is very similar: one thinks of $X \in \mathscr{C}$ as a collection of finitely supported objects, equipped with a renaming operation that extends renamings $\rho\colon \mathbb{A} \to \mathbb{A}$ to the level of elements of X. The difference between the four categories lies in whether elements admit a *least* support, or just some finite support, and in whether renamings ρ are injective or arbitrary maps; see Figure 2. The last column classifies the respective finitely presentable objects, which underly automata. We now recall the latter concept and describe the categories in more detail.

Finitely presentable objects. A diagram $D\colon I \to \mathscr{C}$ in a category $\mathscr{C}$ is *directed* if its scheme I is a directed poset: every finite subset of I has an upper bound. A *directed colimit* is a colimit of a directed diagram. An object X of $\mathscr{C}$ is *finitely presentable* if its hom-functor $\mathscr{C}(X, -)\colon \mathscr{C} \to \mathbf{Set}$ to the category of sets and functions preserves directed colimits. In many categories, finitely presentable objects correspond to the objects with a finite description. For example, the finitely presentable objects of $\mathbf{Set}$ are precisely finite sets, and if $\mathscr{C}$ is a variety of algebras (e.g. monoids, groups, rings), an algebra is a finitely presentable object of $\mathscr{C}$ iff it is presentable by finitely many generators and relations [3, Thm. 3.12].

Nominal (renaming) sets. We let $\mathbf{Nom}$ denote the category of nominal sets and $\mathsf{Perm}(\mathbb{A})$-equivariant maps, and $\mathbf{RnNom}$ the category of nominal renaming sets and $\mathsf{Fin}(\mathbb{A})$-equivariant maps. Both categories are toposes, that is, they are finitely complete (with limits formed as in $\mathbf{Set}$), cartesian closed, and admit a subobject classifier. Note that $\mathbf{Nom}$ is a boolean topos (its subobject classifier is $2 = \{0, 1\}$ with the trivial group action), which is not true for $\mathbf{RnNom}$ [14, Sec. 5]. The next proposition provides a categorical description of orbit-finite nominal (renaming) sets; for nominal sets this result is well-known, see [29, Prop. 2.3.7] or [30, Thm. 5.16], and the statement for nominal renaming sets may be deduced from it.

▶ **Proposition 5.1.** *A nominal (renaming) set is orbit-finite iff it is a finitely presentable object of* $\mathbf{Nom}$ *or* $\mathbf{RnNom}$*, respectively.*

The forgetful functor $U\colon \mathbf{RnNom} \to \mathbf{Nom}$ given by restricting the $\mathsf{Fin}(\mathbb{A})$- to a $\mathsf{Perm}(\mathbb{A})$-action has a left adjoint $F\colon \mathbf{Nom} \to \mathbf{RnNom}$ [27, Thm. 2.6]. We refer to *op. cit.* for its explicit description, but remark that $F(\mathbb{A}^{\# n}) = \mathbb{A}^n$ for every $n \in \mathbb{N}$ [27, Thm. 3.7].

Category	Objects	Least supp.	Renamings	Finitely pres. objects
Nom	nominal sets	yes	injective	orbit-finite sets
RnNom	nominal renaming sets	yes	arbitrary	orbit-finite sets
$\mathbf{Set}^{\mathbb{I}}$	presheaves over $\mathbb{I}$	no	injective	super-finitary presheaves
$\mathbf{Set}^{\mathbb{F}}$	presheaves over $\mathbb{F}$	no	arbitrary	super-finitary presheaves

Figure 2 Toposes that model sets of finitely supported objects

Presheaves. A *(covariant) presheaf* over a small category $\mathscr{C}$ is a functor $P\colon \mathscr{C} \to \mathbf{Set}$. We write $\mathbf{Set}^{\mathscr{C}}$ for the category of presheaves and natural transformations. We specifically consider presheaves over $\mathbb{F}$ and $\mathbb{I}$, the categories whose objects are finite subsets $S \subseteq_{\mathsf{f}} \mathbb{A}$ and whose morphisms $\rho\colon S \to T$ are functions or injective functions, respectively. The categories **Nom** and **RnNom** form full reflective subcategories of $\mathbf{Set}^{\mathbb{I}}$ and $\mathbf{Set}^{\mathbb{F}}$ via embeddings

$$I_\star\colon \mathbf{Nom} \rightarrowtail \mathbf{Set}^{\mathbb{I}} \qquad \text{and} \qquad J_\star\colon \mathbf{RnNom} \rightarrowtail \mathbf{Set}^{\mathbb{F}}.$$

Here, $I_\star$ is given for $X \in \mathbf{Nom}$, $S \subseteq_{\mathsf{f}} \mathbb{A}$, $\rho\colon S \to T$ in $\mathbb{I}$ and $f\colon X \to Y$ in **Nom** by

$$(I_\star X)S = \{\, x \in X : \mathsf{supp}\, x \subseteq S \,\}, \qquad (I_\star X)\rho(x) = \overline{\rho} \cdot x, \qquad (I_\star f)_S(x) = f(x),$$

where $\overline{\rho} \in \mathsf{Perm}(\mathbb{A})$ is any permutation extending the injective map ρ. The embedding $J_\star$ is defined analogously. In both cases, the essential image of the embedding consists precisely of the presheaves preserving pullbacks of injective maps, see [30, Thm. 6.8] and [14, Thm. 38]. Informally, a presheaf $P \in \mathbf{Set}^{\mathscr{C}}$, where $\mathscr{C} \in \{\mathbb{I}, \mathbb{F}\}$, specifies a set PS of S-supported objects for every $S \subseteq_{\mathsf{f}} \mathbb{A}$, and the pullback preservation property asserts precisely that every object admits a least support. A presheaf $P \in \mathbf{Set}^{\mathscr{C}}$ is *super-finitary* if there exists $S \subseteq_{\mathsf{f}} \mathbb{A}$ such that (i) PS' is a finite set for all $S' \subseteq S$, and (ii) for every $T \subseteq_{\mathsf{f}} \mathbb{A}$ and $x \in PT$, there exists $S' \subseteq S$ and $\rho \in \mathscr{C}(S', T)$ such that $x \in P\rho[PS']$. (This implies that PT is finite.) Such an S is called a *generating set* for P. The next proposition shows that super-finitary presheaves are the analogue of orbit-finite sets; see [2, Cor. 3.34] for the case $\mathscr{C} = \mathbb{F}$:

▶ **Proposition 5.2.** *For $\mathscr{C} \in \{\mathbb{I}, \mathbb{F}\}$ and $P \in \mathbf{Set}^{\mathscr{C}}$, the following are equivalent: (i) P is super-finitary; (ii) P is finitely presentable; (iii) there exists an epimorphism (a componentwise surjective natural transformation) $\coprod_{i \in I} \mathscr{C}(S_i, -) \twoheadrightarrow P$ with I finite and $S_i \subseteq_{\mathsf{f}} \mathbb{A}$. Moreover, super-finitary presheaves are closed under sub-presheaves and finite products.*

To relate the two presheaf categories $\mathbf{Set}^{\mathbb{I}}$ and $\mathbf{Set}^{\mathbb{F}}$, recall that every functor $E\colon \mathscr{C} \to \mathscr{D}$ between small categories induces an adjunction (5.1), where the right adjoint $E^\star$ is given by $E^\star(P) = P \circ E$, and the left adjoint sends a presheaf $P \in \mathbf{Set}^{\mathscr{C}}$ to its *left Kan extension* $\mathsf{Lan}_E P$. For the inclusion functor $E\colon \mathbb{I} \hookrightarrow \mathbb{F}$, we obtain the commutative diagram (5.2) of adjunctions. Here, $I^\star$ and $J^\star$ are the reflectors, i.e. the left adjoints of $I_\star$ and $J_\star$.

$$\begin{tikzcd} \mathbf{Set}^{\mathscr{C}} \arrow[r, shift right, "\mathsf{Lan}_E"'] \arrow[r, phantom, "\top"] & \mathbf{Set}^{\mathscr{D}} \arrow[l, shift right, "E^\star"'] \end{tikzcd} \qquad (5.1)$$

$$\begin{tikzcd} \mathbf{Set}^{\mathbb{I}} \arrow[r, shift right, "\mathsf{Lan}_E"'] \arrow[r, phantom, "\top"] \arrow[d, shift left, "I^\star"] \arrow[d, phantom, "\vdash"] & \mathbf{Set}^{\mathbb{F}} \arrow[l, shift right, "E^\star"'] \arrow[d, shift right, "J^\star"'] \arrow[d, phantom, "\dashv"] \\ \mathbf{Nom} \arrow[u, shift left, "I_\star"] \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{RnNom} \arrow[u, shift right, "J_\star"'] \arrow[l, shift left, "U"] \end{tikzcd} \qquad (5.2)$$

▶ **Proposition 5.3.** *All functors in* (5.2) *preserve finitely presentable objects.*

Hence, the adjunctions (5.2) restrict to the full subcategories of finitely presentable objects.

6 Nondeterministic Automata in a Category

Our aim is to investigate nondeterministic automata and their languages in the toposes of Figure 2, and to compare their expressive power. To this end, we first introduce the required automata-theoretic concepts uniformly at the level of abstract categories.

▶ **Assumptions 6.1.** Fix a category $\mathscr{C}$ with finite limits and (strong epi, mono)-factorizations. We assume that strong epimorphisms are stable under finite products (that is, $e \times e'$ is a strong epimorphism whenever e and e' are) and pullbacks (that is, in every pullback square $e \circ \overline{f} = f \circ \overline{e}$, the morphism $\overline{e}$ is a strong epimorphism whenever e is).

The (strong epi, mono)-factorization $f = (A \xrightarrow{e} I \xrightarrow{m} B)$ of a morphism $f \colon A \to B$ in $\mathscr{C}$ is its *image factorization*, and the subobject represented by m is the *image* of f.

▶ **Example 6.2.** Every topos satisfies Assumptions 6.1, including $\mathbf{Set}$, $\mathbf{Nom}$, $\mathbf{RnNom}$, $\mathbf{Set}^{\mathbb{I}}$ and $\mathbf{Set}^{\mathbb{F}}$. Note that in a topos all epimorphisms are strong. In the five categories above, epi- and monomorphisms are the (componentwise) surjective and injective morphisms, resp. Pullbacks and finite products are formed (componentwise) at the level of underlying sets.

▶ **Definition 6.3.** A *language* over $\Sigma \in \mathscr{C}$ is a family of subobjects of Σ^n for each $n \in \mathbb{N}$:

$$L \;=\; (\, m_n^{(L)} \colon L^{(n)} \rightarrowtail \Sigma^n \,)_{n \in \mathbb{N}}.$$

We write $L \leq L'$ iff $L^{(n)} \leq L'^{(n)}$ for all n, using the partial order $\leq$ on subobjects of Σ^n.

▶ **Remark 6.4.** If $\mathscr{C}$ is countably extensive (e.g. a topos with countable coproducts), languages correspond bijectively to subobjects of $\Sigma^\star = \coprod_{n\in\mathbb{N}} \Sigma^n$. Indeed, every language L yields the subobject $\coprod_n m_n^{(L)} \colon \coprod_n L^{(n)} \rightarrowtail \Sigma^\star$, and conversely every subobject of $\Sigma^\star$ is of this form. In particular, this holds in the categories of Example 6.2.

▶ **Definition 6.5.** A *nondeterministic $\mathscr{C}$-automaton* is a quintuple $A = (Q, \Sigma, \delta, I, F)$ consisting of an object $Q \in \mathscr{C}$ of *states*, an *input alphabet* $\Sigma \in \mathscr{C}$, and subobjects

$$m_\delta \colon \delta \rightarrowtail Q \times \Sigma \times Q, \qquad m_I \colon I \rightarrowtail Q, \qquad m_F \colon F \rightarrowtail Q,$$

representing *transitions*, *initial states*, and *final states*, respectively. A *morphism* $h \colon A' \to A$ of nondeterministic $\mathscr{C}$-automata is given by a pair of morphisms $h_{\mathsf{s}} \colon Q' \to Q$ and $h_{\mathsf{a}} \colon \Sigma' \to \Sigma$ of $\mathscr{C}$ that restrict as shown below (note that h_{t}, h_{i} and h_{f} are uniquely determined):

$$\begin{tikzcd} \delta' \arrow[r, dashed, "h_{\mathsf{t}}"] \arrow[d, tail, "m_{\delta'}"'] & \delta \arrow[d, tail, "m_\delta"] & I' \arrow[r, dashed, "h_{\mathsf{i}}"] \arrow[d, tail, "m_{I'}"'] & I \arrow[d, tail, "m_I"] & F' \arrow[r, dashed, "h_{\mathsf{f}}"] \arrow[d, tail, "m_{F'}"'] & F \arrow[d, tail, "m_F"] \\ Q' \times \Sigma' \times Q' \arrow[r, "h_{\mathsf{s}} \times h_{\mathsf{a}} \times h_{\mathsf{s}}"] & Q \times \Sigma \times Q & Q' \arrow[r, "h_{\mathsf{s}}"] & Q & Q' \arrow[r, "h_{\mathsf{s}}"] & Q \end{tikzcd} \tag{6.1}$$

We write $\mathbf{NAut}(\mathscr{C})$ for the category of nondeterministic automata in $\mathscr{C}$ and their morphisms, and $\mathbf{NAut}_{\mathsf{fp}}(\mathscr{C})$ for its full subcategory given by *nondeterministic fp-automata*, viz. automata where Q, Σ, δ, I, F are finitely presentable objects of $\mathscr{C}$.

▶ **Definition 6.6.** For every nondeterministic $\mathscr{C}$-automaton $A = (Q, \Sigma, \delta, I, F)$, its *accepted language* is the language $L(A)$ over Σ given as follows:

1. $m_{L(A)}^{(0)} \colon L^{(0)}(A) \rightarrowtail 1 = \Sigma^0$ is the image of the unique morphism $I \cap F \xrightarrow{!} 1$, where 1 is the terminal object of $\mathscr{C}$ and $I \cap F$ is the intersection (pullback) of m_I and m_F.

2. For $n > 0$, the subobject $m^{(n)}_{L(A)} \colon L^{(n)}(A) \rightarrowtail \Sigma^n$ is defined via the commutative diagram

$$\begin{tikzcd}
L^{(n)}(A) \arrow[d, tail, "m^{(n)}_{L(A)}"'] & \mathsf{AccRun}^{(n)}_A \arrow[l, two heads, "e_{n,A}"'] \arrow[r, tail, "\overline{d}_{n,A}"] \arrow[d, tail, "\overline{m}^{(n)}_\delta"] \arrow[dr, phantom, "\lrcorner", very near start] & \delta^n \arrow[d, tail, "m^n_\delta"] \\
\Sigma^n & I \times (\Sigma \times Q)^{n-1} \times \Sigma \times F \arrow[l, "p_{n,A}"'] \arrow[r, tail, "d_{n,A}"] & (Q \times \Sigma \times Q)^n
\end{tikzcd}$$

Here, letting $\Delta \colon Q \rightarrowtail Q \times Q$ denote the diagonal, $d_{n,A}$ is the monomorphism

$$I \times (\Sigma \times Q)^{n-1} \times \Sigma \times F \xrightarrow{m_I \times (\mathsf{id} \times \Delta)^{n-1} \times \mathsf{id} \times m_F} Q \times (\Sigma \times Q \times Q)^{n-1} \times \Sigma \times Q \cong (Q \times \Sigma \times Q)^n,$$

the morphisms $\overline{d}_{n,A}$ and $\overline{m}^{(n)}_\delta$ form the pullback of $d_{n,A}$ and m^n_δ, the morphism $p_{n,A}$ is the projection, and $e_{n,A}$ and $m^{(n)}_{L(A)}$ form the image factorization of $p_{n,A} \circ \overline{m}^{(n)}_\delta$.

▶ **Example 6.7.** **1.** A nondeterministic fp-automaton in **Set** is a classical nondeterministic finite automaton. The pullback $\mathsf{AccRun}^{(n)}_A$ is the set of accepting runs of length n, hence $L(A)$ is the usual accepted language: the set of words with an accepting run.

2. A nondeterministic fp-automaton in **Nom** or **RnNom** with alphabet $\Sigma = \mathbb{A}$ is a NOFA or NOFRA, respectively. The two notions of accepted language in Definition 2.3 and Definition 6.6 match, that is, $L(A)$ is the set of words with an accepting run.

3. In the next section, we will also look into nondeterministic $\mathbf{Set}^{\mathbb{I}}$- and $\mathbf{Set}^{\mathbb{F}}$-automata.

▶ **Remark 6.8.** Readers familiar with coalgebras [31] may note that if $\mathscr{C}$ is a topos, the final states and transitions of a nondeterministic $\mathscr{C}$-automaton correspond to a coalgebra $\gamma \colon Q \to \Omega \times (\mathcal{P}Q)^\Sigma$ where Ω is the subobject classifier and $\mathcal{P} \colon \mathscr{C} \to \mathscr{C}$ is the covariant power object functor [18, Sec. A.2.3]. We expect our above definition of accepted language to match the one given by coalgebraic trace semantics [17, 33], with the required arguments relying on the internal logic of the topos $\mathscr{C}$. Details are left for future work; we have found that the present relational approach to automata leads to shorter and more direct proofs.

▶ **Proposition 6.9.** *Let $h \colon A' \to A$ be an $\mathbf{NAut}(\mathscr{C})$-morphism where $\Sigma' = \Sigma$ and $h_{\mathsf{a}} = \mathsf{id}_\Sigma$.*

1. *The accepted language of A' is contained in that of A, that is, $L(A') \leq L(A)$.*

2. *If h_{s} is strongly epic in $\mathscr{C}$ and the squares (6.1) are pullbacks, then $L(A') = L(A)$.*

Hence, the construction $A \mapsto A'$ of Remark 2.6 indeed yields an equivalent NOFA.

▶ **Proposition 6.10.** *Let $\mathscr{C}$ and $\mathscr{D}$ be categories satisfying the Assumptions 6.1.*

1. *Every functor $G \colon \mathscr{C} \to \mathscr{D}$ lifts to a functor $\bar{G} \colon \mathbf{NAut}(\mathscr{C}) \to \mathbf{NAut}(\mathscr{D})$ defined by*

$$\bar{G}(Q, \Sigma, \delta, I, F) = (GQ, G\Sigma, \overline{G\delta}, \overline{GI}, \overline{GF}) \qquad \text{and} \qquad \bar{G}f = Gf.$$

Here, $\overline{G\delta}$, $\overline{GI}$, $\overline{GF}$ are the images of the morphisms shown below, with can *denoting the canonical morphism induced by the product projections:*

$$G\delta \xrightarrow{Gm_\delta} G(Q \times \Sigma \times Q) \xrightarrow{\mathsf{can}} GQ \times G\Sigma \times GQ, \qquad GI \xrightarrow{Gm_I} GQ, \qquad GF \xrightarrow{Gm_F} GQ.$$

2. *Every adjunction $L \dashv R \colon \mathscr{C} \to \mathscr{D}$ lifts to an adjunction $\bar{L} \dashv \bar{R} \colon \mathbf{NAut}(\mathscr{C}) \to \mathbf{NAut}(\mathscr{D})$.*

In particular, the adjunctions (5.2) lift to adjunctions between the respective categories of nondeterministic automata, which in turn restrict to fp-automata by Proposition 5.3:

$$\begin{tikzcd}
\mathbf{NAut}_{\mathsf{fp}}(\mathbf{Set}^{\mathbb{I}}) \arrow[r, shift right=1.5ex, "\overline{\mathsf{Lan}_E}"'] \arrow[r, phantom, "\top"] \arrow[d, shift left=1.5ex, "\bar{I}^\star"] \arrow[d, phantom, "\vdash"] & \mathbf{NAut}_{\mathsf{fp}}(\mathbf{Set}^{\mathbb{F}}) \arrow[l, shift right=1.5ex, "\bar{E}^\star"'] \arrow[d, shift right=1.5ex, "\bar{J}^\star"'] \arrow[d, phantom, "\dashv"] \\
\mathbf{NAut}_{\mathsf{fp}}(\mathbf{Nom}) \arrow[u, shift left=1.5ex, "\bar{I}_\star"] \arrow[r, shift left=1.5ex, "\bar{F}"] \arrow[r, phantom, "\perp"] & \mathbf{NAut}_{\mathsf{fp}}(\mathbf{RnNom}) \arrow[l, shift left=1.5ex, "\bar{U}"] \arrow[u, shift right=1.5ex, "\bar{J}_\star"']
\end{tikzcd} \tag{6.2}$$

The positive closure $A \mapsto \bar{A}$ of Construction 2.7, which is key to our results in Sections 2 through 4, is an instance of the proposition since $\bar{A} = \bar{F}A$ for the left adjoint $F\colon \mathbf{Nom} \to \mathbf{RnNom}$.

7 Nondeterministic Presheaf Automata

We proceed to relate the expressive power of the four automata models in (6.2). Specifically, for $\mathscr{C} \in \{\mathbb{I}, \mathbb{F}\}$ we consider nondeterministic $\mathbf{Set}^{\mathscr{C}}$-automata $A = (Q, \Sigma, \delta, I, F)$ with a super-finitary (= finitely presentable) presheaf Q of states and input alphabet $\Sigma = V_{\mathscr{C}} \in \mathbf{Set}^{\mathscr{C}}$, for the inclusion functor $V_{\mathscr{C}}(S) = S$. (This implies that δ, I and F are super-finitary by Proposition 5.2.) Note that $V_{\mathscr{C}}$ corresponds to the input alphabet $\mathbb{A}$ used for NOF(R)As:

$$V_{\mathbb{I}} = I_\star(\mathbb{A}) \qquad \text{and} \qquad V_{\mathbb{F}} = J_\star(\mathbb{A}) = \mathsf{Lan}_E(V_{\mathbb{I}}).$$

A *language* in $\mathbf{Set}^{\mathscr{C}}$ is a sub-presheaf $L \subseteq V_{\mathscr{C}}^\star$, or equivalently a family of sub-presheaves $L^{(n)} \subseteq V_{\mathscr{C}}^n$ for $n \in \mathbb{N}$ (Definition 6.3 and Remark 6.4). Here, $V_{\mathscr{C}}^\star(S) = S^\star$, the set of words over the finite alphabet $S \subseteq_{\mathsf{f}} \mathbb{A}$, and $V_{\mathscr{C}}^n(S) = S^n$, the subset of words of length n.

▶ **Remark 7.1.** For the sake of distinction, we refer to languages in $\mathbf{Set}^{\mathscr{C}}$ as *presheaf languages*, and to subsets of $\mathbb{A}^\star$ as *word languages*. Both concepts are closely related: Every presheaf language $L \subseteq V_{\mathbb{I}}^\star$ in $\mathbf{Set}^{\mathbb{I}}$ induces a $\mathsf{Perm}(\mathbb{A})$-equivariant word language $\mathsf{W}(L) \subseteq \mathbb{A}^\star$ given by $\mathsf{W}(L) = \bigcup_{S \subseteq_{\mathsf{f}} \mathbb{A}} L(S)$, and, conversely, every $\mathsf{Perm}(\mathbb{A})$-equivariant word language $K \subseteq \mathbb{A}^\star$ induces a presheaf language $\mathsf{P}(K) \subseteq V_{\mathbb{I}}^\star$ given by $[\mathsf{P}(K)]S = K \cap S^\star$ for $S \subseteq_{\mathsf{f}} \mathbb{A}$. Analogously for presheaf languages in $\mathbf{Set}^{\mathbb{F}}$ and $\mathsf{Fin}(\mathbb{A})$-equivariant word languages. In both cases, these translations almost yield a bijective correspondence: one has $K = \mathsf{W}(\mathsf{P}(K))$, but generally only $L \subseteq \mathsf{P}(\mathsf{W}(L))$. For instance, for $L \subseteq V_{\mathbb{F}}^\star$ given by $L(\emptyset) = \emptyset$ and $L(S) = \{\varepsilon\}$ for $S \neq \emptyset$ one has $[\mathsf{P}(\mathsf{W}(L))]\emptyset = \{\varepsilon\}$, so $L \subsetneq \mathsf{P}(\mathsf{W}(L))$. The equality $L = \mathsf{P}(\mathsf{W}(L))$ holds iff L is *downwards closed*, that is, $L(S') = L(S) \cap (S')^\star$ for all $S' \subseteq S \subseteq_{\mathsf{f}} \mathbb{A}$.

The presheaf version of positive word languages and positive closures is as follows:

▶ **Definition 7.2.** Let $L \subseteq V_{\mathbb{I}}^\star$ be a presheaf language in $\mathbf{Set}^{\mathbb{I}}$.
1. The language L is *positive* if $L = KE$ for some (unique) language $K \subseteq V_{\mathbb{F}}^\star$ in $\mathbf{Set}^{\mathbb{F}}$.
2. A *positive closure* of L is a language $\bar{L}$ in $\mathbf{Set}^{\mathbb{F}}$ such that $L \subseteq \bar{L}E$ and $\bar{L}$ is minimal with that property, that is, $\bar{L} \subseteq K$ for every language $K \subseteq V_{\mathbb{F}}^\star$ in $\mathbf{Set}^{\mathbb{F}}$ such that $L \subseteq KE$.

A positive closure is clearly unique; its existence is ensured by the next proposition, which is proved using the universal property of left Kan extensions.

▶ **Proposition 7.3.** *The positive closure of $L \subseteq V_{\mathbb{I}}^*$ is given by the image of the morphism*

$$\varphi\colon \mathsf{Lan}_E(L) \xrightarrow{\mathsf{Lan}_E(\iota)} \mathsf{Lan}_E(V_{\mathbb{I}}^*) \cong \coprod_k \mathsf{Lan}_E(V_{\mathbb{I}}^k) \xrightarrow{\coprod_k \mathsf{can}_k} \coprod_k \mathsf{Lan}_E(V_{\mathbb{I}})^k = \coprod_k V_{\mathbb{F}}^k = V_{\mathbb{F}}^*$$

where $\iota\colon L \hookrightarrow V_{\mathbb{I}}^*$ is the inclusion, the isomorphism witnesses preservation of coproducts by the left adjoint Lan_E, and can_k is the canonical map induced by the product projections.

▶ **Remark 7.4.** A presheaf $P \in \mathbf{Set}^{\mathbb{I}}$ is *strong* if $P = I_\star(X)$ for a strong nominal set X. Since $I_\star$ preserves coproducts, (super-finitary) strong presheaves are exactly (finite) coproducts $\coprod_{j \in J} \mathbb{I}(S_j, -)$ of representable presheaves. By Proposition 5.2 and Proposition 6.9, every super-finitary $\mathbf{Set}^{\mathbb{I}}$-automaton is equivalent to one whose presheaf of states is strong. Given such an automaton A with states $Q = \coprod_{j \in J} \mathbb{I}(S_j, -)$, applying the lifted left adjoint

$\overline{\mathsf{Lan}}_E$ yields a super-finitary $\mathbf{Set}^{\mathbb{F}}$-automaton $\bar{A}$ with states $\mathsf{Lan}_E(Q) = \coprod_{j \in J} \mathbb{F}(S_j, -)$, using that Lan_E preserves coproducts and representables (see e.g. [24, Ex. X.3.2]). This is the analogue of Construction 2.7 for presheaf automata. Similar to Proposition 2.8, we have

▶ **Proposition 7.5.** *For every super-finitary nondeterministic* $\mathbf{Set}^{\mathbb{I}}$*-automaton* A *with a strong presheaf of states, the* $\mathbf{Set}^{\mathbb{F}}$*-automaton* $\bar{A} = \overline{\mathsf{Lan}}_E(A)$ *accepts the language* $\overline{L(A)}$*.*

While by definition nondeterministic presheaf automata accept presheaf languages, using Remark 7.1 we can also naturally associate a word language semantics to them:

▶ **Definition 7.6.** **1.** The word language *accepted* by a nondeterministic $\mathbf{Set}^{\mathscr{C}}$-automaton A is $\mathsf{W}(L(A)) \subseteq \mathbb{A}^\star$, the word language induced by the presheaf language of A.
2. A word language $L \subseteq \mathbb{A}^\star$ is $\mathbf{Set}^{\mathscr{C}}$*-recognizable* if there exists a super-finitary nondeterministic $\mathbf{Set}^{\mathscr{C}}$-automaton accepting it.

This enables a classification of the expressive power of nondeterministic $\mathbf{Set}^{\mathscr{C}}$-automata:

▶ **Theorem 7.7.** **1.** *A word language is NOFA-recognizable iff it is* $\mathbf{Set}^{\mathbb{I}}$*-recognizable.*
2. *A word language is positive and NOFA-recognizable iff it is* $\mathbf{Set}^{\mathbb{F}}$*-recognizable.*

For item 1 one shows that the functors $\bar{I}_\star$ and $\bar{I}^\star$ of (6.2) preserve the accepted word languages of automata. For item 2 one uses Proposition 7.5 and the observation that every nondeterministic $\mathbf{Set}^{\mathbb{F}}$-automaton accepts a positive word language.

This shows that the theory of data languages can be based on presheaves rather than nominal sets [7]. In particular, the conceptual difference between the two approaches (viz. existence of least supports) is largely inessential from the perspective of automata theory.

8 Conclusions and Future Work

We have characterized positive data languages recognizable by NOFAs in terms of register automata, logic, and category theory; see Figure 1 for a summary of our contributions. Our results underline the phenomenon that weak classes of data languages tend to have a rich theory and admit many equivalent perspectives, paralleling classical regular languages over finite alphabets. For example, a similar observation has been made for data languages recognizable by orbit-finite nominal monoids [5,9,11].

The logic $\mathrm{MSO}^{\sim,+}$ defines positive data languages, but is more expressive than NOFAs. Identifying a suitable syntactic fragment of $\mathrm{MSO}^{\sim,+}$ that captures precisely the positive NOFA-recognizable languages remains an open problem. The same holds for the decidability of the satisfiability problem for $\mathrm{MSO}^{\sim,+}$, which is known to be undecidable for $\mathrm{MSO}^{\sim}$ [22]. On a related note, it might be interesting to characterize the expressive power of full $\mathrm{MSO}^{\sim,+}$. Specifically, does it capture precisely the $\mathrm{MSO}^{\sim}$-definable positive languages?

Finally, besides register automata, a number of further automata models for data languages have been proposed, most notably pebble automata [28] and data automata [6,8]. In general, these models differ in their expressive power. However, it is conceivable that some or all of them may become equivalent when restricted to positive data languages.

References

1 Jiří Adámek, Stefan Milius, Lurdes Sousa, and Thorsten Wißmann. Finitely presentable algebras for finitary monads. *Theory Appl. Categ.*, 34(37):1179–1195, 2019.
2 Jiří Adámek, Stefan Milius, Lurdes Sousa, and Thorsten Wißmann. On finitary functors. *Theory Appl. Categ.*, 34(37):1134–1164, 2019.

3 Jiří Adámek and Jiří Rosický. *Locally Presentable and Accessible Categories*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1994.

4 Michał Bielecki, Jan Hidders, Jan Paredaens, Jerzy Tyszkiewicz, and Jan Van den Bussche. Navigating with a browser. In *Proc. 29th International Colloquium on Automata, Languages and Programming (ICALP 2002)*, volume 2380 of *Lect. Notes Comput. Sci.*, pages 764–775. Springer, 2002.

5 Mikołaj Bojańczyk. Nominal monoids. *Theory Comput. Syst.*, 53(2):194–222, 2013.

6 Mikołaj Bojańczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data trees and XML reasoning. In *Proc. 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2006)*, pages 10–19. ACM, 2006.

7 Mikołaj Bojańczyk, Bartek Klin, and Sławomir Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10(3), 2014.

8 Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, and Claire David. Two-variable logic on words with data. In *Proc. 21th IEEE Symposium on Logic in Computer Science (LICS 2006)*, pages 7–16. IEEE Computer Society, 2006.

9 Mikołaj Bojańczyk and Rafał Stefański. Single-use automata and transducers for infinite alphabets. In *Proc. 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *LIPIcs*, pages 113:1–113:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

10 Francis Borceux. *Handbook of Categorical Algebra 1 – Basic Category Theory*. Cambridge University Press, 1994.

11 Thomas Colcombet, Clemens Ley, and Gabriele Puppis. Logics with rigidly guarded data tests. *Log. Methods Comput. Sci.*, 11(3), 2015.

12 Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *Proc. 14th Annual IEEE Symposium on Logic in Computer Science (LICS 1999)*, pages 193–202. IEEE Computer Society, 1999.

13 Murdoch J. Gabbay. Nominal renaming sets (technical report). `http://gabbay.org.uk/papers/nomrs-tr.pdf`, 2007.

14 Murdoch J. Gabbay and Martin Hofmann. Nominal renaming sets. In *Proc. 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2008)*, page 158–173. Springer, 2008.

15 Murdoch J. Gabbay and Andrew M. Pitts. A new approach to abstract syntax involving binders. In *Proc. 14th Annual IEEE Symposium on Logic in Computer Science (LICS 1999)*, pages 214–224. IEEE Computer Society, 1999.

16 Fabio Gadducci, Marino Miculan, and Ugo Montanari. About permutation algebras, (pre)sheaves and named sets. *High. Order Symb. Comput.*, 19(2-3):283–304, 2006.

17 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Log. Methods Comput. Sci.*, 3(4:11):1–36, 2007.

18 Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*. Oxford Logic Guides. Oxford Univ. Press, 2002.

19 Michael Kaminski and Nissim Francez. Finite-memory automata. *Theor. Comput. Sci.*, 134(2):329–363, 1994.

20 Michael Kaminski and Tony Tan. Regular expressions for languages over infinite alphabets. *Fundam. Informaticae*, 69(3):301–318, 2006.

21 Michael Kaminski and Daniel Zeitlin. Finite-memory automata with non-deterministic reassignment. *Int. J. Found. Comput. Sci.*, 21(5):741–760, 2010.

22 Bartek Klin, Sławomir Lasota, and Szymon Torunczyk. Nondeterministic and co-nondeterministic implies deterministic, for data languages. In *Proc. 24th International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2021)*, volume 12650 of *Lect. Notes Comput. Sci.*, pages 365–384. Springer, 2021.

23 Klaas Kürtz, Ralf Küsters, and Thomas Wilke. Selecting theories and nonce generation for recursive protocols. In *Proc. 2007 ACM Workshop on Formal Methods in Security Engineering (FMSE 2007)*, pages 61–70. ACM, 2007.

24 Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.

25 Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer, 1992.

26 Stefan Milius and Henning Urbat. Equational axiomatization of algebras with structure. In *Proc. 22nd International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2019)*, volume 11425 of *Lect. Notes Comput. Sci.*, pages 400–417. Springer, 2019.

27 Joshua Moerman and Jurriaan Rot. Separation and Renaming in Nominal Sets. In *Proc. 28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, volume 152 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020.

28 Frank Neven, Thomas Schwentick, and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Logic*, 5(3):403–435, 2004.

29 Daniela Petrişan. *Investigations into Algebra and Topology over Nominal Sets*. PhD thesis, University of Leicester, 2012.

30 Andrew M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*, volume 57 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2013.

31 Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.

32 Lutz Schröder, Dexter Kozen, Stefan Milius, and Thorsten Wißmann. Nominal automata with name binding. In *Proc. 20th International Conference on Foundations of Software Science and Computation Structures, (FOSSACS 2017)*, volume 10203 of *Lect. Notes Comput. Sci.*, pages 124–142, 2017.

33 Alexandra Silva, Filippo Bonchi, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Generalizing determinization from automata to coalgebras. *Log. Methods Comput. Sci.*, 9(1:9), 2013.

34 Ian Stark. Categorical models for local names. *LISP Symb. Comput.*, 9(1):77–107, 1996.

35 A. Tal. Decidability of inclusion for unification based automata. Master's thesis, Department of Computer Science, Technion – Israel Institute of Technology, 1999.

A Appendix

This Appendix provides proof details and additional explanations omitted for lack of space.

Proof of Proposition 2.5

Let $A = (Q, \delta, I, F)$ be a NOFRA. Given a word $w \in L(A)$ with accepting run

$$(j_0, q_0) \xrightarrow{a_1} (j_1, q_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, q_n)$$

and a renaming $\rho\colon \mathbb{A} \to \mathbb{A}$, we have the accepting run

$$(j_0, \rho^\star q_0) \xrightarrow{\rho a_1} (j_1, \rho^\star q_1) \xrightarrow{\rho a_2} \cdots \xrightarrow{\rho a_n} (j_n, \rho^\star q_n)$$

by $\mathsf{Fin}(\mathbb{A})$-equivariance of δ, I, F. Hence $\rho^\star(w) \in L(A)$, so $L(A)$ is $\mathsf{Fin}(\mathbb{A})$-equivariant.

Proof of Proposition 2.8

Our task is to prove $L(\bar{A}) = \overline{L(A)}$.

($\supseteq$) We have $L(A) \subseteq L(\bar{A})$ because the NOFA A is a sub-NOFA of $\bar{A}$. Moreover, the language $L(\bar{A})$ is positive by Proposition 2.5, so $\overline{L(A)} \subseteq L(\bar{A})$.

($\subseteq$) We prove that for every run

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, q_n) \tag{A.1}$$

in $\bar{A}$, there exists a renaming $\rho\colon \mathbb{A} \to \mathbb{A}$ and a run

$$(j_0, p_0) \xrightarrow{a_1} (j_1, p_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, p_n) \tag{A.2}$$

in A such that $\rho^\star p_i = q_i$ for $i = 0, \ldots, n$ and $\rho a_i = b_i$ for $i = 1, \ldots, n$. Note that if (A.1) is accepting then so is (A.2); therefore $L(\bar{A}) \subseteq \overline{L(A)}$. We construct (A.2) by induction on n.

Induction base ($n = 0$). Choose $p_0 \in \mathbb{A}^{\#m}$ arbitrary and a renaming $\rho\colon \mathbb{A} \to \mathbb{A}$ mapping each letter of p_0 to the corresponding letter of $q_0 \in \mathbb{A}^m$. Then $\rho^\star q_0 = p_0$, as required.

Induction step ($n \to n + 1$). Suppose that

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, q_n) \xrightarrow{b_{n+1}} (j_{n+1}, q_{n+1})$$

is a run in $\bar{A}$. By induction, we know that there exists a renaming $\rho\colon \mathbb{A} \to \mathbb{A}$ and a run

$$(j_0, p_0) \xrightarrow{a_1} (j_1, p_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, p_n)$$

in A such that $\rho^\star p_i = q_i$ for $i = 0, \ldots, n$ and $\rho a_i = b_i$ for $i = 1, \ldots, n$. Furthermore, since $(j_n, q_n) \xrightarrow{b_{n+1}} (j_{n+1}, q_{n+1})$ in $\bar{A}$, there exists a renaming $\sigma\colon \mathbb{A} \to \mathbb{A}$ and a transition $(j_n, p'_n) \xrightarrow{a_{n+1}} (j_{n+1}, p_{n+1})$ in A such that $\sigma^\star p'_n = q_n$, $\sigma^\star p_{n+1} = q_{n+1}$ and $\sigma a_{n+1} = b_{n+1}$. We show below that we can choose this transition in such a way that (1) $p'_n = p_n$, (2) all names in $\{a_{n+1}\} \cup \mathsf{supp}(p_{n+1})$ that are fresh for p_n are fresh for $p_0, \ldots, p_n$ and $a_1, \ldots, a_n$, and (3) $\rho = \sigma$. Then, by (1) and (3), we obtain the run

$$(j_0, p_0) \xrightarrow{a_1} (j_1, p_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, p_n) \xrightarrow{a_{n+1}} (j_{n+1}, p_{n+1})$$

in A with the required properties. It remains to show how to enforce (1), (2), (3).

Ad (1). Since $p_n, p'_n \in \mathbb{A}^{\#m}$, there exists a permutation $\pi \in \mathsf{Perm}(\mathbb{A})$ such that $\pi \cdot p'_n = p_n$. Then, by equivariance, we have the transition

$$p_n = \pi \cdot p'_n \xrightarrow{\pi \cdot a_{n+1}} \pi \cdot p_{n+1}$$

in A, and

$$(\sigma \circ \pi^{-1})^\star(p_n) = q_n, \quad (\sigma \circ \pi^{-1})^\star(\pi \cdot p_{n+1}) = q_{n+1}, \quad (\sigma \circ \pi^{-1}) \cdot \pi \cdot a_{n+1} = b_{n+1}.$$

Thus (1) holds after replacing $p'_n, a_{n+1}, p_{n+1}, \sigma$ with $p_n, \pi \cdot a_{n+1}, \pi \cdot p_{n+1}, \sigma \circ \pi^{-1}$.

Ad (2). Suppose that (1) holds. Let $c_1, \dots, c_k$ be the names in $\{a_{n+1}\} \cup \mathsf{supp}(p_{n+1})$ that are fresh for p_n. Choose names $d_1, \dots, d_k$ fresh for $p_0, \dots, p_n$, $a_1, \dots, a_n, c_1, \dots, c_k$. Then, by equivariance, the permutation $\pi = (c_1\, d_1) \cdots (c_k\, d_k)$ yields the transition

$$(j_n, p_n) = (j_n, \pi \cdot p_n) \xrightarrow{\pi \cdot a_{n+1}} (j_{n+1}, \pi \cdot p_{n+1})$$

By definition of π, the names in $\{\pi \cdot a_{n+1}\} \cup \mathsf{supp}(\pi \cdot p_{n+1}) = \{\pi \cdot a_{n+1}\} \cup \pi \cdot \mathsf{supp}(p_{n+1})$ that are fresh for $p_n = \pi \cdot p_n$ are precisely $d_1, \dots, d_k$, and thus are fresh for $p_0, \dots, p_n$ and $a_1, \dots, a_n$. Thus (1) and (2) hold after replacing $p_n, a_{n+1}, p_{n+1}, \sigma$ with $p_n, \pi \cdot a_{n+1}, \pi \cdot p_{n+1}, \sigma \circ \pi^{-1}$.

Ad (3). Finally, suppose that (1) and (2) hold. Choose a renaming $\tau \colon \mathbb{A} \to \mathbb{A}$ that agrees with ρ on $\{a_1, \dots, a_n\} \cup \mathsf{supp}(p_0) \cup \dots \cup \mathsf{supp}(p_n)$ and with σ on $\{a_{n+1}\} \cup \mathsf{supp}(p_n) \cup \mathsf{supp}(p_{n+1})$. Such τ exists by (2) and because $\rho^\star(p_n) = q_n = \sigma^\star(p_n)$ implies that ρ and σ agree on $\mathsf{supp}(p_n)$. Thus, after replacing ρ and σ with τ, all three conditions (1), (2), (3) hold.

Proof of Proposition 2.12

($\Rightarrow$) Suppose that $(j, q) \xrightarrow{b} (j', q')$ in $\bar{A}$. By definition of $\bar{\delta}$, this means that there exists a transition $(j, p) \xrightarrow{a} (j', p')$ in A and a renaming $\rho \colon \mathbb{A} \to \mathbb{A}$ such that $\rho^\star p = q$, $\rho^\star p' = q'$, $\rho a = b$. Then the induced abstract transition (j, E, j') lies in $\mathsf{abs}(\delta)$, i.e. $j \xrightarrow{E} j'$, and the triple $((j, q), b, (j', q'))$ is consistent with it. Indeed, if $k = \bullet$ in E then $p_k = a$, hence $q_k = \rho p_k = \rho a = b$. Similarly for equations $\bullet = k$ and $k = \bar{k}$ in E.

($\Leftarrow$) Suppose that the triple $((j, q), b, (j', q'))$ is consistent with some $j \xrightarrow{E} j'$. Choose a transition $(j, p) \xrightarrow{a} (j, p')$ in A inducing the abstract transition $j \xrightarrow{E} j'$, and a renaming $\rho \colon \mathbb{A} \to \mathbb{A}$ mapping p_k to q_k, p'_k to q'_k and a to b. (Note that a well-defined choice of ρ is possible: If $p_k = a$ then $k = \bullet$ in E and hence $q_k = b$ by consistency. Similarly, $a = p'_k$ implies $b = q'_k$ and $p_k = p'_{\bar{k}}$ implies $q_k = q'_{\bar{k}}$.) Since $\rho^\star p = q$, $\rho^\star p' = q'$ and $\rho a = b$, we conclude that $(j, q) \xrightarrow{b} (j', q')$ in $\bar{A}$.

Proof of Proposition 2.15

We start with a remark and a technical lemma:

▶ **Remark A.1.** The abstract transition (j, E, j') induced by $((j, p), a, (j, p')) \in Q \times \mathbb{A} \times Q$ contains (i) at most one equation $k = \bullet$, (ii) at most one equation $\bullet = k$, (iii) for each k at most one equation $k = \bar{k}$, (iv) for each $\bar{k}$ at most one equation $k = \bar{k}$. Indeed, since $p, p' \in \mathbb{A}^{\#m}$ every data value occurs at most once in p or p', respectively. Moreover if E contains any two of the equations $k = \bullet$, $\bullet = \bar{k}$, $k = \bar{k}$, then it contains the third one.

▶ **Lemma A.2.** *For every abstract transition $j \xrightarrow{E} j'$ in $\mathsf{abs}(\delta)$ and $q \in \mathbb{A}^m$, there exists a transition $(j, q) \xrightarrow{b} (j', q')$ in $\bar{A}$ consistent with it.*

Proof. Choose a transition $(j,p) \xrightarrow{a} (j',p')$ in A inducing the abstract transition $j \xrightarrow{E} j'$, and let $\rho\colon \mathbb{A} \to \mathbb{A}$ be a renaming sending p_k to q_k for each $k \in \{1,\ldots,m\}$; hence $\rho^\star p = q$. Then, putting $b = \rho a$ and $q' = \rho^\star p'$, we obtain the transition $(j,q) \xrightarrow{b} (j',q')$ in $\bar{A}$ consistent with $j \xrightarrow{E} j'$. ◀

Now we prove Proposition 2.15.

($\Rightarrow$) Suppose that the word $b_1 \cdots b_n \in \mathbb{A}^\star$ is accepted by $\bar{A}$ via the accepting run

$$(j_0,q_0) \xrightarrow{b_1} (j_1,q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n,q_n).$$

By Proposition 2.12, each transition $(j_{r-1},q_{r-1}) \xrightarrow{b_r} (j_r,q_n)$ is consistent with some abstract transition $j_{r-1} \xrightarrow{E_r} j_r$ in $\mathsf{abs}(\delta)$. Thus each transition of the above run is consistent with the transitions of the abstract run

$$j_0 \xrightarrow{E_1} j_1 \xrightarrow{E_2} \cdots \xrightarrow{E_n} j_n.$$

Then by definition of the predicates $\mathsf{Eq}_k^{(i)}$, the condition (2.1) holds.

($\Leftarrow$) Suppose that there exists an accepting abstract run

$$j_0 \xrightarrow{E_1} j_1 \xrightarrow{E_2} \cdots \xrightarrow{E_n} j_n$$

such that (2.1) holds. We show that for each $r = 0,\ldots,n$ there exists a run

$$(j_0,q_0) \xrightarrow{b_1} (j_1,q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_r} (j_r,q_r) \tag{A.3}$$

whose transitions are consistent with the first r abstract transitions of the abstract run; in particular, putting $r = n$ this proves that $\bar{A}$ accepts $b_1 \cdots b_n$.

The run is constructed by induction on r. For $r = 0$, any choice of $q_0 \in \mathbb{A}^m$ will do. For $r = 1$ choose the transition $(j_0, b_1^m) \xrightarrow{b_1} (j_1, b_1^m)$, which is trivially consistent with $j_0 \xrightarrow{E_1} j_1$. Thus suppose that $0 < r < n$ and that a consistent run (A.3) of length r has been constructed. By Lemma A.2 there exists a transition $(j_r,q_r) \xrightarrow{b} (j_{r+1},q_{r+1})$ in $\bar{A}$ consistent with $j_r \xrightarrow{E_{r+1}} j_{r+1}$. We show how to turn the run

$$(j_0,q_0) \xrightarrow{b_1} (j_1,q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_r} (j_r,q_r) \xrightarrow{b} (j_{r+1},q_{r+1})$$

into a run for the word $b_1 \ldots b_r b_{r+1}$ satisfying the required consistency property. This requires a case distinction depending on the equations occurring in E_{r+1}:

<u>Case 1:</u> $k = \bullet$ in E_{r+1} for some k.

<u>Subcase 1.1:</u> $\mathsf{Eq}_k^{(i)}(r)$ for some i.
Note that necessarily $i \leq r$ by definition of $\mathsf{Eq}_k^{(i)}$. Then $b = q_{r,k} = b_i = b_{r+1}$: The first equality holds because the transition $(j_r,q_r) \xrightarrow{b} (j_{r+1},q_{r+1})$ is consistent with $k = \bullet$, the second one because $\mathsf{Eq}_k^{(i)}(r)$, and the third one by (2.1). We thus obtain the following consistent run for $b_1 \ldots b_r b_{r+1}$:

$$(j_0,q_0) \xrightarrow{b_1} (j_1,q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_r} (j_r,q_r) \xrightarrow{b_{r+1}} (j_{r+1},q_{r+1}).$$

<u>Subcase 1.2:</u> $\mathsf{Eq}_k^{(i)}(r)$ does not hold for any i.
Consider the unique $s \in \{0,\ldots,r\}$ and the unique $k_s, k_{s+1}, \ldots, k_r = k$ such that $k_{t-1} = k_t$ in E_t for $t \in \{s+1,\ldots,r\}$ and no equation $\bar{k} = k$ is contained in E_s (putting $E_0 = \emptyset$). Then

- for $t \in \{s, \dots, r\}$ one has $q_{k_t} = b$;
- for $t \in \{s+1, \dots, r\}$ one does not have $\mathsf{Eq}_{k_t}^{(i)}(t)$ for any i (for otherwise $\mathsf{Eq}_{k}^{(i)}(r)$ by definition of the predicates);
- for $t \in \{s, \dots, r\}$ the equation $\bullet = k_t$ is not contained in E_t (for otherwise $\mathsf{Eq}_{k_t}^{(t)}(t)$).
- for $t \in \{s+1, \dots, r\}$ the equation $k_{t-1} = \bullet$ is not contained in E_t (for otherwise $\bullet = k_t$ in E_t since $k_{t-1} = k_t$ in E_t).

For $t \in \{s, \dots, r\}$ let q'_t emerge from q_t by replacing the letter b at position k_t by the letter b_{r+1}. Then the triples $((j_{t-1}, q'_{t-1}), b_t, (j_t, q'_t))$ for $t \in \{s+1, \dots, r\}$ are consistent with (j_{t-1}, E_t, j_t), as is the triple $((j_{s-1}, q_{s-1}), b_s, (j_s, q'_s))$ if $s > 0$. It follows by Proposition 2.12 that $(j_t, q'_t) \xrightarrow{b_{t+1}} (j_{t+1}, q'_{t+1})$ and $(j_{s-1}, q_{s-1}) \xrightarrow{b_s} (j_s, q'_s)$ (if $s > 0$) are transitions in $\bar{A}$. Thus we obtain the consistent run

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_{s-1}} (j_{s-1}, q_{s-1}) \xrightarrow{b_s} (j_s, q'_s) \xrightarrow{b_{s+1}} \cdots \xrightarrow{b_r} (j_r, q'_r).$$

If $\bullet = \bar{k}$ in E_{r+1} for some $\bar{k}$, then let q'_{r+1} emerge from q_{r+1} by replacing the $\bar{k}$-th letter of q_{r+1} with b_{r+1}; otherwise put $q'_{r+1} = q_{r+1}$. Then the triple $((j_r, q'_r), b_{r+1}, (j_{r+1}, q'_{r+1}))$ is consistent with (j_r, E_{r+1}, j_{r+1}), so $(j_r, q'_r) \xrightarrow{b_{r+1}} (j_{r+1}, q'_{r+1})$ in $\bar{A}$ and thus

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \cdots (j_{s-1}, q_{s-1}) \xrightarrow{b_s} (j_s, q'_s) \xrightarrow{b_{s+1}} \cdots \xrightarrow{b_r} (j_r, q'_r) \xrightarrow{b_{r+1}} (j_{r+1}, q'_{r+1})$$

is a consistent run for $b_1 \dots b_r b_{r+1}$.

Case 2: No $k = \bullet$ in E_{r+1}.

Subcase 2.1: $\bullet = \bar{k}$ in E_{r+1} for some $\bar{k}$.
Let q'_{r+1} emerge from q_{r+1} by replacing the $\bar{k}$-th letter (viz. b) with b_{r+1}. It then follows that the triple $((j_r, q_r), b_{r+1}, (j_{r+1}, q'_{r+1}))$ is consistent with (j_r, E_{r+1}, j_{r+1}). (To see this, note that E_{r+1} does not contain an equation $k = \bar{k}$, for otherwise $k = \bullet$ in E_{r+1}.) Hence $((j_r, q_r) \xrightarrow{b_{r+1}} (j_{r+1}, q'_{r+1}))$ in $\bar{A}$ and we obtain the following consistent run for $b_1 \cdots b_r b_{r+1}$:

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_r} (j_r, q_r) \xrightarrow{b_{r+1}} (j_{r+1}, q'_{r+1}).$$

Subcase 2.2: No $\bullet = \bar{k}$ in E_{r+1}.
Since also no $k = \bullet$ in E_{r+1}, the triple $((j_r, q_r), b_{r+1}, (j_{r+1}, q_{r+1}))$ is consistent with (j_r, E_{r+1}, j_{r+1}). It follows that $((j_r, q_r) \xrightarrow{b_{r+1}} (j_{r+1}, q_{r+1}))$ in $\bar{A}$, which yields the following consistent run for $b_1 \cdots b_r b_{r+1}$:

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_r} (j_r, q_r) \xrightarrow{b_{r+1}} (j_{r+1}, q_{r+1}).$$

This concludes the proof.

Proof of Theorem 2.17

By Proposition 2.8 every positive NOFA-recognizable language is accepted by some NOFRA $\bar{A}$ as given by Construction 2.7. Therefore, it suffices to turn $\bar{A}$ into an equivalent non-guessing NOFRA. To this end, we first modify $\bar{A}$ in such a way that it keeps track of the set $S \subseteq \{1, \dots, m\}$ of those registers whose content is determined by previous abstract transitions of A, and modifies the content of registers outside that set arbitrarily.

▶ **Construction A.3.** Let $\bar{A} = (\bar{Q}, \bar{\delta}, \bar{I}, \bar{F})$ be a NOFRA as in Construction 2.7. Then the NOFRA $\widetilde{A} = (\widetilde{Q}, \widetilde{\delta}, \widetilde{I}, \widetilde{F})$ is given by

- states $\widetilde{Q} = J \times \mathcal{P}(\{1,\ldots,m\}) \times \mathbb{A}^m$, where $\mathcal{P}$ denotes the powerset;
- initial states $\widetilde{I} = J_I \times \{\emptyset\} \times \mathbb{A}^m$ and final states $\widetilde{F} = J_F \times \mathcal{P}(\{1,\ldots,m\}) \times \mathbb{A}^m$;
- transitions defined as follows: for a set E of equations and $S \subseteq \{1,\ldots,m\}$ let E_S denote the restriction of E to those equations whose left-hand side refers to a register in S:

$$E_S = \{k = \bullet \in E \,:\, k \in S\} \cup \{k = \bar{k} \in E \,:\, k \in S\}.$$

For $S, S' \subseteq \{1,\ldots,m\}$ we write $S \rightsquigarrow_E S'$ if

$$S' = \{k \in \{1,\ldots,m\} : \bullet = k \in E\} \cup \{\bar{k} \in \{1,\ldots,m\} : k = \bar{k} \in E \text{ for some } k \in S\}.$$

Given $((j,S,q), b, (j',S',q')) \in \widetilde{Q} \times \mathbb{A} \times \widetilde{Q}$ we have the transition $(j,\, S,\, q) \xrightarrow{b} (j',\, S',\, q')$ in $\widetilde{A}$ iff there exists some abstract transition $(j, E, j') \in \mathsf{abs}(\delta)$ such that (i) the triple $((j,q), b, (j',q'))$ is consistent with (j, E_S, j'), and (ii) $S \rightsquigarrow_E S'$.

▶ **Remark A.4.** Since property (i) only requires consistency with (j, E_S, j'), transitions can be modified arbitrarily outside of S and S': for every transition $(j,\, S,\, q) \xrightarrow{b} (j',\, S',\, q')$ of $\widetilde{A}$ one also has the transitions $(j,\, S,\, \overline{q}) \xrightarrow{b} (j',\, S',\, \overline{q}')$ for all $\overline{q}, \overline{q}' \in \mathbb{A}^m$ such that $q_k = \overline{q}_k$ for $k \in S$ and $\overline{q}'_k = q'_k$ for $k \in S'$.

▶ **Lemma A.5.** *The NOFRA $\bar{A}$ and $\widetilde{A}$ are equivalent.*

Proof. *$L(\bar{A}) \subseteq L(\widetilde{A})$:* Suppose that $b_1 \ldots b_n \in L(\bar{A})$ with accepting run

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, q_n)$$

in $\bar{A}$. Then for all $r < n$ the transition $(j_r, q_r) \xrightarrow{b_{r+1}} (j_{r+1}, q_{r+1})$ is consistent with some abstract transition $j_r \xrightarrow{E_{r+1}} j_{r+1}$ (Proposition 2.12). Hence it is also consistent with $(j_r, (E_{r+1})_S, j_{r+1})$ for every $S \subseteq \{1, \cdots, m\}$. It follows that we have the accepting run

$$(j_0, S_0, q_0) \xrightarrow{b_1} (j_1, S_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, S_n, q_n)$$

in $\widetilde{A}$ where $S_0 = \emptyset$ and $S_r \rightsquigarrow_{E_r} S_{r+1}$ for all $r < n$, whence $b_1 \ldots b_n \in L(\widetilde{A})$.

$L(\widetilde{A}) \subseteq L(\bar{A})$: Suppose that $b_1 \ldots b_n \in L(\widetilde{A})$ with accepting run

$$(j_0, S_0, q_0) \xrightarrow{b_1} (j_1, S_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, S_n, q_n)$$

in $\widetilde{A}$. By definition of the transitions of $\widetilde{A}$, for all $r < n$ there exists an abstract transition $j_r \xrightarrow{E_{r+1}} j_{r+1}$ of A such that $((j_r, q_r), b_{r+1}, (j_{r+1}, q_{r+1}))$ is consistent with $(j_r, (E_{r+1})_{S_r}, j_{r+1})$, and moreover $S_r \rightsquigarrow_{E_r} S_{r+1}$ (where $S_0 = \emptyset$). To prove $b_1 \ldots b_n \in L(\bar{A})$ we employ Proposition 2.15: we verify that the abstract run

$$j_0 \xrightarrow{E_1} j_1 \xrightarrow{E_2} \cdots \xrightarrow{E_n} j_n$$

with its associated predicates $\mathsf{Eq}_i^{(k)}$ satisfies property (2.1). Thus let $r < n$, $k = \bullet$ in E_{r+1} and $\mathsf{Eq}_k^{(i)}(r)$ for some k. By definition of $\mathsf{Eq}_k^{(i)}(r)$ and $\rightsquigarrow$, we have $b_i = (q_r)_k$ and $k \in S_r$. Since $k = \bullet \in (E_{r+1})_{S_r}$ and the triple $((j_r, q_r), b_{r+1}, (j_{r+1}, b_{r+1}))$ is consistent with $(j_r, (E_{r+1})_{S_r}, j_{r+1})$, it follows that $b_i = (q_r)_k = b_{r+1}$, as required. ◀

Theorem 2.17 now follows from the above lemma and the following one:

▶ **Lemma A.6.** *The NOFRA $\widetilde{A}$ is equivalent to a non-guessing NOFRA.*

Proof. We turn $\widetilde{A}$ into an equivalent non-guessing NOFRA $\widetilde{A}_{\mathsf{ng}}$ by first removing all guessing transitions, and then dealing with initial states with non-empty support. In more detail:

1. Let $\widetilde{A}_{\mathsf{ngt}}$ be the sub-NOFRA of $\widetilde{A}$ obtained by restricting to non-guessing transitions, i.e. transitions $(j,S,q) \xrightarrow{b} (j',S',q')$ where $\mathsf{supp}\, q' \subseteq \mathsf{supp}\, q \cup \{b\}$. We claim that $L(\widetilde{A}_{\mathsf{ngt}}) = L(\widetilde{A})$. The left-to-right inclusion is clear. For the right-to-left inclusion, suppose that $b_1 \cdots b_n \in L(\widetilde{A})$ with accepting run
$$(j_0,S_0,q_0) \xrightarrow{b_1} (j_1,S_1,q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n,S_n,q_n)$$
in $\widetilde{A}$. By Remark A.4 we obtain another accepting run
$$(j_0,S_0,\overline{q}_0) \xrightarrow{b_1} (j_1,S_1,\overline{q}_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n,S_n,\overline{q}_n)$$
where $\overline{q}_0 = b_1^m$ and for $r = 1,\dots,n$ we put $(\overline{q}_r)_k = (q_r)_k$ if $k \in S_r$ and $(\overline{q}_r)_k = b_{r-1}$ if $k \notin S_r$. Since all these transitions are non-guessing, this an accepting run in $\widetilde{A}_{\mathsf{ngt}}$, so $b_1 \cdots b_n \in L(\widetilde{A}_{\mathsf{ngt}})$.
2. Now let $\widetilde{A}_{\mathsf{ng}}$ emerge from $\widetilde{A}_{\mathsf{ngt}}$ by adding a new initial state q_0 with $\mathsf{supp}\, q_0 = \emptyset$ (which is also final if $J_I \cap J_F \neq \emptyset$), making all states of $\widetilde{A}_{\mathsf{ngt}}$ non-initial, and adding a transition $q_0 \xrightarrow{b} (j',S',q')$ for each transition $(j,\emptyset,b^m) \xrightarrow{b} (j',S',q')$ of $\widetilde{A}_{\mathsf{ngt}}$ where $j \in J_I$. The NOFRA $\widetilde{A}_{\mathsf{ng}}$ is non-guessing and satisfies $L(\widetilde{A}_{\mathsf{ng}}) = L(\widetilde{A}_{\mathsf{ngt}})$ by Remark A.4. ◀

Proof of Theorem 3.2

The "if" direction follows from the fact that every register automaton admits an equivalent NOFA [7] and from

▶ **Proposition A.7.** *Every positive register automaton accepts a positive language.*

Proof. Let A be a positive register automaton and $w = a_1 \dots a_n \in L(A)$ with accepting run
$$(c_0,r_0) \xrightarrow{a_1} (c_1,r_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (c_n,r_n).$$
For $i = 0,\dots,n-1$ we have that $(c_i,r_i) \xrightarrow{a_{i+1}} (c_{i+1},r_{i+1})$ is consistent with some transition $c_i \xrightarrow{\varphi_{i+1}} c'_{i+1}$. Then $(c_i,\rho^\star r_i) \xrightarrow{\rho a_{i+1}} (c_{i+1},\rho^\star r_{i+1})$ for every renaming $\rho\colon \mathbb{A} \to \mathbb{A}$ since this is also consistent with $c_i \xrightarrow{\varphi_{i+1}} c'_{i+1}$. Hence A accepts $\rho^\star w = \rho a_1 \dots \rho a_n$ via the run
$$(c_0,\rho^\star r_0) \xrightarrow{\rho a_1} (c_1,\rho^\star r_1) \xrightarrow{\rho a_2} \cdots \xrightarrow{\rho a_n} (c_n,\rho^\star r_n).$$
This proves $\rho^\star w \in L(A)$, showing that $L(A)$ is a positive language. ◀

For the "only if" direction, suppose that $L \subseteq \mathbb{A}^\star$ is a positive NOFA-recognizable language. Then L is accepted by a NOFA of the form $\bar{A} = (\bar{Q},\bar{\delta},\bar{I},\bar{F})$, cf. Construction 2.7, in particular $\bar{Q} = J \times \mathbb{A}^{\#m}$ for some J and m. We regard an equation as per Definition 2.11.1 as an equation in Φ by identifying
$$k = \bullet \;\leftrightarrow\; (k,\text{before}) = \bullet, \quad \bullet = k \;\leftrightarrow\; \bullet = (k,\text{after}), \quad k = \bar{k} \;\leftrightarrow\; (k,\text{before}) = (\bar{k},\text{after}),$$
and turn $\bar{A}$ into a positive register automaton $\bar{A}_{\text{reg}} = (J_{\text{reg}}, m, \delta_{\text{reg}}, \{j_{0,\text{reg}}\}, F_{\text{reg}})$ as follows:

- The set of control states is $J_{\text{reg}} = J \cup \{j_{0,\text{reg}}\}$ where $j_{0,\text{reg}} \notin J$;
- $j_{0,\text{reg}}$ is the only initial state;
- every state in J_F is final; additionally $j_{0,\text{reg}}$ is final if $J_I \cap J_F \neq \emptyset$;

- for every abstract transition $j \xrightarrow{E} j'$ of A, the automaton $\bar{A}_{\mathrm{reg}}$ contains the transition $j \xrightarrow{\bigwedge E} j'$, where $\bigwedge E$ is the conjunction of all equations in E (note that $\bigwedge \emptyset = \mathrm{true}$);
- for every abstract transition $j_0 \xrightarrow{E} j$ of A where $j_0 \in J_I$, the automaton $\bar{A}_{\mathrm{reg}}$ contains the transition $j_{0,\mathrm{reg}} \xrightarrow{\varphi_E} j$ where $\varphi_E = (\bullet = k)$ if $\bullet = k$ in E, and $\varphi_E = \mathrm{true}$ otherwise.

We claim that $L = L(\bar{A}_{\mathrm{reg}})$. For the inclusion ($\subseteq$), let $w = b_1 \dots b_n \in L = L(\bar{A})$. Then in $\bar{A}$ we have an accepting run

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, q_n).$$

whose transitions are consistent with some accepting abstract run

$$j_0 \xrightarrow{E_1} j_1 \xrightarrow{E_2} \cdots \xrightarrow{E_n} j_n.$$

It follows that the register automaton $\bar{A}_{\mathrm{reg}}$ admits the transitions

$$j_{0,\mathrm{reg}} \xrightarrow{\varphi_{E_1}} j_1 \xrightarrow{\bigwedge E_2} \cdots \xrightarrow{\bigwedge E_n} j_n$$

and that

$$(j_{0,\mathrm{reg}}, \bot^m) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, q_n)$$

is an accepting run consistent with them. Therefore $w \in L(\bar{A}_{\mathrm{reg}})$.

For the inclusion ($\supseteq$), let $w = b_1 \cdots b_n \in L(\bar{A}_{\mathrm{reg}})$ with accepting run

$$(j_{0,\mathrm{reg}}, \bot^m) \xrightarrow{b_1} (j_1, r_1) \xrightarrow{b_2} \cdots \xrightarrow{n_n} (j_n, r_n),$$

in $\bar{A}_{\mathrm{reg}}$, where $r_i \in (\mathbb{A} \cup \{\bot\})^m$. Note that $j_1, \dots, j_n \in J$. By definition of the transitions of $\bar{A}_{\mathrm{reg}}$, there exists an abstract transition $j_0 \xrightarrow{E_1} j_1$ of A such that $(j_{0,\mathrm{reg}}, \bot^m) \xrightarrow{b_1} (j_1, r_1)$ is consistent with the transition $j_{0,\mathrm{reg}} \xrightarrow{\varphi_E} j_1$ of $\bar{A}_{\mathrm{reg}}$, and for $i = 1, \dots, n$ there exists an abstract transition $j_{i-1} \xrightarrow{E_i} j_i$ of $\bar{A}$ such that $(j_{i-1}, r_{i-1}) \xrightarrow{b_i} (j_i, r_i)$ is consistent with the transition $j_{i-1} \xrightarrow{\bigwedge E_i} j_i$ of $\bar{A}_{\mathrm{reg}}$. Now choose $q_0, \dots, q_n \in \mathbb{A}^m$ as follows:

- For $i = 1, \dots, n$ choose q_i such that $q_{i,k} = r_k$ whenever $r_k \neq \bot$.
- Choose q_0 such that $q_{0,k} = q_{1,\bar{k}}$ if $k = \bar{k}$ in E_1, and $q_{0,k} = b_1$ if $k = \bullet$ in E_1.

Then for each $i = 1, \dots, n$ we have the transition $(j_{i-1}, q_{i-1}) \xrightarrow{b_i} (j_i, q_i)$ in $\bar{A}$, as it consistent with the abstract transition $j_{i-1} \xrightarrow{E_i} j_i$. Therefore

$$(j_0, q_0) \xrightarrow{b_1} (j_1, q_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (j_n, q_n).$$

is an accepting run in $\bar{A}$, showing that $w \in L(\bar{A}) = L$.

Details for Remark 3.3

We provide more details on the stated equivalence between positive register automata and a a version finite-state unification-based automata (FSUBA) [20, 35]. We first recall the definition of the latter.

▶ **Notation A.8.** For any natural number r, we denote by $\underline{r}$ the set of all natural numbers between 1 and r, inclusively. $\underline{0}$ denotes the empty set. We denote by $\mathsf{Perm}(\underline{r})$ the group of all permutations on the finite set $\underline{r}$ and note that there is an obvious group action of $\mathsf{Perm}(\underline{r})$ on $\mathbb{A}^{\#r}$ that is defined as follows: For any $\pi \in \mathsf{Perm}(\underline{r})$ and $w \in \mathbb{A}^{\#r}$, we define the word $\pi \star w \in \mathbb{A}^{\#r}$ by $(\pi \star w)_k = w_{\pi(k)}$ for $k \in \underline{r}$. Note that this action is compatible with the $\mathsf{Perm}(\mathbb{A})$-action: $\pi \star (\rho \cdot w) = \rho \cdot (\pi \star w)$ for $\rho \in \mathsf{Perm}(\mathbb{A})$.

▶ **Definition A.9.** A *finite-state unification-based automaton (FSUBA)* is a quintuple $A = (Q, m, \mu, q_0, F)$ where Q is a finite set of control states, $m \in \mathbb{N}$ is the number of registers (numbered from 1 to m), q_0 is the initial state, $F \subseteq Q$ the set of final states, and $\mu \subseteq Q \times \underline{m} \times \mathcal{P}(\underline{m}) \times Q$ is the transition relation. Here $\mathcal{P}$ denotes the powerset. A *configuration* of A is a pair (q, w) of a state $q \in Q$ and a word $w \in (\mathbb{A} \cup \{\bot\})^m$ corresponding to a partial assignment of data values to the registers. The initial configuration is $(q_0, \bot^m)$, final configurations are all (q_f, w) with $q_f \in F$. We let Q^c and F^c denote the sets of configurations and final configurations, respectively. Given an input $a \in \mathbb{A}$ and configurations $(q, w), (q', w')$ we write $(q, w) \xrightarrow{a} (q', w')$ if this move is *consistent* with some transition (q, k, T, q'), which means that the following conditions are satisfied: (i) $w_k \in \{\bot, a\}$; (ii) $k \notin T \implies w'_k = a$; (iii) $\forall j \in T.\ w_j = \bot$; and (iv) $\forall j \notin T \cup \{k\}.\ w'_j = w_j$. We denote the induced move relation on $Q^c \times \mathbb{A} \times Q^c$ by μ^c. A word $a_1 \cdots a_n \in \mathbb{A}^\star$ is *accepted* by A if there exists an accepting run for it, viz. a sequence of configurations $(q_0, \bot^m) \xrightarrow{a_1} (q_1, w_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (q_n, w_n)$, where $q_n \in F$. We write $L(A) \subseteq \mathbb{A}^\star$ for the language of accepted words.

▶ **Remark A.10.** In comparison to the original definition of Tal [20, 35]) we do not allow an initial assignment of the registers, since otherwise the accepted languages are not equivariant but only finitely supported. Doing this also suppresses the 'read-only' alphabet, a subset of the data values occurring in the initial assignment.

▶ **Remark A.11.** Every FSUBA $A = (Q, m, \mu, q_0, F)$ can be translated into an expressively equivalent NOFA $N = (Q^c, \mu^c, \{(q_0, \bot^m)\}, F^c)$, i.e. the configurations of the FSUBA are simply regarded as states of a NOFA. The corresponding NOFA has the set of configurations Q^c as states, the singleton set $\{(q_0, \bot^m)\}$ as initial states, and F^c as final states. An accepting run of A is then precisely an accepting run of N.

The structural difference between NOFA and FSUBAs is the inherent 'non-guessing' of FSUBAs and the fact that FSUBAs cannot move data values from one register to another; e.g., if register 2 contains the data value a, then it cannot be moved to register 3 in the next step, which is possible with a NOFA. The first issue will be fixed by use of Theorem 2.17, while for the second we will turn a NOFA A given by Remark 2.6 into a *rigid* NOFA, where the data value contained in a register is never moved to another register:

▶ **Definition A.12.** *A NOFA $A = (Q, \delta, I, F)$ with states $Q = J \times \mathbb{A}^{\#m}$ is* rigid *if for every transition $(j, p) \xrightarrow{a} (j', p')$ and for every $b \in \mathsf{supp}\, p \cap \mathsf{supp}\, p'$, there exists $k \in \underline{m}$ such that $p_k = b = q_k$.*

▶ **Remark A.13.** Hence rigid NOFA are those whose abstract transitions are of the form $\bullet = k$, $k = \bullet$, or $k = k$. The construction below turns any NOFA into a rigid one. The idea is to keep track, via the control state, which data values have changed their register.

▶ **Construction A.14.** Let $A = (Q, \delta, I, F)$ be a NOFA with states $Q = J \times \mathbb{A}^{\#m}$. We construct the rigid NOFA $A_{\mathsf{rg}} = (Q_{\mathsf{rg}}, \delta_{\mathsf{rg}}, I_{\mathsf{rg}}, F_{\mathsf{rg}})$ given by

- states $Q_{\mathsf{rg}} = J \times \mathsf{Perm}(\underline{m}) \times \mathbb{A}^{\#m}$;
- initial states $I_{\mathsf{rg}} = J_I \times \{\mathsf{id}_{\underline{m}}\} \times \mathbb{A}^{\#m}$ and final states $F_{\mathsf{rg}} = J_F \times \mathsf{Perm}(\underline{m}) \times \mathbb{A}^{\#m}$;

- transitions defined as follows: Given $((j,\,\pi,\,p)\,,\,a,\,(j',\,\pi',\,p')) \in Q_{\mathsf{rg}} \times \mathbb{A} \times Q_{\mathsf{rg}}$ we have the transition $(j,\,\pi,\,p) \xrightarrow{a} (j',\,\pi',\,p')$ in A_{rg} iff (i) $(j,\,\pi \star p) \xrightarrow{a} (j',\,\pi' \star p')$ is a transition in A and (ii) for every $b \in \mathsf{supp}(p) \cap \mathsf{supp}(p')$ there exists $k \in \underline{m}$, such that $p_k = b = p'_k$.

  Note that by property (ii) of transitions, the NOFA A_{rg} is rigid.

▶ **Lemma A.15.** *The NOFA A and A_{rg} are equivalent.*

Proof. $L(A_{\mathsf{rg}}) \subseteq L(A)$*:* Every accepting run

$$(j_0, \pi_0, p_0) \xrightarrow{a_1} (j_1, \pi_1, p_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, \pi_n, p_n)$$

of A_{rg} yields the following accepting run of A:

$$(j_0, \pi_0 \star p_0) \xrightarrow{a_1} (j_1, \pi_1 \star p_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, \pi_n \star p_n).$$

$L(A) \subseteq L(A_{\mathsf{rg}})$*:* Given $a_1 \dots a_n \in L(A)$ with an accepting run

$$(j_0,\, q_0) \xrightarrow{a_1} \cdots \xrightarrow{a_n} (j_n,\, q_n)$$

in A, we inductively construct an accepting run

$$(j_0,\, \pi_0,\, q'_0) \xrightarrow{a_1} \cdots \xrightarrow{a_n} (j_n,\, \pi_n,\, q'_n)$$

in A_{rg} such that $\pi_r \star q'_r = q_r$ for $r = 0, \dots, n$. We put $(j_0,\, \pi_0,\, q'_0) = \big(j_0,\, \mathsf{id}_{\underline{m}}\,,\, q_0\big)$, which clearly fulfills $\pi_0 \star q'_0 = q_0$. Now suppose that that $0 \le r < n$ and that the first r transitions with the required properties have been constructed. Then we construct the next transition $(j_r,\, \pi_r,\, q'_r) \xrightarrow{a_{r+1}} \big(j_{r+1},\, \pi_{r+1},\, q'_{r+1}\big)$ as follows. Let $\mathcal{I} := \{k \in \underline{m} \,:\, q'_{r,k} \in \mathsf{supp}(q_{r+1})\}$. Choose $q'_{r+1} \in \mathbb{A}^{\#m}$ such that $\mathsf{supp}\, q'_{r+1} = \mathsf{supp}\, q_{r+1}$ and $q'_{r+1,k} = q'_{r,k}$ for $k \in \mathcal{I}$. Choose moreover a permutation $\pi_{r-1} \in \mathsf{Perm}(\underline{m})$ such that $\pi \star q'_{r+1} = q_{r+1}$. Note that since also $\mathsf{supp}\, q'_r = \mathsf{supp}\, q_r$, every data value in $\mathsf{supp}\, q'_r \cap \mathsf{supp}\, q'_{r+1}$ is equal to $q'_{r,k}$ for some $k \in \mathcal{I}$. Therefore $(j_r,\, \pi_r,\, q'_r) \xrightarrow{a_{r+1}} \big(j_{r+1},\, \pi_{r+1},\, q'_{r+1}\big)$ is a transition of A_{rg}, as required. ◀

▶ **Remark A.16.** If we apply the construction $A \mapsto \widetilde{A}_{\mathsf{ng}}$ of Lemma A.5 and Lemma A.6 to a rigid NOFA, we see that for every transition $(j,\, S,\, p) \xrightarrow{a} (j',\, S',\, q)$ the set S' is of the form $S \setminus T$ or $(S \setminus T) \cup \{k\}$ where $T \subseteq S$ and $k \in \underline{m}$. This follows directly from Remark A.13.

We will now show how to translate $\widetilde{A}_{\mathsf{ng}}$ into an equivalent FSUBA. The idea is to maintain in the control state, in addition to the set S of "relevant" registers, a subset $R \subseteq S$ containing all registers that will eventually be compared with a future input value. Registers outside of R then may be deleted.

▶ **Construction A.17.** Let $A = (Q,\, \delta,\, I,\, F)$ be a rigid NOFA and $\widetilde{A}_{\mathsf{ng}} = \Big(\widetilde{Q},\, \widetilde{\delta}_{\mathsf{ng}},\, \{q_0\},\, \widetilde{F}_{\mathsf{ng}}\Big)$ be the corresponding non-guessing NOFRA of Lemma A.6 with states $J \times \mathcal{P}(\underline{m}) \times \mathbb{A}^m \cup \{q_0\}$. Then the FSUBA $A_{\mathsf{fsuba}} = (C,\, m+1,\, \mu,\, q_0,\, F_{\mathsf{fsuba}})$ with $m+1$ registers is given as follows:

- control states $C = \{q_0\} \cup \{(j, R, S) : j \in J \text{ and } R \subseteq S \subseteq \underline{m}\}$;
- all states (j, R, S) where $j \in J_F$ are final, and additionally q_0 is final if $J_I \cap J_F \neq \emptyset$;
- for each abstract transition $j \xrightarrow{E} j'$ in $\mathsf{abs}(\delta)$ and all pairs of sets $R, S \subseteq \underline{m}$ such that $R \subseteq S$ and $k = \bullet \in E_S$ implies $k \in R$, we have the following transitions:
  1. if $k = \bullet$ in E_S (hence E_R) or $\bullet = k$ in E for some (unique) $k \in \underline{m}$, then μ contains the transitions

$$((j, R, S), k, \overline{R'}, (j', R', S')) \qquad \text{and} \qquad ((j, R, S), k, \overline{R' \setminus \{k\}}, (j', R' \setminus \{k\}, S')),$$

  where $R \rightsquigarrow_E R'$, $S \rightsquigarrow_E S'$, and $\overline{T} = \underline{m+1} \setminus T$ for any subset $T \subseteq \underline{m+1}$.

2. otherwise, μ contains the transition

$$((j,R,S),m+1,\overline{R'},(j',R',S')).$$

- Additionally, for every transition $((j_0,\emptyset,\emptyset),k,T,(j,R,S))$ where $j_0 \in J_I$, $k \in \underline{m+1}$ and $R,S,T \subseteq \underline{m+1}$, we have the transition $(q_0,k,T,(j,R,S))$.

▶ **Lemma A.18.** *The automata $\widetilde{A}_{\mathsf{ng}}$ and A_{fsuba} are equivalent.*

Proof. $L(\widetilde{A}_{\mathsf{ng}}) \subseteq L(A_{\mathsf{fsuba}})$*:* Let $a_1 \cdots a_n \in L(\widetilde{A}_{\mathsf{ng}}) = L(\widetilde{A})$ with accepting run

$$(j_0,S_0,v_0) \xrightarrow{a_1} (j_1,S_1,v_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n,S_n,v_n)$$

in $\widetilde{A}$. Each transition $(j_{r-1},S_{r-1},v_{r-1}) \xrightarrow{a_r} (j_r,S_r,v_r)$, $r=1,\dots,n$, is induced by some abstract transition $j_{r-1} \xrightarrow{E_r} j_r$ in $\mathsf{abs}(\delta)$, that is, the triple $((j_{r-1},v_{r-1}),a_r,(j_r,v_r))$ is consistent with $(j_{r-1},(E_r)_{S_{r-1}},j_r)$ and $S_{r-1} \rightsquigarrow S_r$. We now construct an accepting run

$$(q_0,w_0) \xrightarrow{a_1} ((j_1,R_1,S_1),w_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} ((j_n,R_n,S_n),w_n)$$

in A_{fsuba} whose data is defined as follows for $r=1,\dots,n$:

- R_r is the set of registers in S_r that will be compared with some later input, that is,

$$R_r = \{\, k \in S_r : \exists s \in \{r+1,\dots,n\} : k = k \in E_{r+1},\dots E_{s-1} \text{ and } k = \bullet \in E_s \,\}.$$

- $w_0 = \bot^{m+1}$, and $w_{r,k} = v_{r,k}$ for $k \in R_r$ and $w_{r,k} = \bot$ for $k \in \overline{R_r}$.

Note that the first move is equivalent to having a move $((j_0,R_0,S_0),w_0) \xrightarrow{a_1} ((j_1,R_1,S_1),w_1)$ where $j_0 \in J_I$ and $S_0 = R_0 = \emptyset$. Let us now verify that $((j_{r-1},R_{r-1},S_{r-1}),w_{r-1}) \xrightarrow{a_r} ((j_r,R_r,S_r),w_r)$ is indeed a valid move for $r=1,\dots,n$, i.e. consistent with some transition of A_{fsuba}. We distinguish two cases:

- If $k = \bullet$ in $(E_r)_{S_{r-1}}$ or $\bullet = k$ in E_r for some $k \in \underline{m}$, take the transition

$$((j_{r-1},R_{r-1},S_{r-1}),k,\overline{R_r},(j_r,R_r,S_r)). \tag{A.4}$$

  Note that this transition is induced by the abstract transition $j_{r-1} \xrightarrow{E_r} j_r$ and the pair R_{r-1},S_{r-1} as per Construction A.17: if $R_{r-1} \rightsquigarrow_{E_r} R'$ then either $R_r = R'$ or $R_r = R' \backslash \{k\}$ by definition of the sets R_{r-1} and R_r, and moreover if $k = \bullet$ in $(E_r)_{S_{r-1}}$ then $k \in R_{r-1}$ by definition of R_{r-1}. The move $((j_{r-1},R_{r-1},S_{r-1}),w_{r-1}) \xrightarrow{a_r} ((j_r,R_r,S_r),w_r)$ clearly satisfies the consistency conditions (i)–(iv) of Definition A.9 w.r.t. (A.4).
- Otherwise, take the transition

$$((j_{r-1},R_{r-1},S_{r-1}),m+1,\overline{R_r},(j_r,R_r,S_r)). \tag{A.5}$$

  Note again that this transition is induced by the abstract transition $j_{r-1} \xrightarrow{E_r} j_r$ and the pair R_{r-1},S_{r-1} as per Construction A.17: one has $R_{r-1} \rightsquigarrow R_r$ by definition of the sets R_{r-1} and R_r. The move $((j_{r-1},R_{r-1},S_{r-1}),w_{r-1}) \xrightarrow{a_r} ((j_r,R_r,S_r),w_r)$ clearly satisfies the consistency conditions (i)–(iv) of Definition A.9 w.r.t. (A.5).

$L(A_{\mathsf{fsuba}}) \subseteq L(\widetilde{A}_{\mathsf{ng}})$: Let $a_1 \cdots a_n \in L(A_{\mathsf{fsuba}})$ with accepting run

$$(q_0,w_0) \xrightarrow{a_1} ((j_1,R_1,S_1),w_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} ((j_n,R_n,S_n),w_n).$$

The first move is equivalent to having a move $((j_0,R_0,S_0),w_0) \xrightarrow{a_1} ((j_1,R_1,S_1),w_1)$ where $j_0 \in J_I$ and $S_0 = R_0 = \emptyset$. For $r=1,\dots,n$ the move $((j_{r-1},R_{r-1},S_{r-1}),w_{r-1}) \xrightarrow{a_r}$

$((j_r, R_r, S_r), w_r)$ is consistent with a transition of A_{fsuba} induced by some abstract transition $j_{r-1} \xrightarrow{E_r} j_r$ in $\mathsf{abs}(\delta)$ and the pair R_{r-1}, S_{r-1}. This yields the accepting run

$$(j_0, S_0, v_0) \xrightarrow{a_1} (j_1, S_1, v_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (j_n, S_n, v_n)$$

in $\widetilde{A}$ where $v_0 = a_1^m$, and $v_r \in \mathbb{A}^m$ for $r = 1, \dots, n$ is defined as follows:

- for $k \in R_r$ put $v_{r,k} = w_{r,k}$;
- for $k \notin S_r$ put $v_{r,k} = a_r$;
- for $k \in S_r \setminus R_r$, if $k = k$ in $(E_r)_{S_{r-1}}$ then take $v_{r,k} = v_{r-1,k}$, and if $\bullet = k$ in E_r then take $v_{r,k} = a_k$. Note that at least one of these cases must occur because $S_{r-1} \rightsquigarrow_{E_r} S_r$. Moreover, if both $k = k$ and $\bullet = k$ in E_r then also $k = \bullet$ in E_r and moreover $k \in S_{r-1}$, whence $k \in R_{r-1}$. Therefore $v_{r-1,k} = a_r$ as the move $((j_{r-1}, R_{r-1}, S_{r-1}), w_{r-1}) \xrightarrow{a_1} ((j_r, R_r, S_r), w_r)$ is consistent with a transition of A_{fsuba} induced by $j_{r-1} \xrightarrow{E_r} j_r$ and R_{r-1}, S_{r-1}, which is of the form $((j_{r-1}, R_{r-1}, S_{r-1}), k, \cdots)$ since $\bullet = k$ in E_r. Hence $v_{r,k}$ is properly defined.

It remains to show that $((j_{r-1}, S_{r-1}), v_{r-1}) \xrightarrow{a_1} ((j_r, S_r), v_r)$ is a valid transition of $\widetilde{A}$. By definition we have $S_{r-1} \rightsquigarrow_{E_r} S_r$, so we only need to show that the transition is consistent with $(E_r)_{S_{r-1}}$. Indeed:

- If $k = \bullet$ in $(E_r)_{S_{r-1}}$ (hence $k \in R_{r-1}$) then a transition of A_{fsuba} induced by $j_{r-1} \xrightarrow{E_r} j_r$ and R_{r-1}, S_{r-1} is of the form $((j_{r-1}, R_{r-1}, S_{r-1}), k, \cdots)$, and since by assumption the move $((j_{r-1}, R_{r-1}, S_{r-1}), w_{r-1}) \xrightarrow{a_r} ((j_r, R_r, S_r), w_r)$ is consistent with it, we have $v_{r-1,k} = w_{r-1,k} = a_r$.
- Now suppose that $k = k$ in $(E_r)_{S_{r-1}}$ but not $k = \bullet$ in $(E_r)_{S_{r-1}}$ (hence not $\bullet = k$ in E_r). We distinguish two subcases. If $k \in S_{r-1} \setminus R_{r-1}$, then $k \in S_r \setminus R_r$, so by definition of $v_{r,k}$ we get $v_{r-1,k} = v_{r,k}$. If $k \in R_{r-1}$, then the transition of A_{fsuba} induced by $j_{r-1} \xrightarrow{E_r} j_r$ and R_{r-1}, S_{r-1} is given by $((j_{r-1}, R_{r-1}, S_{r-1}), m+1, \overline{R_r}, (j_r, R_r, S_r))$ where $R_{r-1} \rightsquigarrow_{E_r} R_r$ and $S_{r-1} \rightsquigarrow_{E_r} S_r$. Since the move $((j_{r-1}, R_{r-1}, S_{r-1}), w_{r-1}) \xrightarrow{a_r} ((j_r, R_r, S_r), w_r)$ is consistent with that transition, we have $v_{r-1,k} = w_{r-1,k} = w_{r,k} = v_{r,k}$.

This concludes the proof. ◀

We conclude:

▶ **Theorem A.19.** *Positive register automata and FSUBA are equivalent.*

Proof. Every language accepted by some FSUBA as in Definition A.9 is clearly positive, and by Remark A.11 it is NOFA-recognizable, hence by Theorem 3.2 it is accepted by some positive register automaton. Conversely, every language accepted by positive register automaton is positive and NOFA-recognizable (using Theorem 3.2 again), and so Theorem 2.17 and Lemma A.18 show that is accepted by an FSUBA. ◀

Details for Remark 4.3

We prove that the language $L \subseteq \mathbb{A}^\star$ of all words where no data value occurs exactly once is not NOFA-recognizable. Suppose that L is recognized by a NOFA A, and let $m \in \mathbb{N}$ such that every state has a support of size m. Choose $m+1$ distinct names $a_1, \dots, a_{m+1}$. Then the word $w = a_1 \dots a_{m+1} a_1 \dots a_{m+1}$ lies in L, hence it admits an accepting run

$$q_0 \xrightarrow{a_1} \cdots \xrightarrow{a_{m+1}} q_{m+1} \xrightarrow{a_1} q_1' \xrightarrow{a_2} \cdots \xrightarrow{a_{m+1}} q_{m+1}'.$$

Since $\mathsf{supp}\, q_{m+1}$ has at most m elements, some a_j is fresh for q_{m+1}. Choose a name a_j' fresh for $a_1, \ldots, a_{m+1}$ and q_{m+1}, and let $\pi = (a_j\, a_j')$. Then by equivariance of transitions we have the following accepting run:

$$\pi \cdot q_0 \xrightarrow{a_1} \cdots \xrightarrow{a_j'} \pi \cdot q_j \xrightarrow{a_{j+1}} \cdots \xrightarrow{a_{m+1}} \pi \cdot q_{m+1} = q_{m+1} \xrightarrow{a_1} q_1' \xrightarrow{a_2} \cdots \xrightarrow{a_{m+1}} q_{m+1}'.$$

This means that the word $a_1 \cdots a_j' \cdots a_{m+1} a_1 \cdots a_j \cdots a_{m+1}$ is accepted by A although it does not lie in L, a contradiction.

Proof of Theorem 4.4

The "if" statement follows from Proposition 4.2. For the "only if" statement, suppose that $L \subseteq \mathbb{A}^\star$ is a positive NOFA-recognizable data language; then L is accepted by a NOFRA $\bar{A} = (\bar{Q}, \bar{\delta}, \bar{I}, \bar{F})$ as given by Construction 2.7. Our task is to construct an MSO$^{\sim,+}$-sentence φ such that, for all $w \in \mathbb{A}^\star$,

$$\bar{A} \text{ accepts } w \qquad \iff \qquad w \text{ satisfies } \varphi.$$

We shall make use of the characterization of accepted words from Proposition 2.15: when interpreted over $w = b_1 \ldots b_n$, the sentence φ will state existence of an abstract run satisfying condition (2.1). For this purpose, we introduce for every abstract transition $(j, E, j') \in \mathsf{abs}(\delta)$ a second-order variable $R_{(j,E,j')}$ with the intended interpretation

$$R_{(j,E,j')}(x) \quad \hat{=} \quad (j, E, j') \text{ is the } x\text{-th transition of an accepting abstract run of } \bar{A}.$$

The sentence φ is then given as follows:

$$\varphi \quad = \quad \vec{\exists}_{(j,E,j) \in \mathsf{abs}(\delta)} R_{(j,E,j')}.\, \varphi_{\mathrm{run}} \wedge \forall x.\, \exists \mathsf{Eq}_1^{(x)} \ldots \exists \mathsf{Eq}_m^{(x)}.\, \varphi_{\mathrm{aux}} \wedge \varphi_{\mathrm{eq}}.$$

Here $\vec{\exists}_{(j,E,j) \in \mathsf{abs}(\delta)} R_{(j,E,j')}$ denotes the concatenation of all $\exists R_{(j,E,j')}$ where $(j, E, j') \in \mathsf{abs}(\delta)$, and we make the convention that quantifiers have maximal scope. The subformula φ_{run} ensures that the variables $R_{(j,E,j')}$ define an accepting abstract run of $\bar{A}$; the subformula φ_{aux} ensures that the second-order variables $\mathsf{Eq}_k^{(x)}$ ($k = 1, \ldots, m$) are interpreted as the auxiliary predicates of Notation 2.14; the subformula φ_{eq} states the equality condition (2.1). In more detail, the three subformulas are given as follows:

Definition of φ_{run}:

$$\begin{aligned}
\varphi_{\mathrm{run}} \; = \; & \forall x.\, \bigvee_{(j,E,j') \in \mathsf{abs}(\delta)} R_{(j,E,j')}(x) \\
& \wedge \; \forall x.\, \bigwedge_{(j,E,j') \neq (\bar{j},\bar{E},\bar{j}') \in \mathsf{abs}(\delta)} \neg\big(R_{(j,E,j')}(x) \wedge R_{(\bar{j},\bar{E},\bar{j}')}(x) \big) \\
& \wedge \; \forall x. \forall y.\, \mathrm{succ}(x,y) \implies \bigvee_{(j,E,j'),(j',E',j'') \in \mathsf{abs}(\delta)} R_{(j,E,j')}(x) \wedge R_{(j',E',j'')}(y) \\
& \wedge \; \forall x.\, \mathrm{first}(x) \implies \bigvee_{\substack{(j,E,j') \in \mathsf{abs}(\delta) \\ j \in J_I}} R_{(j,E,j')}(x) \\
& \wedge \; \forall x.\, \mathrm{last}(x) \implies \bigvee_{\substack{(j,E,j') \in \mathsf{abs}(\delta) \\ j' \in J_F}} R_{(j,E,j')}(x)
\end{aligned}$$

As usual $\psi \Rightarrow \xi$ means $\neg\psi \vee \xi$, and the formulas $\mathrm{succ}(x,y)$, $\mathrm{first}(x)$ and $\mathrm{last}(x)$ define the successor relation and the first and last position, respectively:

$$\mathrm{succ}(x,y) = x < y \wedge \neg\exists z.\, x < z \wedge z < y, \qquad \mathrm{first}(x) = \neg\exists y.\, y < x, \qquad \mathrm{last}(x) = \neg\exists y. x < y.$$

The first two lines assert that every position x is associated with a unique abstract transition of A, and the remaining part that these transitions form an accepting abstract run.

Definition of φ_{aux}:

$$\begin{aligned}
\varphi_{\mathrm{aux}} \;=\; & \bigwedge_{k=1}^{m} \bigwedge_{\substack{(j,E,j')\in\mathsf{abs}(\delta)\\ \bullet=k\in E}} \big(R_{(j,E,j')}(x) \implies \mathsf{Eq}_k^{(x)}(x)\big)\\
& \wedge\; \forall y.\forall z.\,\mathrm{succ}(y,z) \implies \bigwedge_{k,\bar{k}=1}^{m} \bigwedge_{\substack{(j,E,j')\in\mathsf{abs}(\delta)\\ k=\bar{k}\in E}} \big(R_{(j,E,j')}(z) \wedge \mathsf{Eq}_k^{(x)}(y) \implies \mathsf{Eq}_{\bar{k}}^{(x)}(z)\big)\\
& \wedge\; \forall\overline{\mathsf{Eq}}_1^{(x)} \cdots \forall\overline{\mathsf{Eq}}_m^{(x)}.\, \Big[\bigwedge_{k=1}^{m} \bigwedge_{\substack{(j,E,j')\in\mathsf{abs}(\delta)\\ \bullet=k\in E}} \big(R_{(j,E,j')}(x) \implies \overline{\mathsf{Eq}}_k^{(x)}(x)\big)\\
& \wedge\; \forall y.\forall z.\,\mathrm{succ}(y,z) \implies \bigwedge_{k,\bar{k}=1}^{m} \bigwedge_{\substack{(j,E,j')\in\mathsf{abs}(\delta)\\ k=\bar{k}\in E}} \big(R_{(j,E,j')}(z) \wedge \overline{\mathsf{Eq}}_k^{(x)}(y) \implies \overline{\mathsf{Eq}}_{\bar{k}}^{(x)}(z)\big)\Big]\\
& \implies \bigwedge_{k=1}^{m} \forall y.\, \mathsf{Eq}_k^{(x)}(y) \implies \overline{\mathsf{Eq}}_k^{(x)}(y).
\end{aligned}$$

The first two lines state that the predicates $\mathsf{Eq}_k^{(x)}$ satisfy the two clauses of Notation 2.14, and the remaining part of the formula asserts that they are minimal with that property. This entails that $\mathsf{Eq}_k^{(x)}$ is precisely the inductively defined predicate of Notation 2.14.

Definition of φ_{eq}:

$$\varphi_{\mathrm{eq}} \;=\; \forall y.\forall z.\,\mathrm{succ}(y,z) \implies \bigwedge_{k=1}^{m} \bigwedge_{\substack{(j,E,j')\in\mathsf{abs}(\delta)\\ k=\bullet\in E}} \big(R_{(j,E,j')}(z) \wedge \mathsf{Eq}_k^{(x)}(y) \implies x \sim z\big).$$

Proof of Proposition 5.1

We start with two preliminary remarks on directed colimits,

▶ **Remark A.20.** Given a directed diagram $D\colon I \to \mathbf{Set}$, its colimit cocone $D_i \xrightarrow{c_i} \mathsf{colim}\, D$ $(i \in I)$ is characterized by two properties: (i) the morphisms c_i are jointly surjective (every element of $\mathsf{colim}\, D$ lies in the image of some c_i), and (ii) for every $i \in I$ and $x, y \in D_i$ such that $c_i(x) = c_i(y)$, there exists $j \geq i$ in I such that $D_{i,j}(x) = D_{i,j}(y)$, where $D_{i,j} = D(i \to j)$ is the connecting morphism induced by the unique arrow $i \to j$ in I. The same characterization applies to directed colimits in **Nom** and **RnNom**, since colimits in these two categories are formed at the level of underlying sets.

▶ **Remark A.21.** Recall that an object X of a category $\mathscr{C}$ is finitely presentable if the functor $\mathscr{C}(X,-)\colon \mathscr{C} \to \mathbf{Set}$ preserves directed colimits. By Remark A.20, this means precisely that for every directed diagram $D\colon I \to \mathscr{C}$ with colimit cocone $c_i\colon D_i \to \mathsf{colim}\, D$ $(i \in I)$,

1. every morphism $f\colon X \to \mathsf{colim}\, D$ factorizes as $f = c_i \circ g$ for some $i \in I$ and $g\colon D_i \to C$;
2. the factorization is essentially unique: given another factorization $f = c_i \cdot h$, one has $D_{i,j} \circ g = D_{i,j} \circ h$ for some $j \geq i$ in I.

Moreover, we need

▶ **Lemma A.22.** *Let X be a nominal renaming set generated by a single element $x \in X$, that is, $X = \{\, \rho \cdot x : \rho \in \mathsf{Fin}(\mathbb{A}) \,\}$. Then X is orbit-finite.*

Proof. Take a $\mathsf{Perm}(\mathbb{A})$-equivariant map $e\colon \mathbb{A}^{\#m} \to X$, where $m \in \mathbb{N}$, whose image contains x. Since the forgetful functor $U\colon \mathbf{RnNom} \to \mathbf{Nom}$ has a left adjojnt $F\colon \mathbf{Nom} \to \mathbf{RnNom}$ sending $\mathbb{A}^{\#m}$ to $\mathbb{A}^m$ [27, Thm. 3.7], the map e uniquely extends to a $\mathsf{Fin}(\mathbb{A})$-equivariant map $\widehat{e}\colon \mathbb{A}^m \to X$. Choose $w \in \mathbb{A}^m$ such that $x = \widehat{e}(y)$. The map $\widehat{e}$ is surjective because X is generated by x. Since $\mathbb{A}^m$ is orbit-finite, it follows that X is orbit-finite (using that e is $\mathsf{Perm}(\mathbb{A})$-equivariant, hence it sends orbits to orbits). ◀

Now we prove Proposition 5.1. By [29, Prop. 2.3.7] orbit-finite nominal sets are precisely the finitely presentable objects of $\mathbf{Nom}$. Hence we only need to prove the corresponding statement for nominal renaming sets, which can be reduced to the one for nominal sets.

Thus suppose that $X \in \mathbf{RnNom}$ is finitely presentable. Express X as the directed union of its orbit-finite $\mathsf{Fin}(\mathbb{A})$-equivariant subsets. To see that this is indeed a directed colimit, note that the cocone of inclusions is jointly surjective: every $x \in X$ is contained in the $\mathsf{Fin}(\mathbb{A})$-equivariant subset $\{\sigma \cdot x : \sigma \in \mathsf{Fin}(\mathbb{A})\}$, which is orbit-finite by Lemma A.22. Thus the identity map $\mathsf{id}_X\colon X \to X$ factorizes through some inclusion $X' \hookrightarrow X$ of an orbit-finite $\mathsf{Fin}(\mathbb{A})$-equivariant subset, which implies that $X \cong X'$ and thus X is orbit-finite.

Conversely, suppose that $X \in \mathbf{RnNom}$ is orbit-finite. Let $c_i\colon C_i \to C$ $(i \in I)$ be a directed colimit in $\mathbf{RnNom}$ (with connecting morphisms $c_{i,j}\colon C_i \to C_j$ for $i \leq j$), and let $f\colon X \to C_i$ be a $\mathsf{Fin}(\mathbb{A})$-equivariant map. Since X is finitely presentable as a nominal set, f factorizes in $\mathbf{Nom}$ as $f = c_i \cdot g$ for some $i \in I$. The map g is $\mathsf{Perm}(\mathbb{A})$-equivariant, but may not be $\mathsf{Fin}(\mathbb{A})$-equivariant. To fix this, choose elements $x_1, \ldots, x_n \in X$ representing the orbits of X, and names $a_1, \ldots, a_d \in \mathbb{A}$ such that $\mathsf{supp}\, x_r \subseteq \{a_1, \ldots, a_d\}$ for $r = 1, \ldots, n$. Moreover, let $\rho_1, \ldots, \rho_k \in \mathsf{Fin}(\mathbb{A})$ be all renamings that restrict to a map from $\{a_1, \ldots, a_d\}$ to $\{a_1, \ldots, a_d\}$ and fix all names in $\mathbb{A} \setminus \{a_1, \ldots, a_d\}$. For each x_r, ρ_s we have

$$c_i(g(\rho_s \cdot x_r)) = f(\rho_s \cdot x_r) = \rho_s \cdot f(x_r) = \rho_s \cdot c_i(g(x_r)) = c_i(\rho_s \cdot g(x_r))$$

using that c_i and f are $\mathsf{Fin}(\mathbb{A})$-equivariant. Thus $c_{i,j}(g(\rho_s \cdot x_r)) = c_{i,j}(\rho_s \cdot g(x_r))$ for some $j \geq i$, and so $c_{i,j}(g(\rho_s \cdot x_r)) = \rho_s \cdot c_{i,j}(g(x_r))$. Since I is directed, we may choose j independently of r and s. Thus, after replacing g with $c_{i,j} \circ g$ and i with j, we may assume that $g(\rho_s \cdot x_r) = \rho_s \cdot g(x_r)$ for all r, s.

We now show that this implies $g(\rho \cdot x) = \rho \cdot g(x)$ for all $x \in X$ and $\rho \in \mathsf{Fin}(\mathbb{A})$, hence g is $\mathsf{Fin}(\mathbb{A})$-equivariant. First, since the elements $x_1, \ldots, x_n$ represent the orbits of X, we have $x = \pi \cdot x_r$ for some r and $\pi \in \mathsf{Perm}(\mathbb{A})$. Choose $\tau \in \mathsf{Perm}(\mathbb{A})$ that restricts to an injective map from $\rho \circ \pi[\{a_1, \ldots, a_d\}]$ to $\{a_1, \ldots, a_d\}$. Then $\tau \cdot \rho \cdot \pi$ restricts to a map from

$\{a_1,\dots,a_d\}$ to $\{a_1,\dots,a_d\}$, hence is equal on $\{a_1,\dots,a_d\}$ to some ρ_s. It follows that

$$\begin{aligned} g(\rho\cdot x) &= g(\tau^{-1}\cdot\tau\cdot\rho\cdot\pi\cdot x_r)\\ &= \tau^{-1}\cdot g(\tau\cdot\rho\cdot\pi\cdot x_r)\\ &= \tau^{-1}\cdot g(\rho_s\cdot x_r)\\ &= \tau^{-1}\cdot\rho_s\cdot g(x_r)\\ &= \tau^{-1}\cdot\tau\cdot\rho\cdot\pi\cdot g(x_r)\\ &= \rho\cdot g(\pi\cdot x_r)\\ &= \rho\cdot g(x). \end{aligned}$$

The third and fifth step follows from $\mathsf{supp}\,x_r$, $\mathsf{supp}\,g(x_r)\subseteq\{a_1,\dots,a_d\}$, the fourth step by the choice of g, and the second and sixth step use $\mathsf{Perm}(\mathbb{A})$-equivariance of g.

This concludes the proof that f factorizes through c_i in **RnNom**. That the factorization is essentially unique is immediate from the corresponding property in **Nom**.

Proof of Proposition 5.2

An object X of a category $\mathscr{C}$ is *finitely generated* if the hom-functor $\mathscr{C}(X,-)$ preserves directed unions, i.e. colimits of directed diagrams $D\colon I\to\mathscr{C}$ for which each connecting morphism $D_{i,j}$ $(i\le j)$, is monic. (In locally finitely presentable categories [3], including all presheaf categories, the colimit injections of a directed union are also monic.) Clearly every finitely presentable object is finitely generated. We will first prove that super-finitary presheaves in $\mathbf{Set}^{\mathbb{I}}$ and $\mathbf{Set}^{\mathbb{F}}$ coincide with finitely generated presheaves, and subsequently prove that the latter coincide with finitely presentable presheaves.

▶ **Proposition A.23.** *For a presheaf $P\in\mathbf{Set}^{\mathscr{C}}$, $\mathscr{C}\in\{\mathbb{I},\mathbb{F}\}$, the following are equivalent:*
(a) *P is a finitely generated object of $\mathbf{Set}^{\mathscr{C}}$;*
(b) *P is super-finitary;*
(c) *P is a quotient of a finite coproduct of representables; that is, there exists a componentwise surjective natural transformation $\varphi\colon\coprod_{i\in I}\mathscr{C}(S_i,-)\twoheadrightarrow P$ with I finite and $S_i\subseteq_{\mathsf{f}}\mathbb{A}$.*

Proof. (a) $\Longrightarrow$ (b) Let $P\colon\mathscr{C}\to\mathbf{Set}$ be finitely generated. Since every presheaf in $\mathbf{Set}^{\mathscr{C}}$ is the directed union of its componentwise finite sub-presheaves, P itself is componentwise finite. For every $S\subseteq_{\mathsf{f}}\mathbb{A}$ we consider the sub-presheaf $P_S\subseteq P$ defined by

$$P_ST=\bigcup_{S'\subseteq S}\ \bigcup_{\rho\in\mathscr{C}(S',T)}P\rho[PS']\qquad\text{for}\qquad T\subseteq_{\mathsf{f}}\mathbb{A}.$$

Note that P_S is super-finitary with generating set S. Since $P_R,P_S\subseteq P_{R\cup S}$ for all $R,S\subseteq_{\mathsf{f}}\mathbb{A}$, the map $D\colon(\mathcal{P}_{\mathsf{f}}\mathbb{A},\subseteq)\to\mathbf{Set}^{\mathscr{C}}$, $S\mapsto P_S$, yields a directed diagram of monomorphisms with colimit cocone $P_S\xrightarrow{\subseteq}P$ $(S\subseteq_{\mathsf{f}}\mathbb{A})$. By hypothesis the presheaf P is finitely generated, so the identity map $\mathsf{id}\colon P\to P$ factorizes through some $P_S\xrightarrow{\subseteq}P$. This implies that the inclusion is surjective, whence $P=P_S$ is super-finitary.

(b) $\Longrightarrow$ (c) For every presheaf $P\in\mathbf{Set}^{\mathscr{C}}$ and $S\subseteq_{\mathsf{f}}\mathbb{A}$ we have the natural transformation φ whose component at $T\subseteq_{\mathsf{f}}\mathbb{A}$ is given by

$$\varphi_T\colon\coprod_{S'\subseteq S}PS'\times\mathscr{C}(S',T)\to PT,\qquad(x,\rho)\mapsto P\rho(x).$$

If P is super-finitary and S a generating set, then φ_T is surjective for each T.

(c) $\implies$ (a) This follows from two general facts: First, in every presheaf category the representable functors are finitely presentable [3, Example 1.2(7)], thus finitely generated. Second, in every locally finitely presentable category (hence in every presheaf category), finitely generated objects are closed under finite coproducts and strong quotients [3, Prop. 1.69]. ◀

▶ **Lemma A.24.** *Finitely generated objects in* $\mathbf{Set}^{\mathscr{C}}$*,* $\mathscr{C} \in \{\mathbb{I}, \mathbb{F}\}$*, are closed under finite products and sub-presheaves (hence under finite limits).*

Proof. *Closure under finite products.* We first prove that a product $\mathscr{C}(S_1,-) \times \mathscr{C}(S_2,-)$ of representables is finitely generated, i.e. super-finitary. Assuming w.l.o.g. that $S_1, S_2 \subseteq_{\mathsf{f}} \mathbb{A}$ are disjoint, we prove that $S_1 \cup S_2$ is a generating set. Thus let $T \subseteq_{\mathsf{f}} \mathbb{A}$ and $(\sigma_1,\sigma_2) \in \mathscr{C}(S_1,T) \times \mathscr{C}(S_2,T)$. Then there exists $S \subseteq S_1 \cup S_2$ and $\rho \in \mathscr{C}(C,T)$ that corestricts to a bijection $\pi\colon S \to \sigma_1[S_1] \cup \sigma_2[S_2]$. Let σ_k' denote the corestriction of σ_k to $\sigma_1[S_1] \cup \sigma_2[S_2]$. Then $\sigma_k = \rho \circ \pi^{-1} \circ \sigma_k'$, i.e. $(\sigma_1,\sigma_2) = (\mathscr{C}(S_1,\rho) \times \mathscr{C}(S_2,\rho))\,(\pi^{-1}\circ\sigma_1', \pi^{-1}\circ\sigma_2')$. This proves $\mathscr{C}(S_1,-) \times \mathscr{C}(S_2,-)$ to be super-finitary.

Now we turn to the general case. Suppose that $P_1, P_2\colon \mathscr{C} \to \mathbf{Set}$ are finitely generated presheaves. Then, by Proposition A.23, there exist componentwise surjective natural transformations

$$\varepsilon_k\colon \coprod_{i\in I_k} \mathscr{C}(S_i^{(k)},-) \twoheadrightarrow P_k$$

for $k = 1,2$, where I_k is finite and $S_i^{(k)} \subseteq_{\mathsf{f}} \mathbb{A}$. They induce the componentwise surjective natural transformation

$$\begin{tikzcd}
\coprod_{i\in I_1, j\in I_2} \mathscr{C}(S_i^{(1)},-) \times \mathscr{C}(S_j^{(2)},-) \arrow[r, phantom, "\cong"] & (\coprod_{i\in I_1} \mathscr{C}(S_i^{(1)},-)) \times (\coprod_{j\in I_2} \mathscr{C}(S_j^{(2)},-)) \arrow[d, two heads, "\varepsilon_1\times\varepsilon_2"] \\
 & P_1 \times P_2,
\end{tikzcd}$$

Since $\mathscr{C}(S_i^{(1)},-) \times \mathscr{C}(S_j^{(2)},-)$ is finitely generated as shown above, and finitely generated objects in any locally finitely presentable category are closed under finite coproducts and strong quotients, we conclude that $P_1 \times P_2$ is finitely generated.

Closure under sub-presheaves. We first prove that every sub-presheaf $Q \subseteq \mathscr{C}(C,-)$ of a representable presheaf is finitely generated, i.e. super-finitary. We may assume that $C \neq \emptyset$ and that Q is not the constant functor on $\emptyset$, for otherwise the claim is obvious. We consider the cases $\mathscr{C} = \mathbb{I}$ and $\mathscr{C} = \mathbb{F}$ separately:

- $\mathscr{C} = \mathbb{I}$: Choose $S \subseteq_{\mathsf{f}} \mathbb{A}$ of least cardinality such that $QS \neq \emptyset$; since $QS \subseteq \mathbb{I}(C,S)$ one has $|C| \leq |S|$. Note that $QS = \mathbb{I}(C,S)$: for any two maps $\sigma,\tau \in \mathbb{I}(C,S)$ one has $\tau = \pi \circ \sigma$ for some bijection $\pi\colon S \to S$, hence $\sigma \in QS$ implies $\tau \in QS$. Given $T \subseteq_{\mathsf{f}} \mathbb{A}$ and $\sigma \in QT$, one has $|C| \leq |S| \leq |T|$ and hence σ factorizes as $C \xrightarrow{\tau} S \xrightarrow{\rho} T$ for some $\rho,\tau \in \mathbb{I}$. Then $\tau \in QS = \mathbb{I}(C,S)$ and $\sigma = \mathscr{C}(C,\rho)(\tau) = Q\rho(\tau)$. Hence Q is super-finitary, as claimed.
- $\mathscr{C} = \mathbb{F}$: We prove that the set C generates Q. Given $T \subseteq_{\mathsf{f}} \mathbb{A}$ and $\sigma \in QT$, choose a map $\tau \in \mathbb{F}(T,C)$ that sends every element of $\sigma[C] \subseteq T$ to a preimage under σ, and is arbitrary otherwise. (Here we use that $C \neq \emptyset$.) Then $\tau \circ \sigma = Q\tau(\sigma) \in QC$ and $\sigma = \sigma \circ \tau \circ \sigma = \mathbb{F}(C,\sigma)(\tau\circ\sigma) = Q\sigma(\tau\circ\sigma)$, proving that Q is super-finitary.

Now we turn to the general case. Suppose that $Q \subseteq P$ is a sub-presheaf of a finitely generated presheaf $P \in \mathbf{Set}^{\mathscr{C}}$. By Proposition A.23 we have a componentwise surjective natural transformation $\varepsilon\colon \coprod_{i\in I} \mathscr{C}(S_i,-) \twoheadrightarrow P$ with I finite. Form the following pullback:

$$\begin{tikzcd} H \arrow[r, hook, "\subseteq"] \arrow[d, two heads, "\overline{\varepsilon}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \coprod_{i\in I} \mathscr{C}(S_i,-) \arrow[d, two heads, "\varepsilon"] \\ Q \arrow[r, hook, "\subseteq"'] & P \end{tikzcd}$$

By extensivity of the presheaf topos $\mathbf{Set}^{\mathscr{C}}$, the presheaf H is of the form $H = \coprod_{i\in I} H_i$ for sub-presheaves $H_i \subseteq \mathscr{C}(S_i,-)$. Each H_i is finitely generated as shown above, hence so is H because finitely generated objects are closed under finite coproducts. ◀

Finally, we have the following simple criterion for coincidence of finitely presentable and finitely generated objects, see [2, Lemma 3.32]:

▶ **Lemma A.25.** *Let $\mathscr{C}$ be a locally finitely presentable category where strong and regular epimorphisms coincide and finitely generated objects are closed under kernel pairs. Then the finitely presentable and finitely generated objects of $\mathscr{C}$ coincide.*

With these preparations, Proposition 5.2 now easily follows. By Proposition A.23 we only need to show coincidence of finitely presentable and finitely generated objects in $\mathbf{Set}^{\mathscr{C}}$, where $\mathscr{C} \in \{\mathbb{I}, \mathbb{F}\}$. To this end apply Lemma A.25: every presheaf category is locally finitely presentable, regular and strong epis coincide in every topos (and are just the epimorphisms), and closure of finitely generated objects under kernel pairs follows from Lemma A.24.

Proof of Proposition 5.3

For the proof the following result (see e.g. [1, Lem. 2.4]) will be helpful:

▶ **Lemma A.26.** *For every adjunction $F \dashv U \colon \mathscr{C} \to \mathscr{D}$ between categories with directed colimits, if U preserves directed colimits then F preserves finitely presentable objects.*

To prove that the left adjoints in (5.2) preserve finitely presentable objects, it suffices to show that their right adjoints preserve directed colimits (Lemma A.26).

- The forgetful functors $E^\star$ and U preserve all colimits because colimits in the four categories are formed at the level of underlying sets.
- To show that $I_\star$ preserves directed colimits, let $c_k \colon C_k \to C$ ($k \in K$) be a directed colimit cocone in **Nom**. Then the morphisms $(I_\star c_k)_S \colon (I_\star C_k)S \to (I_\star C)S$ are jointly surjective for every $S \subseteq_{\mathsf{f}} \mathbb{A}$ by [30, Lem. 5.14], and any two elements of $(I_\star C_k)S$ merged by $(I_\star c_k)_S$ are merged by $(I_\star c_{k,l})_S$ for some $k \le l$ because this holds in the directed colimit cocone (c_k) in **Nom**. Therefore the morphisms $I_\star c_k$ form a colimit cocone in $\mathbf{Set}^{\mathbb{I}}$.
- The proof that $J_\star$ preserves directed colimits is analogous.

It remains to show that the four right adjoints preserve finitely presentable objects.

- U clearly preserves finitely presentable objects. i.e. orbit-finite sets (Proposition 5.1).
- To show that $E^\star$ preserves finitely presentable objects, by Proposition 5.3 we need to show that for every super-finitary presheaf $P \colon \mathbb{F} \to \mathbf{Set}$ the presheaf $E^\star P = P \circ E \colon \mathbb{I} \to \mathbf{Set}$ is super-finitary. Clearly $P \circ E(T) = P(T)$ is finite for every $T \subseteq_{\mathsf{f}} \mathbb{A}$. Moreover, we claim that any set $S \subseteq_{\mathsf{f}} \mathbb{A}$ generating P also generates $P \circ E$. Indeed, suppose that $T \subseteq_{\mathsf{f}} \mathbb{A}$ and $x \in (P \circ E)T = PT$. Then $x = P\rho(x')$ for some $S' \subseteq S$ and $\rho \colon S' \to T$. The map ρ factorizes as $\rho = \rho_1 \circ \rho_0$ where $\rho_1 \colon S'' \to T$ is injective and $S'' \subseteq S'$. Thus, putting $x'' = P\rho_0(x')$, we have

$$x = P\rho(x') = P\rho_1(P\rho_0(x')) = P\rho_1(x'') = (P \circ E)\rho_1(x'') \in (P \circ E)\rho_1[(P \circ E)S''],$$

proving that $P \circ E$ is super-finitary.

- To show that the functor $I_\star$ preserves finitely presentable objects, by Proposition 5.1 and Proposition 5.3 we need to show that its sends orbit-finite nominal sets to super-finitary presheaves. Thus let $X \in \mathbf{Nom}$ be orbit-finite. Then $(I_\star X)T$ is finite for every $T \subseteq_{\mathsf{f}} \mathbb{A}$ because an orbit-finite set contains only finitely elements of any given finite support. We claim that $I_\star X$ is generated by any set $S \subseteq_{\mathsf{f}} \mathbb{A}$ of $n = \max_{x\in X} |\mathsf{supp}\, x|$ names. To see this, let $x \in (I_\star X)T$, that is, $x \in X$ with support T. Then x is supported by some subset $T' \subseteq T$ with at most n elements, that is, $x \in (I_\star X)T'$. Choose a bijection $\pi\colon S' \to T'$ where $S' \subseteq S$, which extends an injection $\rho\colon S' \to T$. It then follows that $x = (I_\star X)\rho(\pi^{-1} \cdot x) = x$, as required.
- An analogous argument shows that $J_\star$ preserves finitely presentable objects.

Proof of Proposition 6.9

▶ **Remark A.27.** Recall that by definition strong epimorphisms satisfy the *diagonal fill-in* property: For every commutative square as shown below where e is a strong epimorphism and m is a monomorphism, there exists a unique $d\colon B \to C$ making both triangles commute.

$$\begin{tikzcd}
A \arrow[r, two heads, "e"] \arrow[d, "f"'] & B \arrow[d, "g"] \arrow[dl, dashed, "d"'] \\
C \arrow[r, tail, "m"] & D
\end{tikzcd}$$

We will make use of the *pullback lemma*, see e.g. [10, Prop. 2.5.9]:

▶ **Lemma A.28.** *Given a commutative diagram as shown below in a category with pullbacks,*
1. *if (I) and (II) are pullbacks, then the outer rectangle is a pullback;*
2. *if (II) and the outer rectangle are pullbacks, then (I) is a pullback.*

$$\begin{tikzcd}
\bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(I)}"] & \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(II)}"] & \bullet \arrow[d] \\
\bullet \arrow[r] & \bullet \arrow[r] & \bullet
\end{tikzcd}$$

In the following let $A = (Q, \Sigma, \delta, I, F)$ and $A' = (Q', \Sigma, \delta', I', F')$ be nondeterministic $\mathscr{C}$-automata over the same alphabet Σ and suppose that $h\colon A' \to A$ is a morphism such that $h_{\mathsf{a}} = \mathsf{id}$. We prove the two parts of the proposition.

1. For every $n \geq 0$ we show that $L^{(n)}(A') \leq L^{(n)}(A)$ as subobjects of Σ^n. We only consider the case $n > 0$, as the argument for $n = 0$ is very similar. The universal property of the pullback AccRun_A yields a unique morphism e such that the upper part and the left-hand part of the diagram below commute; note that the outside and the other parts commute by definition.

$$\begin{tikzcd}
\mathsf{AccRun}_{A'} \arrow[r, tail, "\overline{m}^{(n)}_{\delta'}"] \arrow[d, dashed, "e"'] \arrow[ddd, bend right=60, "\overline{d}_{n,A'}"'] & I' \times (\Sigma \times Q')^{n-1} \times \Sigma \times F' \arrow[d, "h_{\mathsf{i}} \times (\mathsf{id} \times h_{\mathsf{s}})^{n-1} \times \mathsf{id} \times h_{\mathsf{f}}"] \arrow[ddd, bend left=60, "d_{n,A'}"] \\
\mathsf{AccRun}_A \arrow[r, tail, "\overline{m}^{(n)}_{\delta}"] \arrow[d, "\overline{d}_{n,A}"'] & I \times (\Sigma \times Q)^{n-1} \times \Sigma \times F \arrow[d, "d_{n,A}"] \\
\delta^n \arrow[r, "m^n_\delta"] & (Q \times \Sigma \times Q)^n \\
(\delta')^n \arrow[u, "h^n_\delta"] \arrow[r, "m^n_{\delta'}"] & (Q' \times \Sigma \times Q')^n \arrow[u, "(h_{\mathsf{s}} \times \mathsf{id} \times h_{\mathsf{s}})^n"']
\end{tikzcd} \tag{A.6}$$

Diagonal fill-in yields a unique morphism $i\colon L^{(n)}(A') \to L^{(n)}(A)$ making the upper part and the left-hand part of the diagram below commute; the outside and the other parts commute by definition. Hence the morphism i witnesses that $L^n(A') \le L^{(n)}(A)$.

$$\begin{tikzcd}
L^{(n)}(A') \arrow[dr, dashed, "i"] \arrow[ddd, tail, "m^{(n)}_{L(A)}"'] & & & & \mathsf{AccRun}_{A'} \arrow[llll, two heads, "e_{n,A'}"'] \arrow[dll, "e"'] \arrow[ddd, "\overline{m}^{(n)}_{\delta'}"] \\
& L^{(n)}(A) \arrow[d, tail, "m^{(n)}_{L(A)}"'] & \mathsf{AccRun}_A \arrow[l, two heads, "e_{n,A}"'] \arrow[d, "\overline{m}^{(n)}_{\delta}"] & & \\
& \Sigma^n \arrow[dl, "\mathsf{id}"'] & I \times (\Sigma \times Q)^{n-1} \times \Sigma \times F \arrow[l, "p_{n,A}"'] & & \\
\Sigma^n & & & & I' \times (\Sigma \times Q')^{n-1} \times \Sigma \times F' \arrow[llll, "p_{n,A'}"] \arrow[ull, "h_{\mathsf{i}} \times (\mathsf{id} \times h_{\mathsf{s}})^{n-1} \times \mathsf{id} \times h_{\mathsf{f}}"]
\end{tikzcd} \tag{A.7}$$

2. Now suppose that h_{s} is a strong epimorphism in $\mathscr{C}$ and that the three squares (6.1) are pullbacks. Our task is to show that $L^{(n)}(A') = L^{(n)}(A)$ for all $n \ge 0$; as above we only consider the case $n > 0$. We first observe that the upper rectangle of (A.6) is a pullback. To see this, note that the outside is a pullback by definition, and that the lower rectangle is a pullback by our assumption on h and the fact that in every category pullbacks commute with products. Thus Lemma A.28.1 shows that the composite of the upper and the central rectangle forms a pullback, as it is is equal to the composite of the outside and the lower rectangle. Moreover the central rectangle is a pullback by definition, and so by Lemma A.28.2, the upper rectangle is a pullback as well.
Since strong epimorphisms in $\mathscr{C}$ are stable under pullbacks and products (Assumptions 6.1), the morphism $h_{\mathsf{i}} \times (\mathsf{id} \times h_{\mathsf{s}})^{n-1} \times \mathsf{id} \times h_{\mathsf{f}}$ appearing in the upper rectangle is a strong epimorphism. Using stability under pullbacks again, we see that e is a strong epimorphism. Therefore, by the uniqueness of image factorizations, the diagonal fill-in i in (A.7) is an isomorphism, proving that $L^{(n)}(A') = L^{(n)}(A)$ as subobjects of Σ^n.

Proof of Proposition 6.10

1. Given a functor $G\colon \mathscr{C} \to \mathscr{D}$ we define the lifted functor as follows:

$$\begin{array}{rccc}
\overline{G}\colon & \mathbf{NAut}(\mathscr{C}) & \longrightarrow & \mathbf{NAut}(\mathscr{D}) \\
\left\{\begin{array}{l} \\ \\ \end{array}\right. & \begin{array}{c}(Q,\, \Sigma,\, \delta,\, I,\, F) \\ (h_{\mathsf{s}},\, h_{\mathsf{a}})\end{array} & \begin{array}{c}\longmapsto \\ \longmapsto\end{array} & \begin{array}{l}\big(GQ,\, G\Sigma,\, \overline{G\delta},\, \overline{GI},\, \overline{GF}\big) \\ (Gh_{\mathsf{s}},\, Gh_{\mathsf{a}})\end{array}
\end{array}$$

Herein, the objects $\overline{G\delta}$, $\overline{GI}$, and $\overline{GF}$ are given by the image factorizations shown below, with can denoting the canonical morphism induced by the product projections:

$$\begin{tikzcd}
G\delta \arrow[rr, two heads, "e_{\overline{G\delta}}"] \arrow[d, "Gm_\delta"'] & & \overline{G\delta} \arrow[d, tail, "m_{\overline{G\delta}}"] \\
G(Q \times \Sigma \times Q) \arrow[rr, "\mathsf{can}"'] & & GQ \times G\Sigma \times GQ
\end{tikzcd}
\qquad
\begin{tikzcd}
GI \arrow[r, two heads, "e_{\overline{GI}}"] \arrow[d, "Gm_I"'] & \overline{GI} \arrow[dl, tail, "m_{\overline{GI}}"] \\
GQ &
\end{tikzcd}
\qquad
\begin{tikzcd}
GF \arrow[r, two heads, "e_{\overline{GF}}"] \arrow[d, "Gm_F"'] & \overline{GF} \arrow[dl, tail, "m_{\overline{GF}}"] \\
GQ &
\end{tikzcd}$$

We only need to prove that $(Gh_{\mathsf{s}},\, Gh_{\mathsf{a}})$ is an $\mathbf{NAut}(\mathscr{D})$-morphism for every $\mathbf{NAut}(\mathscr{C})$-morphism $h = (h_{\mathsf{s}},\, h_{\mathsf{a}}) : A' \to A$ between automata $A' = (Q',\, \Sigma',\, \delta',\, I',\, F')$ and $A = (Q,\, \Sigma,\, \delta,\, I,\, F)$. Indeed, via diagonal fill-in we obtain the dashed morphisms making the diagrams below commute; note that in all three diagrams the outside commutes because h is an $\mathbf{NAut}(\mathscr{C})$-morphism, and the parts not involving the dashed morphisms commute either by definition or by naturality of can. The central part of the first diagram and the lower parts of the other two diagrams show that $(Gh_\mathsf{s}, Gh_\mathsf{a})$ is an $\mathbf{NAut}(\mathscr{C})$-morphism.

$$\begin{tikzcd}
G\delta' \arrow[rrr, "Gh_\mathsf{t}"] \arrow[dddd, "Gm_{\delta'}"'] \arrow[dr, two heads, "e_{\overline{G\delta'}}"'] & & & G\delta \arrow[dddd, "Gm_\delta"] \arrow[dl, two heads, "e_{\overline{G\delta}}"] \\
& \overline{G\delta'} \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_{\overline{G\delta'}}"'] & \overline{G\delta} \arrow[d, tail, "m_{\overline{G\delta}}"] & \\
& GQ' \times G\Sigma' \times GQ' \arrow[r, "Gh_\mathsf{s} \times Gh_\mathsf{a} \times Gh_\mathsf{s}"'] & GQ \times G\Sigma \times GQ & \\
& & & \\
G(Q' \times \Sigma' \times Q') \arrow[rrr, "G(h_\mathsf{s} \times h_\mathsf{a} \times h_\mathsf{s})"'] \arrow[uur, "\mathsf{can}"] & & & G(Q \times \Sigma \times Q) \arrow[uul, "\mathsf{can}"']
\end{tikzcd}$$

$$\begin{tikzcd}
GI' \arrow[rrr, "Gh_\mathsf{i}"] \arrow[dd, "Gm_{I'}"'] \arrow[dr, two heads, "e_{\overline{GI'}}"'] & & & GI \arrow[dd, "Gm_I"] \arrow[dl, two heads, "e_{\overline{GI}}"] \\
& \overline{GI'} \arrow[r, dashed, "\exists"] \arrow[dl, tail, "m_{\overline{GI'}}"'] & \overline{GI} \arrow[dr, tail, "m_{\overline{GI}}"] & \\
GQ' \arrow[rrr, "Gh_\mathsf{s}"] & & & GQ
\end{tikzcd}
\qquad
\begin{tikzcd}
GF' \arrow[rrr, "Gh_\mathsf{f}"] \arrow[dd, "Gm_{F'}"'] \arrow[dr, two heads, "e_{\overline{GF'}}"'] & & & GF \arrow[dd, "Gm_F"] \arrow[dl, two heads, "e_{\overline{GF}}"] \\
& \overline{GF'} \arrow[r, dashed, "\exists"] \arrow[dl, tail, "m_{\overline{GF'}}"'] & \overline{GF} \arrow[dr, tail, "m_{\overline{GF}}"] & \\
GQ' \arrow[rrr, "Gh_\mathsf{s}"] & & & GQ
\end{tikzcd}$$

2. Let $L \dashv R \colon \mathscr{C} \to \mathscr{D}$ be an adjunction with unit $\eta \colon \mathsf{id}_\mathscr{D} \to RL$ and counit $\varepsilon \colon LR \to \mathsf{id}_\mathscr{C}$. We only need to establish the following two statements:

a. for every $\mathscr{D}$-automaton $A = (Q, \Sigma, \delta, I, F)$ the pair $\overline{\eta}_A = (\eta_Q, \eta_\Sigma) \colon A \to \overline{R}\,\overline{L}A$ is an $\mathbf{NAut}(\mathscr{D})$-morphism;

b. for every $\mathscr{C}$-automaton $A = (Q, \Sigma, \delta, I, F)$ the pair $\overline{\varepsilon}_A = (\varepsilon_Q, \varepsilon_\Sigma) \colon \overline{L}\,\overline{R}A \to A$ is an $\mathbf{NAut}(\mathscr{C})$-morphism.

Then $\overline{L} \dashv \overline{R}$ is an adjunction with unit $\overline{\eta}$ and counit $\overline{\varepsilon}$. Indeed, naturality of $\overline{\eta}$ and $\overline{\varepsilon}$ and the triangle laws are immediate from the corresponding properties of η and ε.

The proof of the first statement is given by the commutative diagrams below, where we write L_I, L_F, L_δ for $\overline{LI}$, $\overline{LF}$, $\overline{L\delta}$ and the dashed morphisms are just given by composition. In all three diagrams the outside commutes by naturality of η, and the parts not involving the dashed morphisms commute by definition.

$$\begin{tikzcd}
\delta \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_\delta"'] \arrow[rrr, bend left, "\eta_\delta"] & \overline{RL_\delta} \arrow[d, tail, "m_{\overline{RL_\delta}}"] & RL_\delta \arrow[l, two heads, "e_{\overline{RL_\delta}}"'] \arrow[d, "Rm_{L_\delta}"] & RL\delta \arrow[l, "Re_{L_\delta}"'] \arrow[d, "RLm_\delta"] \\
Q \times \Sigma \times Q \arrow[r, "\eta_Q \times \eta_\Sigma \times \eta_Q"'] \arrow[rrr, bend right, "\eta_{Q \times \Sigma \times Q}"'] & RLQ \times RL\Sigma \times RLQ & R(LQ \times L\Sigma \times LQ) \arrow[l, "\mathsf{can}"] & RL(Q \times \Sigma \times Q) \arrow[l, "R\mathsf{can}"]
\end{tikzcd}$$

$$\begin{tikzcd}
& & RLI \arrow[d, "Re_{L_I}"] \arrow[ddl, bend left, "RLm_I"] \\
I \arrow[urr, bend left, "\eta_I"] \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_I"'] & \overline{RL_I} \arrow[d, tail, "m_{\overline{RL_I}}"'] & RL_I \arrow[l, two heads, "e_{\overline{RL_I}}"'] \arrow[dl, "Rm_{L_I}"] \\
Q \arrow[r, "\eta_Q"'] & RLQ &
\end{tikzcd}
\qquad
\begin{tikzcd}
& & RLF \arrow[d, "Re_{L_F}"] \arrow[ddl, bend left, "RLm_F"] \\
F \arrow[urr, bend left, "\eta_F"] \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_F"'] & \overline{RL_F} \arrow[d, tail, "m_{\overline{RL_F}}"'] & RL_F \arrow[l, two heads, "e_{\overline{RL_F}}"'] \arrow[dl, "Rm_{L_F}"] \\
Q \arrow[r, "\eta_Q"'] & RLQ &
\end{tikzcd}$$

Similarly, the second statement is proven by the three diagrams below, where we write R_I, R_F, R_δ for $\overline{RI}, \overline{RF}, \overline{R\delta}$ and the dashed morphisms are given by diagonal fill-in. Here we use the fact that L preserves strong epimorphisms, being a left adjoint, and that strong epimorphisms are closed under composition.

$$\begin{tikzcd}
LR\delta \arrow[r, two heads, "Le_{R_\delta}"] \arrow[d, "LRm_\delta"'] \arrow[rrr, bend left, "\varepsilon_\delta"] & LR_\delta \arrow[r, two heads, "e_{\overline{LR_\delta}}"] \arrow[d, "Lm_{R_\delta}"] & \overline{LR_\delta} \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_{\overline{LR_\delta}}"] & \delta \arrow[d, tail, "m_\delta"] \\
LR(Q \times \Sigma \times Q) \arrow[r, "L\mathsf{can}"'] \arrow[rrr, bend right, "\varepsilon_{Q \times \Sigma \times Q}"'] & L(RQ \times R\Sigma \times RQ) \arrow[r, "\mathsf{can}"'] & LRQ \times LR\Sigma \times LRQ \arrow[r, "\varepsilon_Q \times \varepsilon_\Sigma \times \varepsilon_Q"'] & Q \times \Sigma \times Q
\end{tikzcd}$$

$$\begin{tikzcd}
LRI \arrow[d, two heads, "Le_{R_I}"] \arrow[rr, bend left, "\varepsilon_I"] \arrow[ddr, bend right, "LRm_I"'] & & \\
LR_I \arrow[r, two heads, "e_{\overline{LR_I}}"] \arrow[dr, "Lm_{R_I}"'] & \overline{LR_I} \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_{\overline{LR_I}}"] & I \arrow[d, tail, "m_I"] \\
 & LRQ \arrow[r, "\varepsilon_Q"'] & Q
\end{tikzcd}
\qquad
\begin{tikzcd}
LRF \arrow[d, two heads, "Le_{R_F}"] \arrow[rr, bend left, "\varepsilon_F"] \arrow[ddr, bend right, "LRm_F"'] & & \\
LR_F \arrow[r, two heads, "e_{\overline{LR_F}}"] \arrow[dr, "Lm_{R_F}"'] & \overline{LR_F} \arrow[r, dashed, "\exists"] \arrow[d, tail, "m_{\overline{LR_F}}"] & F \arrow[d, tail, "m_F"] \\
 & LRQ \arrow[r, "\varepsilon_Q"'] & Q
\end{tikzcd}$$

This concludes the proof.

Proof of Proposition 7.3

Put $\mathsf{Lan} = \mathsf{Lan}_E$ and consider the morphism

$$\varphi\colon \mathsf{Lan}(L) \xrightarrow{\mathsf{Lan}(\iota)} \mathsf{Lan}(V_{\mathbb{I}}^*) \cong \coprod_k \mathsf{Lan}(V_{\mathbb{I}}^k) \xrightarrow{\coprod_k \mathsf{can}_k} \coprod_k \mathsf{Lan}(V_{\mathbb{I}})^k = \coprod_k V_{\mathbb{F}}^k = V_{\mathbb{F}}^*$$

in $\mathbf{Set}^{\mathbb{F}}$, where $\iota\colon L \hookrightarrow V_{\mathbb{I}}^\star$ is the inclusion and can_k is the canonical morphism induced by the product projections, and form its image factorization

$$\varphi \;=\; \big(\mathsf{Lan}(L) \xrightarrow{\mathsf{coim}\,\varphi} \overline{L} \xhookrightarrow{\mathsf{im}\,\varphi} V_{\mathbb{F}}^\star\big).$$

We prove that $\overline{L}$ is a positive closure of L. First, the diagram below demonstrates that $L \subseteq \overline{L}E$, witnessed by the morphism $(\mathsf{coim}\,\varphi)E \circ \eta_L$, where η is the unit of the adjunction $\mathsf{Lan} \dashv E^\star\colon \mathbf{Set}^{\mathbb{F}} \to \mathbf{Set}^{\mathbb{I}}$. Indeed, all parts commute either by definition or by naturality.

$$\begin{tikzcd}
L \arrow[r, "\eta_L"] \arrow[dddd, "\iota"'] & \mathsf{Lan}(L)E \arrow[rr, "(\mathsf{coim}\,\varphi)E"] \arrow[d, "\mathsf{Lan}(\iota)E"] \arrow[ddddrr, bend left, "\varphi E"] & & \overline{L}E \arrow[dddd, "(\mathsf{im}\,\varphi)E"] \\
 & \mathsf{Lan}(V_{\mathbb{I}}^\star)E \arrow[d, "\cong"] & & \\
 & (\coprod_k \mathsf{Lan}(V_{\mathbb{I}}^k))E \arrow[dr, "(\coprod_k \mathsf{can}_k)E"] & & \\
 & \coprod_k V_{\mathbb{I}}^k \arrow[u, "\coprod_k \eta_{V_{\mathbb{I}}^k}"] \arrow[r, "\coprod_k \eta_{V_{\mathbb{I}}}^k"] & (\coprod_k \mathsf{Lan}(V_{\mathbb{I}})^k)E \arrow[d, "\cong"] & \\
V_{\mathbb{I}}^\star \arrow[ru, "\cong"] \arrow[uuur, bend left, "\eta_{V_{\mathbb{I}}^\star}"] \arrow[rr, equal] & & V_{\mathbb{F}}^\star E \arrow[r, equal] & V_{\mathbb{F}}^\star E
\end{tikzcd}$$

To show that $\overline{L}$ is minimal with that property, let $K \subseteq V_{\mathbb{F}}^{\star}$ such that $L \subseteq KE$; denote the inclusions by $\psi\colon K \hookrightarrow V_{\mathbb{F}}^{\star}$ and $\xi\colon L \hookrightarrow KE$. By the universal property of $\mathsf{Lan}(L)$, there exists a unique $\overline{\xi}\colon \mathsf{Lan}(L) \to K$ such that $\xi = \overline{\xi}E \circ \eta$. Now consider the first diagram below; its left-hand part commutes by definition, and the outside commutes because it does so when restricted to E and precomposed with the universal map η, see the second diagram (note that $\iota = \varphi E \circ \eta$ by the diagram above). Hence we obtain the dashed morphism via diagonal fill-in, witnessing that $\overline{L} \subseteq K$.

$$\begin{tikzcd}
\mathsf{Lan}(L) \arrow[rr, "\overline{\xi}"] \arrow[d, "\mathsf{Lan}(\iota)"'] \arrow[rdd, two heads, "\mathsf{coim}\,\varphi"] & & K \arrow[dddd, tail, "\psi"] \\
\mathsf{Lan}(V_{\mathbb{I}}^{*}) \arrow[d, phantom, sloped, "\cong"] & & \\
\coprod_k \mathsf{Lan}(V_{\mathbb{I}}^{k}) \arrow[d, "\coprod_k \mathsf{can}_k"'] & \overline{L} \arrow[ruu, dashed, "\exists"] \arrow[dd, tail, "\mathsf{im}\,\varphi"] & \\
\coprod_k \mathsf{Lan}(V_{\mathbb{I}})^{k} \arrow[d, phantom, sloped, "\cong"] & & \\
V_{\mathbb{F}}^{*} \arrow[r, equal] & V_{\mathbb{F}}^{*} & \arrow[l]
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathsf{Lan}(L)E \arrow[ld, "\overline{\xi}E"'] \arrow[dd, bend left=60, "\varphi E"] \\
KE \arrow[rd, hook, "\psi E"'] & L \arrow[l, hook', "\xi"'] \arrow[u, hook, "\eta"'] \arrow[d, hook, "\iota"] \\
& V_{\mathbb{F}}^{*}E
\end{tikzcd}$$

Proof of Proposition 7.5

▶ **Remark A.29.** The left Kan extension $\mathsf{Lan}_E P\colon \mathbb{F} \to \mathbf{Set}$ of a presheaf $P\colon \mathbb{I} \to \mathbf{Set}$ along $E\colon \mathbb{I} \hookrightarrow \mathbb{F}$ is computed as follows, see e.g. [24, Thm. X.3.1]:

- For $S \subseteq_{\mathsf{f}} \mathbb{A}$ the set $\mathsf{Lan}_E P(S)$ is the colimit of the diagram

  $$D_S\colon E{\downarrow}S \to \mathbf{Set}, \quad (\rho\colon ET \to S) \mapsto PT.$$

  Here $E{\downarrow}S$ is the comma category whose objects are maps $\rho\colon ET \to S$ in $\mathbb{F}$ where $T \subseteq_{\mathsf{f}} \mathbb{A}$ and whose morphisms from ρ to $\rho'\colon ET' \to S$ are maps $\tau\colon T \to T'$ in $\mathbb{I}$ such that $\rho = \rho' \circ E\tau$. For an even more explicit description of $\mathsf{Lan}_E P(S)$ consider the set of all pairs (x, ρ) where $\rho\colon ET \to S$ for some $T \subseteq_{\mathsf{f}} \mathbb{A}$ and $x \in PT$, and for any two such pairs put $(x, \rho) \sim (x', \rho')$ iff there exists a ziz-zag

  $$T = T_0 \xrightarrow{\tau_1} T_1 \xleftarrow{\tau_2} T_2 \to \cdots \xleftarrow{\tau_{2n}} T_{2n} = T'$$

  in $\mathbb{I}$ and elements $x_i \in PT_i$ $(i = 0, \ldots, n)$ such that $x_0 = x$, $x_{2n} = x'$, $P\tau_i(x_{i-1}) = x_i$ for i odd, and $P\tau_i(x_i) = x_{i-1}$ for $i > 0$ even. Then $\sim$ is an equivalence relation, and $\mathsf{Lan}_E P(S)$ is the set of equivalence classes $[x, \rho]$ of $\sim$. The colimit injection $c_\rho\colon PT \to \mathsf{Lan}_E P(S)$ associated to $\rho \in E{\downarrow}S$ maps $x \in PT$ to $[x, \rho]$.
- For $\sigma\colon S \to S'$ in $\mathbb{F}$, the map $\mathsf{Lan}_E P(\sigma)\colon \mathsf{Lan}_E P(S) \to \mathsf{Lan}_E P(S')$ sends $[x, \rho]$ to $[x, \sigma \circ \rho]$.
- For a morphism $f\colon P \to P'$ in $\mathbf{Set}^{\mathbb{I}}$, the component of $\mathsf{Lan}_E f\colon \mathsf{Lan}_E P \to \mathsf{Lan}_E P'$ at $S \subseteq_{\mathsf{f}} \mathbb{A}$ is given by $[x, \rho] \mapsto [f_T(x), \rho]$ where $\rho\colon ET \to S$ and $x \in PT$.

Let $A = (Q, V_{\mathbb{I}}, \delta, I, F)$ be a nondeterministic $\mathbf{Set}^{\mathbb{I}}$-automaton with a strong presheaf Q of states. We put $\mathsf{Lan} = \mathsf{Lan}_E$ and $\bar{A} = \overline{\mathsf{Lan}}\,A$, that is,

$$\bar{A} = (\mathsf{Lan}\,Q, V_{\mathbb{F}}, \bar{\delta}, \bar{I}, \bar{F}),$$

where $\bar{\delta} = \overline{\mathsf{Lan}\,\delta}$, $\bar{I} = \overline{\mathsf{Lan}\,I}$ and $\bar{F} = \overline{\mathsf{Lan}\,F}$ are obtained via the image factorizations of Proposition 6.10. Our task is to show that $\bar{A}$ accepts the language $\overline{L(A)}$, that is, $L^{(n)}(\bar{A}) = \overline{L(A)}^{(n)}$ for all $n \geq 0$. We shall only treat the case $n > 0$; the argument for $n = 0$ is similar.

Step 1. The universal property of the pullback $\mathsf{AccRun}^{(n)}_{\overline{A}}$ yields a unique morphism ε making the diagram below commute:

$$\begin{tikzcd}
\mathsf{Lan}(\mathsf{AccRun}^{(n)}_{A}) \arrow[rrrr, "\mathsf{Lan}\,\overline{d}_{n,A}"] \arrow[rrd, dashed, "\varepsilon"] \arrow[dddd, "\mathsf{Lan}\,\overline{m}^{(n)}_{\delta}"'] & & & & \mathsf{Lan}\,\delta^n \arrow[ld, "e^n_{\bar\delta}"'] \arrow[dddd, "\mathsf{Lan}\,m^n_{\delta}"] \\
& & \mathsf{AccRun}^{(n)}_{\overline{A}} \arrow[r, "\overline{d}_{n,\overline{A}}"] \arrow[d, "\overline{m}^{(n)}_{\bar\delta}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \bar\delta^n \arrow[d, "m^n_{\bar\delta}"] & \\
& & \bar I \times (V_{\mathbb{F}} \times \mathsf{Lan}\,Q)^{n-1} \times V_{\mathbb{F}} \times \bar F \arrow[r, "d_{n,\overline{A}}"] & (\mathsf{Lan}\,Q \times V_{\mathbb{F}} \times \mathsf{Lan}\,Q)^n & \\
& & \mathsf{Lan}\,I \times (\mathsf{Lan}\,V_{\mathbb{I}} \times \mathsf{Lan}\,Q)^{n-1} \times \mathsf{Lan}\,V_{\mathbb{I}} \times \mathsf{Lan}\,F \arrow[u, "e_{\bar I} \times (\mathsf{id} \times \mathsf{id})^{n-1} \times \mathsf{id} \times e_{\overline{F}}"] \arrow[ru, "\mathsf{Lan}\,m_I \times (\mathsf{id} \times \Delta)^{n-1} \times \mathsf{id} \times \mathsf{Lan}\,m_F"'] & & \\
\mathsf{Lan}(I \times (V_{\mathbb{I}} \times Q)^{n-1} \times V_{\mathbb{I}} \times F) \arrow[rrrr, "\mathsf{Lan}\,d_{n,A}"] \arrow[rru, "\mathsf{can}"] & & & & \mathsf{Lan}(Q \times V_{\mathbb{I}} \times Q)^n \arrow[luu, bend right, "\mathsf{can}^n"']
\end{tikzcd}$$

Step 2. We will show below that ε is a (strong) epimorphism. With this we can conclude the proof as follows. Consider the diagram below, where p is the projection. The part marked $(\star)$ commutes because the outside and all other parts commute either by definition or by naturality. By definition, $\overline{L(A)}^{(n)}$ is the image of the morphism $\mathsf{can} \circ \mathsf{Lan}\, m^{(n)}_{L(A)}$ appearing on the left-hand side of the diagram. Since $\mathsf{Lan}\, e_{n,A}$ is an epimorphism (using that the left adjoint Lan preserves epimorphisms), the morphism $\mathsf{can} \circ \mathsf{Lan}\, m^{(n)}_{L(A)} \circ \mathsf{Lan}\, e_{n,A}$ has the same image, and by commutativity of $(\star)$ and because ε is an epimorphism, this image is precisely $m^{(n)}_{L(\bar A)} \colon L^{(n)}(\bar A) \rightarrowtail V^n_{\mathbb{F}}$. Hence $L^{(n)}(\bar A) \cong \overline{L(A)}^{(n)}$ as subobjects of $V^n_{\mathbb{F}}$, as required.

$$\begin{tikzcd}
\mathsf{Lan}(L^{(n)}(A)) \arrow[ddddd, bend right=60, "\mathsf{Lan}\,m^{(n)}_{L(A)}"'] & \mathsf{Lan}(\mathsf{AccRun}^{(n)}_{A}) \arrow[l, two heads, "\mathsf{Lan}\,e_{n,A}"'] \arrow[d, two heads, "\varepsilon"] \arrow[ddddd, bend left=60, "\mathsf{Lan}\,\overline{m}^{(n)}_{A}"] \\
(\star) & \\
L^{(n)}(\bar A) \arrow[d, tail, "m^{(n)}_{L(\bar A)}"'] & \mathsf{AccRun}^{(n)}_{\overline{A}} \arrow[l, two heads, "e_{n,\overline{A}}"'] \arrow[d, "\overline{m}^{(n)}_{\overline{A}}"] \\
V^n_{\mathbb{F}} \arrow[d, equal] & \bar I \times (V_{\mathbb{F}} \times \mathsf{Lan}\,Q)^{n-1} \times V_{\mathbb{F}} \times \bar F \arrow[l, "p_{n,\overline{A}}"] \\
(\mathsf{Lan}\,V_{\mathbb{I}})^n & \mathsf{Lan}\,I \times (\mathsf{Lan}\,V_{\mathbb{I}} \times \mathsf{Lan}\,Q)^{(n-1)} \times \mathsf{Lan}\,V_{\mathbb{I}} \times \mathsf{Lan}\,F \arrow[l, "p"'] \arrow[u, "e_{\bar I} \times (\mathsf{id} \times \mathsf{id})^{n-1} \times \mathsf{id} \times e_{\overline{F}}"'] \\
\mathsf{Lan}(V^n_{\mathbb{I}}) \arrow[u, "\mathsf{can}"] & \mathsf{Lan}(I \times V_{\mathbb{I}} \times Q \times V_{\mathbb{I}} \times F) \arrow[l, "\mathsf{Lan}\,p_{n,A}"'] \arrow[u, "\mathsf{can}"']
\end{tikzcd}$$

Step 3. It remains to prove the above claim that ε is an epimorphism, i.e. each component

$$\varepsilon_S \colon \mathsf{Lan}(\mathsf{AccRun}^{(n)}_{A})S \to \mathsf{AccRun}^{(n)}_{\overline{A}}S \qquad (S \subseteq_{\mathsf{f}} \mathbb{A})$$

is surjective. We first give an explicit description of ε_S using Remark A.29. For each $T \subseteq_{\mathsf{f}} \mathbb{A}$ the set $\mathsf{AccRun}^{(n)}_{A}T$ consists of all T-supported runs of A, that is, all triples $(p_0, a_1, p_1, \dots, a_n, p_n)$ where $p_0 \in IT$, $p_n \in FT$, and $(p_{r-1}, a_r, p_r) \in \delta T$ for $r = 1, \dots, n$. Then the map ε_S is given by

$$[(q_0, a_1, q_1, \dots, a_n, q_n), \rho] \quad \mapsto \quad ([q_0, \rho], \rho(a_1), [q_1, \rho], \dots, \rho(a_n), [q_n, \rho]),$$

for all $\rho \colon ET \to S$ in $\mathbb{F}$ and $(q_0, a_1, q_1, \dots, a_n, q_n) \in \mathsf{AccRun}^{(n)}_{A}T$.

To prove ε_S surjective, regard the pullback $\mathsf{AccRun}^{(n)}_A S$ as a subset of $((\mathsf{Lan}\,Q)S \times V_{\mathbb{F}}S \times (\mathsf{Lan}\,Q)S)^n$, see Definition 6.6. Then every element of $\mathsf{AccRun}^{(n)}_A S$ is a tuple of the form

$$([q_0,\rho_1], \rho_1(a_1), [q_1,\rho_1], [q'_1,\rho_2], \rho_2(a_2), [q_2,\rho_2], \dots [q'_{n-1},\rho_n], \rho_n(a_n), [q_n,\rho_n]) \tag{A.8}$$

where $[q_0,\rho_1] \in (\mathsf{Lan}\,I)S$, $[q_n,\rho_n] \in (\mathsf{Lan}\,F)S$, $[(q'_{r-1},a_r,q_r),\rho_r] \in (\mathsf{Lan}\,\delta)S$ for $r = 1,\dots,n$ (putting $q'_0 := q_0$), and $[q_r,\rho_r] = [q'_r,\rho_{r+1}]$ for $r = 1,\dots,n-1$. We will show that we can choose the representatives such that $q_r = q'_r$ for $r = 1,\dots,n-1$ and $\rho_1 = \dots = \rho_n =: \rho$. Then $(q_0,a_1,q_1,\dots,a_n,q_n) \in \mathsf{AccRun}^{(n)}_A$ and thus ε_S maps $[(q_0,a_1,q_1,\dots,a_n,q_n),\rho]$ to (A.8).

Suppose that we already have $q_r = q'_r$ for $r = 0,\dots,m-1$ and $\rho_1 = \dots = \rho_m =: \rho$ for some $m < n$. We show that we can modify q'_m, q_{m+1}, ρ and ρ_{m+1} in such a way that this property also holds after $m+1$ steps. This is achieved by suitable choice of permutations and fresh names, much like in the proof of Proposition 2.8. Recall that we assume the presheaf Q to be strong, that is, $Q = \coprod_{j\in J} \mathbb{I}(S_j,-)$ for some finite set J and $S_j \subseteq_{\mathsf{f}} \mathbb{A}$.

1. Let $\rho\colon T \to S$ and $\rho_{m+1}\colon T_{m+1} \to S$. Since $[q_m,\rho] = [q'_m,\rho_{m+1}]$ in $\mathsf{Lan}\,Q$, the states q_m, q'_m must belong to the same summand of Q, that is, $q_m \in \mathbb{I}(S_j,T)$ and $q'_m \in \mathbb{I}(S_j,T_{m+1})$ for some $j \in J$. Let $T + T_{m+1}$ denote the disjoint union of T and T_{m+1} with injections $\mathsf{inl}, \mathsf{inr}$. Then we have the following zig-zag in $E{\downarrow}S$:

$$\begin{tikzcd} & S & \\ ET \arrow[ur, "\rho"] \arrow[r, "E\,\mathsf{inl}"'] & E(T+T_{m+1}) \arrow[u, "{[\rho,\rho_{m+1}]}"'] & ET_{m+1} \arrow[ul, "\rho_{m+1}"'] \arrow[l, "E\,\mathsf{inr}"] \end{tikzcd}$$

Hence, by replacing T and T_{m+1} with $T + T_{m+1}$, we may assume that $T = T_{m+1}$.

2. Since $q_m, q'_m\colon S_j \to T$ are injective maps, there exists a bijection $\pi\colon T \to T$ such that $q_m = \pi \circ q'_m$, witnessing that

$$[(q'_m, a_{m+1}, q_{m+1}), \rho_{m+1}] = [(q_m, \pi(a_{m+1}), \pi\circ q_{m+1}), \rho_{m+1}\circ\pi^{-1}] \quad \text{in} \quad \mathsf{Lan}\,\delta,$$

in particular $[q'_m,\rho_{m+1}] = [q_m, \rho_{m+1}\circ\pi^{-1}]$ in $\mathsf{Lan}\,Q$. Therefore, after replacing ρ_{m+1} by $\rho_{m+1}\circ\pi^{-1}$ and q'_m by q_m, we may assume that $q_m = q'_m$.

3. Finally, we consider the following zig-zag in $E{\downarrow}S$:

$$\begin{tikzcd} & S & \\ ET \arrow[ur, "\rho"] \arrow[r, "E\,\mathsf{inl}"'] & E(T+T) \arrow[u, "{[\rho,\rho_{m+1}]}"'] & ET \arrow[ul, "\rho_{m+1}"'] \arrow[l, "E\iota"] \end{tikzcd}$$

where $\iota\colon T \to T + T$ is the injective map sending every element $a \in q_m[S_j] \subseteq T$ to $\mathsf{inl}(a)$, and every other element of T to $\mathsf{inr}(a)$. Note that $\mathsf{inl}\circ q_m = \iota\circ q_m$ and that the right-hand triangle commutes: since $[q_m,\rho] = [q_m,\rho_{m+1}]$ we have $\rho\circ q_m = \rho_{m+1}\circ q_m$, hence the maps $\rho, \rho_{m+1}\colon ET \to S$ agree on $q_m[S_j] \subseteq T$. Therefore, after replacing ρ and ρ_{m+1} with $[\rho,\rho_{m+1}]$, $q_0, q_1, \dots, q_m$ by $\mathsf{inl}\circ q_0, \dots, \mathsf{inl}\circ q_m = \iota\circ q_m$, q_{m+1} by $\iota(q_{m+1})$ and a by $\iota(a)$, we can assume that $\rho = \rho_{m+1}$.

This concludes the proof of Proposition 7.5.

Proof of Theorem 7.7.1

▶ **Remark A.30.** For a presheaf automaton $A = (Q, V_{\mathscr{C}}, \delta, I, F)$ in $\mathbf{Set}^{\mathscr{C}}$, $\mathscr{C} \in \{\mathbb{I},\mathbb{F}\}$, we write $q \xrightarrow[S]{a} q'$ if $(q,a,q') \in \delta S$ for $S \subseteq_{\mathsf{f}} \mathbb{A}$. The accepted word language $\mathsf{W}(L(A))$ is the set of all $a_1 \dots a_n \in \mathbb{A}^\star$ for which there exists an accepting run, i.e. a sequence of transitions $q_0 \xrightarrow[S]{a_1} q_1 \xrightarrow[S]{a_2} \cdots \xrightarrow[S]{a_n} q_n$ where $S \subseteq_{\mathsf{f}} \mathbb{A}$, $q_0 \in IS$ and $q_n \in FS$.

▶ **Remark A.31.** We recall the left adjoint $I^\star\colon \mathbf{Set}^{\mathbb{I}} \to \mathbf{Nom}$ of $I_\star$, a.k.a. the *sheafification functor* [30, Lem. 6.7]. For each $P \in \mathbf{Set}^{\mathbb{I}}$, the nominal set $I^\star P$ is defined as follows:

- The underlying set of $I^\star P$ is the colimit of the directed diagram

  $$D_P\colon \mathbb{I}_{\subseteq} \hookrightarrow \mathbb{I} \xrightarrow{P} \mathbf{Set},$$

  where $\mathbb{I}_{\subseteq}$ is the poset of finite subsets of $\mathbb{A}$, i.e. the restriction of $\mathbb{I}$ to inclusion maps $i_{S,T}\colon S \xrightarrow{\subseteq} T$ for $S \subseteq T \subseteq_{\mathsf{f}} \mathbb{A}$. More explicitly, elements of $I^\star P$ are equivalence classes $[S,x]$ for the equivalence relation on $\coprod_{S\subseteq_{\mathsf{f}}\mathbb{A}} PS = \{\,(S,x) : S \subseteq_{\mathsf{f}} \mathbb{A},\, x \in PS\,\}$ given by

  $$(S,x) \sim (S',x') \qquad \text{iff} \qquad \exists T \supseteq S,S'.\, Pi_{S,T}(x) = Pi_{S,T'}(x').$$

- The group action on $I^\star P$ is given by

  $$\pi\cdot[S,x] = [\pi[S], P\pi|_S(x)] \qquad \text{for } \pi \in \mathsf{Perm}(\mathbb{A}) \text{ and } [S,x] \in I^\star P,$$

  with $\pi|_S\colon S \to \pi[S]$ denoting the domain-codomain restriction of $\pi\colon \mathbb{A} \to \mathbb{A}$.

Since directed colimits in **Set** commute with finite limits, the left adjoint $I^\star$ preserves finite limits, in particular products and monomorphisms. In fact, this property holds in general for sheafification functors [25, Thm. III.5.1].

One direction of Theorem 7.7.1 is established by the following lemma. Recall the embedding $I_\star\colon \mathbf{Nom} \to \mathbf{Set}^{\mathbb{I}}$ (Section 5) and its lifting $\bar{I}_\star\colon \mathbf{NAut}_{\mathsf{fp}}(\mathbf{Nom}) \to \mathbf{NAut}_{\mathsf{fp}}(\mathbf{Set}^{\mathbb{I}})$ from (6.2).

▶ **Lemma A.32.** *Every NOFA A is word-language equivalent to the* $\mathbf{Set}^{\mathbb{I}}$*-automaton $\bar{I}_\star A$.*

Proof. By definition of $I_\star$, every accepting run

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n \tag{A.9}$$

of the NOFA A yields the accepting run

$$q_0 \xrightarrow[S]{a_1} q_1 \xrightarrow[S]{a_2} \cdots \xrightarrow[S]{a_n} q_n \tag{A.10}$$

of the presheaf automaton $\bar{I}_\star A$, where $S \subseteq_{\mathsf{f}} \mathbb{A}$ is any set of names containing $a_1,\ldots,a_n$ and supporting $q_0,\ldots,q_n$. Conversely, every accepting run (A.10) of $\bar{I}_\star A$ yields the accepting run (A.9) of A. ◀

Similarly, for the reverse direction we use the lifting $\bar{I}^\star\colon \mathbf{NAut}_{\mathsf{fp}}(\mathbf{Set}^{\mathbb{I}}) \to \mathbf{NAut}_{\mathsf{fp}}(\mathbf{Nom})$.

▶ **Lemma A.33.** *Every super-finitary nondeterministic* $\mathbf{Set}^{\mathbb{I}}$*-automaton A is word-language equivalent to the NOFA $\bar{I}^\star A$.*

Proof. The inclusion $W(L(A)) \subseteq L(\bar{I}^\star A)$ holds because every accepting run

$$q_0 \xrightarrow[S]{a_1} q_1 \xrightarrow[S]{a_2} \cdots \xrightarrow[S]{a_n} q_n$$

of A yields the accepting run

$$[S,q_0] \xrightarrow{a_1} [S,q_1] \xrightarrow{a_2} \cdots \xrightarrow{a_n} [S,q_n]$$

of $\bar{I}^\star A$. For the proof of $L(\bar{I}^\star A) \subseteq W(L(A))$, suppose that $a_1\cdots a_n \in L(\bar{I}^\star A)$. By definition of $\bar{I}^\star A$, an accepting run of $a_1\cdots a_n$ then has the form

$$[S_1,q_0] \xrightarrow{a_1} [S_1,q_1] = [S_2,q_1'] \xrightarrow{a_2} [S_2,q_2] = [S_3,q_2'] \xrightarrow{a_3} \cdots$$
$$\cdots \xrightarrow{a_{n-1}} [S_{n-1},q_{n-1}] = [S_n,q_{n-1}'] \xrightarrow{a_n} [S_n,q_n]$$

where $q'_{r-1} \xrightarrow[S_r]{a_r} q_r$ in A (putting $q'_0 := q_0$) for $r = 1, \ldots, n$, and $[S_1, q_0] = [\overline{S}_1, \overline{q}_0]$ for some $\overline{S}_1 \subseteq_{\mathsf{f}} \mathbb{A}$ and $\overline{q}_0 \in I\overline{S}_1$, and $[S_n, q_n] = [\overline{S}_n, \overline{q}_n]$ for some $\overline{S}_n \subseteq_{\mathsf{f}} \mathbb{A}$ and $\overline{q}_n \in F\overline{S}_n$. Replacing the sets $S_1, \ldots, S_n, \overline{S}_1, \overline{S}_n$ by their union we can assume that $S_1 = \cdots S_n = \overline{S}_1 = \overline{S}_n =: S$. Since $[S, q_r] = [S, q'_r]$ we know that there exists $T_r \supseteq S$ such that $Pi_{S,T}(q_r) = Pi_{S,T}(q'_r)$. Similarly, we have sets $\overline{T}_0, \overline{T}_n \supseteq S$ witnessing that $[S, q_0] = [S, \overline{q}_0]$ and $[S, q_n] = [S, \overline{q}_n]$. Taking the union again, we can assume that $T_1 = \cdots T_n = \overline{T}_0 = \overline{T}_n =: T$. Hence, after replacing $q_1, \ldots, q_n, \overline{q}_0, \overline{q}_n$ by $Pi_{S,T}(q_1), \ldots Pi_{S,T}(q_n), Pi_{S,T}(\overline{q}_0), Pi_{S,T}(\overline{q}_n)$ we can assume that $q_r = q'_r$ for $r = 1, \ldots, n$ and $q_0 = \overline{q_0} \in IT$ and $q_n = \overline{q}_n \in FT$. We conclude that

$$q_0 \xrightarrow[T]{a_1} q_1 = q'_1 \xrightarrow[T]{a_2} q_2 = q'_2 \xrightarrow[T]{a_3} \cdots q_{n-1} = q'_{n-1} \xrightarrow[T]{a_n} q_n$$

is an accepting run of A, proving $L(\bar{I}^\star A) \subseteq W(L(A))$. ◀

Proof of Theorem 7.7.2

One may argue analogously to Theorem 7.7.1, replacing $\mathbf{Set}^{\mathbb{I}}$ by $\mathbf{Set}^{\mathbb{F}}$ and NOFA by NOFRA (which are equivalent to NOFA for positive word languages by Theorem 2.9). We give an alternative argument that relates $\mathbf{Set}^{\mathbb{I}}$- and $\mathbf{Set}^{\mathbb{F}}$-automata in a more direct manner. By Theorem 7.7.1 it suffices to prove the following two lemmas.

▶ **Lemma A.34.** *Every super-finitary nondeterministic* $\mathbf{Set}^{\mathbb{F}}$*-automaton A accepts a positive word language and is word-language equivalent to the super-finitary nondeterministic* $\mathbf{Set}^{\mathbb{I}}$*-automaton $\bar{E}^\star A$.*

Proof. Let A be a super-finitary nondeterministic $\mathbf{Set}^{\mathbb{F}}$-automaton. We first prove that $\mathsf{W}(L(A))$ is a positive word language. Given $a_1 \cdots a_n \in \mathsf{W}(L(A))$ and a renaming $\rho\colon \mathbb{A} \to \mathbb{A}$, choose $S \subseteq_{\mathsf{f}} \mathbb{A}$ such that $a_1 \cdots a_n \in L(A)(S)$. Then $\rho(a_1) \cdots \rho(a_n) \in L(A)(\rho[S])$ because $L(A) \subseteq V_{\mathbb{F}}^\star$ is a sub-presheaf. Hence $\rho(a_1) \cdots \rho(a_n) \in \mathsf{W}(L(A))$, so $\mathsf{W}(L(A))$ is positive.

Since $E^\star\colon \mathbf{Set}^{\mathbb{F}} \to \mathbf{Set}^{\mathbb{I}}$ is just a forgetful functor, clearly A is word-language equivalent to the $\mathbf{Set}^{\mathbb{I}}$-automaton $\bar{E}^\star A$: both automata have the same accepting runs. ◀

▶ **Remark A.35.** For every presheaf language $L \subseteq V_{\mathbb{I}}^\star$, the positive closure $\overline{L} \subseteq V_{\mathbb{F}}^\star$ is given at $S \subseteq_{\mathsf{f}} \mathbb{A}$ by

$$\overline{L}(S) = \{\, \rho^\star(w) : w \in L(T) \text{ and } \rho \in \mathbb{F}(T, S) \text{ for some } T \subseteq_{\mathsf{f}} \mathbb{A} \,\}.$$

Indeed, this language clearly satisfies the universal property of Definition 7.2. In particular, if $W(L)$ is a positive word language, then $W(\overline{L}) = W(L)$.

▶ **Lemma A.36.** *Every super-finitary nondeterministic* $\mathbf{Set}^{\mathbb{I}}$*-automaton A accepting a positive word language is word-language equivalent to the super-finitary nondeterministic* $\mathbf{Set}^{\mathbb{F}}$*-automaton $\overline{\mathsf{Lan}}_E A$.*

Proof. Let A be a super-finitary nondeterministic $\mathbf{Set}^{\mathbb{I}}$-automaton such that $\mathsf{W}(L(A))$ is a positive word language. Assuming w.l.o.g. that A has a strong presheaf of states, by Proposition 7.5 the automaton $\overline{\mathsf{Lan}}_E A$ accepts the positive closure $\overline{L(A)}$ of $L(A)$, and Remark A.35 shows that $\mathsf{W}(L(A)) = \mathsf{W}(\overline{L(A)})$. Hence A and $\overline{\mathsf{Lan}}_E A$ are word-language equivalent. ◀