Nominal Büchi Automata with Name Allocation

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

Daniel Hausmann
Gothenburg University, Göteborg, Sweden

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

Lutz Schröder
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract

Infinite words over infinite alphabets serve as models of the temporal development of the allocation and (re-)use of resources over linear time. We approach ω-languages over infinite alphabets in the setting of nominal sets, and study languages of *infinite bar strings*, i.e. infinite sequences of names that feature binding of fresh names; binding corresponds roughly to reading letters from input words in automata models with registers. We introduce *regular nominal nondeterministic Büchi automata* (*Büchi RNNAs*), an automata model for languages of infinite bar strings, repurposing the previously introduced *RNNAs* over finite bar strings. Our machines feature explicit binding (i.e. resource-allocating) transitions and process their input via a Büchi-type acceptance condition. They emerge from the abstract perspective on name binding given by the theory of nominal sets. As our main result we prove that, in contrast to most other nondeterministic automata models over infinite alphabets, language inclusion of Büchi RNNAs is decidable and in fact elementary. This makes Büchi RNNAs a suitable tool for applications in model checking.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases Data languages, infinite words, nominal sets, inclusion checking

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2021.4

Related Version *Full Version*: http://arxiv.org/abs/2107.03213

Funding *Henning Urbat*: Supported by Deutsche Forschungsgemeinschaft (DFG) under project SCHR 1118/15-1.
Daniel Hausmann: Supported by Deutsche Forschungsgemeinschaft (DFG) under project MI 717/7-1 and by the European Research Council (ERC) under project 772459.
Stefan Milius: Supported by Deutsche Forschungsgemeinschaft (DFG) under project MI 717/7-1.
Lutz Schröder: Supported by Deutsche Forschungsgemeinschaft (DFG) under project SCHR 1118/15-1.

1 Introduction

Classical automata models and formal languages for finite words over finite alphabets have been extended to infinity in both directions: Infinite words model the long-term temporal development of systems, while infinite alphabets model *data*, such as nonces [19], object identities [14], or abstract resources [5]. We approach *data ω-languages*, i.e. languages of infinite words over infinite alphabets, in the setting of nominal sets [25] where elements of sets are thought of as carrying (finitely many) *names* from a fixed countably infinite reservoir. Following the paradigm of nominal automata theory [3], we take the set of names as the alphabet; we work with infinite words containing explicit name binding in the spirit of previous work on nominal formalisms for finite words such as nominal Kleene algebra [11] and regular nondeterministic nominal automata (RNNAs) [26]. Name binding may be viewed

32nd International Conference on Concurrency Theory (CONCUR 2021).
Editors: Serge Haddad and Daniele Varacca; Article No. 4; pp. 4:1–4:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

as the allocation of resources, or (yet) more abstractly as an operation that reads fresh names from the input. We refer to such infinite words as *infinite bar strings*, in honour of the vertical bar notation we employ for name binding. In the present paper, we introduce a notion of nondeterministic nominal automata over infinite bar strings, and show that it admits inclusion checking in elementary complexity.

Specifically, we reinterpret the mentioned RNNAs to accept infinite rather than finite bar strings by equipping them with a Büchi acceptance condition; like over finite alphabets, automata with more expressive acceptance conditions including Muller acceptance can be translated into the basic (nondeterministic) Büchi model. Our mentioned main result then states that language inclusion of Büchi RNNAs is decidable in parametrized polynomial space [27], with a parameter that may be thought as the number of registers. This is in sharp contrast to other nondeterministic automata models for infinite words over infinite alphabets, which sometimes have decidable emptiness problems but whose inclusion problems are typically either undecidable or of prohibitively high complexity even under heavy restrictions (inclusion is not normally reducible to emptiness since nondeterministic models typically do not determinize, and in fact tend to fail to be closed under complement); details are in the related work section.

Infinite bar strings can be concretized to infinite strings of names (i.e. essentially to data words) by interpreting name binding as reading either globally fresh letters (in which case Büchi RNNAs may essentially be seen as a variant of session automata [4] for infinite words) or locally fresh letters. Both interpretations arise from disciplines of α-renaming as known from λ-calculus [1], with global freshness corresponding to a discipline of *clean* naming where bound names are never shadowed, and local freshness corresponding to an unrestricted naming discipline that does allow shadowing. The latter implies that local freshness can only be enforced w.r.t. names that are expected to be seen again later in the word. It is precisely this fairly reasonable-sounding restriction that buys the comparatively low computational complexity of the model, which on the other hand allows full nondeterminism (and, e.g., accepts the language 'some letter occurs infinitely often', which is not acceptable by deterministic register-based models) and unboundedly many registers. Büchi RNNAs thus provide a reasonably expressive automata model for infinite data words whose main reasoning problems are decidable in elementary complexity.

Related Work. Büchi RNNAs generally adhere to the paradigm of register automata, which in their original incarnation over finite words [16] are equivalent to the nominal automaton model of nondeterministic orbit-finite automata [3]. Ciancia and Sammartino [5] study *deterministic* nominal Muller automata accepting infinite strings of names, and show Boolean closure and decidability of inclusion. This model is incomparable to ours; details are in Section 7. For alternating register automata over infinite data words, emptiness is undecidable even when only one register is allowed (which over finite data words does ensure decidability) [9]. For the one-register safety fragment of the closely related logic Freeze LTL, inclusion (i.e. refinement) is decidable, but even the special case of validity is not primitive recursive [21], in particular not elementary.

Many automata models and logics for data words deviate rather substantially from the register paradigm, especially models in the vicinity of data automata [2], whose emptiness problem is decidable but at least as hard as Petri net reachability, which by recent results is not elementary [7], in fact Ackermann-complete [8, 22]. For weak Büchi data automata [17], the emptiness problem is decidable in elementary complexity. Similarly, Büchi generalized data automata [6] have a decidable emptiness problem; their Büchi component is deterministic.

(Throughout, nothing appears to be said about inclusion of data automata.) Variable finite automata [15] apply to both finite and infinite words; in both versions, the inclusion problem of the nondeterministic variant is undecidable.

2 Preliminaries: Nominal Sets

Nominal sets form a convenient formalism for dealing with names and freshness; for our present purposes, names play the role of data. We briefly recall basic notions and facts and refer to [25] for a comprehensive introduction. Fix a countably infinite set $\mathbb{A}$ of *names*, and let $\mathsf{Perm}(\mathbb{A})$ denote the group of finite permutations on $\mathbb{A}$, which is generated by the *transpositions* $(a\,b)$ for $a \neq b \in \mathbb{A}$ (recall that $(a\,b)$ just swaps a and b). A *nominal set* is a set X equipped with a (left) group action $\mathsf{Perm}(\mathbb{A}) \times X \to X$, denoted $(\pi, x) \mapsto \pi \cdot x$, such that every element $x \in X$ has a finite *support* $S \subseteq \mathbb{A}$, i.e. $\pi \cdot x = x$ for every $\pi \in \mathsf{Perm}(\mathbb{A})$ such that $\pi(a) = a$ for all $a \in S$. Every element x of a nominal set X has a least finite support, denoted $\mathsf{supp}(x)$. Intuitively, one should think of X as a set of syntactic objects (e.g. strings, λ-terms, programs), and of $\mathsf{supp}(x)$ as the set of names needed to describe an element $x \in X$. A name $a \in \mathbb{A}$ is *fresh* for x, denoted $a \# x$, if $a \notin \mathsf{supp}(x)$. The *orbit* of an element $x \in X$ is given by $\{\pi \cdot x : \pi \in \mathsf{Perm}(\mathbb{A})\}$. The orbits form a partition of X. The nominal set X is *orbit-finite* if it has only finitely many orbits.

Putting $\pi \cdot a = \pi(a)$ makes $\mathbb{A}$ into a nominal set. Moreover, $\mathsf{Perm}(\mathbb{A})$ acts on subsets $A \subseteq X$ of a nominal set X by $\pi \cdot A = \{\pi \cdot x : x \in A\}$. A subset $A \subseteq X$ is *equivariant* if $\pi \cdot A = A$ for all $\pi \in \mathsf{Perm}(\mathbb{A})$. More generally, it is *finitely supported* if it has finite support w.r.t. this action, i.e. there exists a finite set $S \subseteq \mathbb{A}$ such that $\pi \cdot A = A$ for all $\pi \in \mathsf{Perm}(\mathbb{A})$ such that $\pi(a) = a$ for all $a \in S$. The set A is *uniformly finitely supported* if $\bigcup_{x \in A} \mathsf{supp}(x)$ is a finite set. This implies that A is finitely supported, with least support $\mathsf{supp}(A) = \bigcup_{x \in A} \mathsf{supp}(x)$ [10, Theorem 2.29]. (The converse does not hold, e.g. the set $\mathbb{A}$ is finitely supported but not uniformly finitely supported.) Uniformly finitely supported orbit-finite sets are always finite (since an orbit-finite set contains only finitely many elements with a given finite support). We respectively denote by $\mathcal{P}_{\mathsf{ufs}}(X)$ and $\mathcal{P}_{\mathsf{fs}}(X)$ the nominal sets of (uniformly) finitely supported subsets of a nominal set X.

A map $f\colon X \to Y$ between nominal sets is *equivariant* if $f(\pi \cdot x) = \pi \cdot f(x)$ for all $x \in X$ and $\pi \in \mathsf{Perm}(\mathbb{A})$. Equivariance implies $\mathsf{supp}(f(x)) \subseteq \mathsf{supp}(x)$ for all $x \in X$. The function supp itself is equivariant, i.e. $\mathsf{supp}(\pi \cdot x) = \pi \cdot \mathsf{supp}(x)$ for $\pi \in \mathsf{Perm}(\mathbb{A})$. Hence $|\mathsf{supp}(x_1)| = |\mathsf{supp}(x_2)|$ whenever x_1, x_2 are in the same orbit of a nominal set.

We denote by **Nom** the category of nominal sets and equivariant maps. The object maps $X \mapsto \mathcal{P}_{\mathsf{ufs}}(X)$ and $X \mapsto \mathcal{P}_{\mathsf{fs}}(X)$ extend to endofunctors $\mathcal{P}_{\mathsf{ufs}}\colon \mathbf{Nom} \to \mathbf{Nom}$ and $\mathcal{P}_{\mathsf{fs}}\colon \mathbf{Nom} \to \mathbf{Nom}$ sending an equivariant map $f\colon X \to Y$ to the map $A \mapsto f[A]$.

The coproduct $X + Y$ of nominal sets X and Y is given by their disjoint union with the group action inherited from the two summands. Similarly, the product $X \times Y$ is given by the cartesian product with the componentwise group action; we have $\mathsf{supp}(x, y) = \mathsf{supp}(x) \cup \mathsf{supp}(y)$. Given a nominal set X equipped with an equivariant equivalence relation, i.e. an equivalence relation $\sim$ that is equivariant as a subset $\sim\, \subseteq X \times X$, the quotient $X/{\sim}$ is a nominal set under the expected group action defined by $\pi \cdot [x]_\sim = [\pi \cdot x]_\sim$.

A key role in the technical development is played by *abstraction sets*, which provide a semantics for binding mechanisms [12]. Given a nominal set X, an equivariant equivalence relation $\sim$ on $\mathbb{A} \times X$ is defined by $(a, x) \sim (b, y)$ iff $(a\,c) \cdot x = (b\,c) \cdot y$ for some (equivalently, all) fresh c. The *abstraction set* $[\mathbb{A}]X$ is the quotient set $(\mathbb{A} \times X)/{\sim}$. The $\sim$-equivalence class of $(a, x) \in \mathbb{A} \times X$ is denoted by $\langle a \rangle x \in [\mathbb{A}]X$. We may think of $\sim$ as an abstract notion of α-equivalence, and of $\langle a \rangle$ as binding the name a. Indeed we have $\mathsf{supp}(\langle a \rangle x) = \mathsf{supp}(x) \setminus \{a\}$ (while $\mathsf{supp}(a, x) = \{a\} \cup \mathsf{supp}(x)$), as expected in binding constructs.

The object map $X \mapsto [\mathbb{A}]X$ extends to an endofunctor $[\mathbb{A}]\colon \mathbf{Nom} \to \mathbf{Nom}$ sending an equivariant map $f\colon X \to Y$ to the equivariant map $[\mathbb{A}]f\colon [\mathbb{A}]X \to [\mathbb{A}]Y$ given by $\langle a \rangle x \mapsto \langle a \rangle f(x)$ for $a \in \mathbb{A}$ and $x \in X$.

3 The Notion of α-Equivalence for Bar Strings

In the following we investigate automata consuming input words over the infinite alphabet

$$\overline{\mathbb{A}} := \mathbb{A} \cup \{|a : a \in \mathbb{A}\}.$$

A *finite bar string* is a finite word $\sigma_1\sigma_2\cdots\sigma_n$ over $\overline{\mathbb{A}}$, and *infinite bar string* is an infinite word $\sigma_1\sigma_2\sigma_3\cdots$ over $\overline{\mathbb{A}}$. We denote the sets of finite and infinite bar strings by $\overline{\mathbb{A}}^*$ and $\overline{\mathbb{A}}^\omega$, respectively. Given $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega$ the set of *names* in w is defined by

$$\mathsf{N}(w) = \{a \in \mathbb{A} : \text{the letter } a \text{ or } |a \text{ occurs in } w\}.$$

An infinite bar string w is *finitely supported* if $\mathsf{N}(w)$ is a finite set; we let $\overline{\mathbb{A}}^\omega_{\mathsf{fs}} \subseteq \overline{\mathbb{A}}^\omega$ denote the set of finitely supported infinite bar strings. Note that $\overline{\mathbb{A}}^*$ and $\overline{\mathbb{A}}^\omega_{\mathsf{fs}}$ are nominal sets w.r.t. the group action defined pointwise. The least support of a bar string is its set of names.

A bar string containing only letters from $\mathbb{A}$ is called a *data word*. We denote by $\mathbb{A}^* \subseteq \overline{\mathbb{A}}^*$, $\mathbb{A}^\omega \subseteq \overline{\mathbb{A}}^\omega$ and $\mathbb{A}^\omega_{\mathsf{fs}} \subseteq \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$ the sets of finite data words, infinite data words, and finitely supported infinite data words, respectively.

We interpret $|a$ as binding the name a to the right. Accordingly, a name $a \in \mathbb{A}$ is said to be *free* in a bar string $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega$ if (i) the letter a occurs in w, and (ii) the first occurrence of a is not preceded by any occurrence of $|a$. For instance, the name a is free in $a|aba$ but not free in $|aaba$, while the name b is free in both bar strings. We put

$$\mathsf{FN}(w) = \{a \in \mathbb{A} : a \text{ is free in } w\}.$$

We obtain a natural notion of α-equivalence for both finite and infinite bar strings; the finite case is taken from [26].

▶ **Definition 3.1** (α-equivalence)**.** Let $=_\alpha$ be the least equivalence relation on $\overline{\mathbb{A}}^*$ such that

$$x|av =_\alpha x|bw \qquad \text{for all } a, b \in \mathbb{A} \text{ and } x, v, w \in \overline{\mathbb{A}}^* \text{ such that } \langle a \rangle v = \langle b \rangle w.$$

This extends to an equivalence relation $=_\alpha$ on $\overline{\mathbb{A}}^\omega$ given by

$$v =_\alpha w \qquad \text{iff} \qquad v_n =_\alpha w_n \quad \text{for all } n \in \mathbb{N},$$

where v_n and w_n are the prefixes of length n of v and w. We denote by $\overline{\mathbb{A}}^*/{=_\alpha}$ and $\overline{\mathbb{A}}^\omega/{=_\alpha}$ the sets of α-equivalence classes of finite and infinite bar strings, respectively, and we write $[w]_\alpha$ for the α-equivalence class of $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega$.

▶ **Remark 3.2.**

1. For any $v, w \in \overline{\mathbb{A}}^*$ the condition $\langle a \rangle v = \langle b \rangle w$ holds if and only if

$$a = b \text{ and } v = w, \qquad \text{or} \qquad b \,\#\, v \text{ and } (a\, b) \cdot v = w.$$

2. The α-equivalence relation is a *left congruence*:

$$v =_\alpha w \quad \text{implies} \quad xv =_\alpha xw \qquad\qquad \text{for all } v, w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega \text{ and } x \in \overline{\mathbb{A}}^*.$$

Moreover, the *right cancellation property* holds:

$$vx =_\alpha wx \quad \text{implies} \quad v =_\alpha w \qquad\qquad \text{for all } v, w \in \overline{\mathbb{A}}^* \text{ and } x \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega.$$

3. The equivalence relation $=_\alpha$ is equivariant. Therefore, both $\overline{\mathbb{A}}^*/{=_\alpha}$ and $\overline{\mathbb{A}}^\omega_{\mathsf{fs}}/{=_\alpha}$ are nominal sets with the group action $\pi \cdot [w]_\alpha = [\pi \cdot w]_\alpha$ for $\pi \in \mathsf{Perm}(\mathbb{A})$ and $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$. The least support of $[w]_\alpha$ is the set $\mathsf{FN}(w)$ of free names of w.

▶ **Remark 3.3.** Our notion of α-equivalence on infinite bar strings differs from the equivalence relation generated by relating $x|av$ and $x|bw$ whenever $\langle a\rangle v = \langle b\rangle w$ (as in Definition 3.1 but now for infinite bar strings $v, w \in \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$): The latter equivalence relates two bar strings iff they can be transformed into each other by finitely many α-renamings, while the definition of α-equivalence as per Definition 3.1 allows infinitely many simultaneous α-renamings; e.g. the infinite bar string $(|aa)^\omega = |aa|aa|aa|aa\cdots$ is α-equivalent to $(|aa|bb)^\omega = |aa|bb|aa|bb\cdots$.

▶ **Definition 3.4.** A *literal language*, *bar language* or *data language* is a subset of $\overline{\mathbb{A}}^*$, $\overline{\mathbb{A}}^*/{=_\alpha}$ or $\mathbb{A}^*$, respectively. Similarly, a *literal ω-language*, *bar ω-language* or *data ω-language* is a subset of $\overline{\mathbb{A}}^\omega$, $\overline{\mathbb{A}}^\omega/{=_\alpha}$ or $\mathbb{A}^\omega$, respectively.

▶ **Notation 3.5.** Given a finite or infinite bar string $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega$ we let $\mathsf{ub}(w)$ denote the data word obtained by replacing every occurrence of $|a$ in w by a; e.g. $\mathsf{ub}(|aa|a|bb) = aaabb$. To every bar ($\omega$-)language L we associate the data (ω-)language $D(L)$ given by

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

▶ **Definition 3.6.** A finite or infinite bar string w is *clean* if for each $a \in \mathsf{FN}(w)$ the letter $|a$ does not occur in w, and for each $a \notin \mathsf{FN}(w)$ the letter $|a$ occurs at most once.

▶ **Lemma 3.7.** *Every bar string $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$ is α-equivalent to a (not necessarily finitely supported) clean bar string.*

Proof. Since w has finite support, for every occurrence of the letter $|a$ for which a or $|a$ has already occurred before, one can replace $|a$ by $|b$ for some fresh name b and replace the suffix v after $|a$ by $(a\,b)v$. Iterating this yields a clean bar string α-equivalent to w. ◀

▶ **Example 3.8.**

1. The finitely supported infinite bar string $|aa|bb|aa|bb\cdots$ is α-equivalent to the clean bar string $|a_1a_1|a_2a_2|a_3a_3|a_4a_4\cdots$ where the a_i are pairwise distinct names. Note that $|aa|bb|aa|bb\cdots$ is not α-equivalent to any finitely supported clean bar string.
2. For non-finitely supported bar strings the lemma generally fails: if $\mathbb{A} = \{a_1, a_2, a_3, \cdots\}$ then the bar string $a_1|a_1a_2a_3a_4a_5a_6a_7\cdots$ is not α-equivalent to any other bar string, in particular not to a clean one.

For readers familiar with the theory of coalgebras we note that on infinite bar strings with finite support, α-equivalence naturally emerges from a coinductive point of view. Kurz et al. [20] use coinduction to devise a general notion of α-equivalence for infinitary terms over a binding signature, which form the final colgebra for an associated endofunctor on **Nom**. The following is the special case for the endofunctors

$$GX = \overline{\mathbb{A}} \times X \cong \mathbb{A} \times X + \mathbb{A} \times X \qquad \text{and} \qquad FX = \mathbb{A} \times X + [\mathbb{A}]X.$$

Let νF and νG denote their respective final coalgebras. The coalgebra νG is carried by the nominal set $\overline{\mathbb{A}}^{\omega}_{\mathsf{fs}}$ with the usual coalgebra structure $\langle \mathsf{hd}, \mathsf{tl}\rangle \colon \overline{\mathbb{A}}^{\omega}_{\mathsf{fs}} \to \overline{\mathbb{A}} \times \overline{\mathbb{A}}^{\omega}_{\mathsf{fs}}$ decomposing an infinite string into its head and tail. The natural transformation

$$\sigma_X \colon GX \twoheadrightarrow FX \qquad \text{given by} \qquad (a,x) \mapsto (a,x), \quad (|a, x) \mapsto \langle a \rangle x,$$

then induces a canonical map $e_\alpha \colon \overline{\mathbb{A}}^{\omega}_{\mathsf{fs}} \to \nu F$, viz. the unique homomorphism from the F-coalgebra $\nu G \xrightarrow{\langle \mathsf{hd},\mathsf{tl}\rangle} G(\nu G) \xrightarrow{\sigma_{\nu G}} F(\nu G)$ into the final coalgebra νF.

▶ **Proposition 3.9.** *For every $v, w \in \overline{\mathbb{A}}^{\omega}_{\mathsf{fs}}$ we have $v =_\alpha w$ if and only if $e_\alpha(v) = e_\alpha(w)$.*

4 Automata over Infinite Bar Strings

We proceed to introduce Büchi RNNAs, our nominal automaton model for bar ω-languages and data ω-languages. It modifies regular nominal nondeterministic automata (RNNAs) [26], originally introduced for bar languages and data languages of finite words, to accept infinite words. Roughly, Büchi RNNAs are to RNNAs what classical Büchi automata [13] are to nondeterministic finite automata. We first recall:

▶ **Definition 4.1** (RNNA [26]).

1. An *RNNA* $A = (Q, R, q_0, F)$ is given by an orbit-finite nominal set Q of *states*, an equivariant relation $R \subseteq Q \times \overline{\mathbb{A}} \times Q$ specifying *transitions*, an *initial state* $q_0 \in Q$ and an equivariant set $F \subseteq Q$ of *final states*. We write $q \xrightarrow{\sigma} q'$ if $(q, \sigma, q') \in R$. The transitions are subject to two conditions:
   (1) *α-invariance*: if $q \xrightarrow{|a} q'$ and $\langle a \rangle q' = \langle b \rangle q''$, then $q \xrightarrow{|b} q''$.
   (2) *Finite branching up to α-invariance:* For each $q \in Q$ the sets $\{(a, q') : q \xrightarrow{a} q'\}$ and $\{\langle a \rangle q' : q \xrightarrow{|a} q'\}$ are finite (equivalently, uniformly finitely supported).
   The *degree* of A, denoted $\mathsf{deg}(A)$, is the maximum of all $|\mathsf{supp}(q)|$ where $q \in Q$.
2. Given a finite bar string $w = \sigma_1\sigma_2\cdots\sigma_n \in \overline{\mathbb{A}}^{*}$ and a state $q \in Q$, a *run* for w from q is a sequence of transitions

   $$q \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} q_n.$$

   The run is *accepting* if q_n is final. The state q *accepts* w if there exists an accepting run for w from q, and the automaton A *accepts* w if its initial state q_0 accepts w. We define

   $$\begin{aligned} L_0(A) &= \{w \in \overline{\mathbb{A}}^{*} : A \text{ accepts } w\}, && \text{the } \textit{literal language accepted by } A, \\ L_\alpha(A) &= \{[w]_\alpha : w \in \overline{\mathbb{A}}^{*},\ A \text{ accepts } w\}, && \text{the } \textit{bar language accepted by } A, \\ D(A) &= D(L_\alpha(A)), && \text{the } \textit{data language accepted by } A. \end{aligned}$$

▶ **Remark 4.2.** Equivalently, an RNNA is an orbit-finite coalgebra

$$Q \longrightarrow 2 \times \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times Q) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]Q)$$

for the functor $FX = 2 \times \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times X) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]X)$ on $\mathbf{Nom}$, with an initial state $q_0 \in Q$.

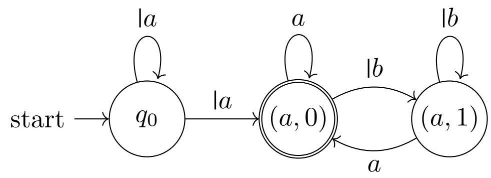
▶ **Definition 4.3** (Büchi RNNA). A *Büchi RNNA* is an RNNA $A = (Q, R, q_0, F)$ used to recognize infinite bar strings as follows. Given $w = \sigma_1\sigma_2\sigma_3 \cdots \in \overline{\mathbb{A}}^{\omega}$ and a state $q \in Q$, a *run* for w from q is an infinite sequence of transitions

$$q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \xrightarrow{\sigma_3} \cdots .$$

The run is *accepting* if q_n is final for infinitely many $n \in \mathbb{N}$. The state q *accepts* w if there exists an accepting run for w from q, and the automaton A *accepts* w if its initial state q_0 accepts w. We define

$$\begin{aligned} L_{0,\omega}(A) &= \{w \in \overline{\mathbb{A}}^{\omega} : A \text{ accepts } w\}, && \text{the } \textit{literal } \omega\textit{-language accepted by } A, \\ L_{\alpha,\omega}(A) &= \{[w]_\alpha : w \in \overline{\mathbb{A}}^{\omega},\ A \text{ accepts } w\}, && \text{the } \textit{bar } \omega\textit{-language accepted by } A, \\ D_\omega(A) &= D(L_{\alpha,\omega}(A)), && \text{the } \textit{data } \omega\textit{-language accepted by } A. \end{aligned}$$

▶ **Example 4.4.** Consider the Büchi RNNA A with states $\{q_0\} \cup \mathbb{A} \times \{0,1\}$ and transitions as displayed below, where a, b range over distinct names in $\mathbb{A}$. Note that the second node represents the orbit $\mathbb{A} \times \{0\}$ and the third one the orbit $\mathbb{A} \times \{1\}$.



The data ω-language $D_\omega(A)$ consists of all infinite words $w \in \mathbb{A}^\omega$ where some name $a \in \mathbb{A}$ occurs infinitely often.

▶ **Remark 4.5.** For automata consuming infinite words over infinite alphabets, a slightly subtle point is whether to admit arbitrary infinite words as inputs or restrict to finitely supported ones. For the bar language semantics of Büchi RNNAs this choice is inconsequential: we shall see in Proposition 4.7 below that modulo α-equivalence all infinite bar strings accepted by Büchi RNNA are finitely supported. However, for the data language semantics admitting strings with infinite support is important for the decidability of language inclusion, see Remark 6.8.

▶ **Lemma 4.6** ([26, Lem. 5.4]). *Let $A = (Q, R, q_0, F)$ be an RNNA, $q, q' \in Q$ and $a \in \mathbb{A}$.*
1. *If $q \xrightarrow{a} q'$ then $\mathsf{supp}(q') \cup \{a\} \subseteq \mathsf{supp}(q)$.*
2. *If $q \xrightarrow{|a} q'$ then $\mathsf{supp}(q') \subseteq \mathsf{supp}(q) \cup \{a\}$.*
3. *For every bar string $w \in \overline{\mathbb{A}}^* \cup \overline{\mathbb{A}}^\omega$ with a run from q one has $\mathsf{FN}(w) \subseteq \mathsf{supp}(q)$.*

The next proposition will turn out to be crucial in the development that follows. It asserts that, up to α-equivalence, one can always restrict the inputs of a Büchi RNNA to bar strings with a finite number of names, bounded by the degree of the automaton.

▶ **Proposition 4.7.** *Let A be a Büchi RNNA accepting the infinite bar string $w \in \overline{\mathbb{A}}^\omega$. Then it also accepts some $w' \in \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$ such that*

$$w' =_\alpha w \qquad \text{and} \qquad |\mathsf{supp}(q_0) \cup \mathsf{N}(w')| \leq \mathsf{deg}(A) + 1.$$

Proof sketch. Put $m := \mathsf{deg}(A)$, and choose $m+1$ pairwise distinct names $a_1, \ldots, a_{m+1}$ such that $\mathsf{supp}(q_0) \subseteq \{a_1, \ldots, a_{m+1}\}$.

1. One first shows that for every finite bar string $\sigma_1\sigma_2 \cdots \sigma_n \in \overline{\mathbb{A}}^*$ and every run

   $$q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \xrightarrow{\sigma_3} \cdots \xrightarrow{\sigma_n} q_n$$

   in A there exists $\sigma_1'\sigma_2' \cdots \sigma_n' \in \overline{\mathbb{A}}^*$ and a run

   $$q_0 \xrightarrow{\sigma_1'} q_1' \xrightarrow{\sigma_2'} q_2' \xrightarrow{\sigma_3'} \cdots \xrightarrow{\sigma_n'} q_n' \tag{4.1}$$

such that (1) q_i and q_i' lie in the same orbit for $i = 1, \dots, n$, (2) $\sigma_1'\sigma_2' \cdots \sigma_n' =_\alpha \sigma_1\sigma_2 \cdots \sigma_n$, and (3) $\mathsf{N}(\sigma_1'\sigma_2' \cdots \sigma_n') \subseteq \{a_1, \dots, a_{m+1}\}$. The proof is by induction on n and rests on the observation that for every state q at least one of the names $a_1, \dots, a_{m+1}$ is fresh for q.

2. Now suppose that the infinite bar string $w = \sigma_1\sigma_2\sigma_3 \cdots \in \overline{\mathbb{A}}^\omega$ is accepted by A via the accepting run

$$q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \xrightarrow{\sigma_3} \cdots$$

Consider the set of all partial runs (4.1) satisfying the above conditions (1)–(3). This set organizes into a tree with the edge relation given by extension of runs; its nodes of depth n are exactly the runs (4.1). By part 1 at least one such run exists for each $n \in \mathbb{N}$, i.e. the tree is infinite. It is finitely branching because $\mathsf{supp}(q_i') \subseteq \{a_1, \dots, a_{m+1}\}$ for all i by Lemma 4.6, i.e. there are only finitely many runs (4.1) for each n. Hence, by Kőnig's lemma the tree contains an infinite path. This yields an infinite run

$$q_0 \xrightarrow{\sigma_1'} q_1' \xrightarrow{\sigma_2'} q_2' \xrightarrow{\sigma_3'} \cdots$$

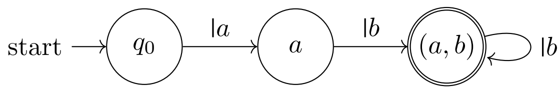
such that q_i' and q_i lie in the same orbit for each i and $\sigma_1' \cdots \sigma_n' =_\alpha \sigma_1 \cdots \sigma_n$ for each n. It follows that the Büchi RNNA A accepts the infinite bar string

$$w' = \sigma_1'\sigma_2'\sigma_3' \cdots .$$

Moreover $w' =_\alpha w$ and $\mathsf{supp}(q_0) \cup \mathsf{N}(w') \subseteq \{a_1, \dots, a_{m+1}\}$ as required. ◀

5 Name-Dropping Modification

Although the transitions of a Büchi RNNA are α-invariant, its literal ω-language generally fails to be closed under α-equivalence. For instance, consider the Büchi RNNA with states $\{q_0\} \cup \mathbb{A} \cup \mathbb{A}^2$ and transitions displayed below, where a, b range over distinct names in $\mathbb{A}$:



The automaton accepts the infinite bar string $\mathsf{I}a\mathsf{I}b\mathsf{I}b\mathsf{I}b\mathsf{I}b\mathsf{I}b \cdots$ for $a \neq b$ but does not accept the α-equivalent $\mathsf{I}a\mathsf{I}a\mathsf{I}a\mathsf{I}a\mathsf{I}a\mathsf{I}a \cdots$: the required transitions $a \xrightarrow{\mathsf{I}a} (a, b)$ and $(a, b) \xrightarrow{\mathsf{I}a} (a, b)$ do not exist since a is in the least support of the state (a, b), which prevents α-renaming of the given transitions $a \xrightarrow{\mathsf{I}b} (a, b)$ and $(a, b) \xrightarrow{\mathsf{I}b} (a, b)$. A possible fix is to add a new state "$(\bot, b)$" that essentially behaves like (a, b) but has the name a dropped from its least support.

This idea can be generalized: We will show below how to transform any Büchi RNNA A into a Büchi RNNA $\widetilde{A}$ such that $L_{0,\omega}(\widetilde{A})$ is the closure of $L_{0,\omega}(A)$ under α-equivalence, using the *name-dropping modification* (Definition 5.3). The latter simplifies a construction of the same name previously given for RNNA over finite words [26], and it is the key to the decidability of bar language inclusion proved later on. First, some technical preparations:

▶ **Remark 5.1** (Strong nominal sets). A nominal set Y is called *strong* [28] if for every $\pi \in \mathsf{Perm}(\mathbb{A})$ and $y \in Y$ one has $\pi \cdot y = y$ if and only if $\pi(a) = a$ for all $a \in \mathsf{supp}(y)$. (Note that "if" holds in all nominal sets.) As shown in [24, Lem. 2.4.2] or [23, Cor. B.27(2)] strong nominal sets are up to isomorphism precisely the nominal sets of the form $Y = \sum_{i \in I} \mathbb{A}^{\#X_i}$ where X_i is a finite set and $\mathbb{A}^{\#X}$ denotes the nominal set of all injective maps from X to $\mathbb{A}$, with the group action defined pointwise; $\mathbb{A}^{\#X}$ may be seen as the set of possible configurations of an X-indexed set of registers containing pairwise distinct names. The set of orbits of Y is in bijection with I; in particular, Y is orbit-finite iff I is finite. Strong nominal sets can be regarded as analogues of free algebras in categories of algebraic structures:

1. For every nominal set Z there exists a strong nominal set Y and a surjective equivariant map $e\colon Y \twoheadrightarrow Z$ that is supp-*nondecreasing*, i.e. $\mathsf{supp}(e(y)) = \mathsf{supp}(y)$ for all $y \in Y$ [23, Cor. B.27.1]. (Recall that $\mathsf{supp}(e(y)) \subseteq \mathsf{supp}(y)$ always holds for equivariant maps.) Specifically, one can choose $Y = \sum_{i\in I} \mathbb{A}^{\# n_i}$ where I is the set of orbits of Z and n_i is the size of the support of any element of the orbit i. In particular, if Z is orbit-finite then Y is orbit-finite with the same number of orbits.
2. Strong nominal sets are *projective* w.r.t. supp-nondecreasing quotients [23, Lem. B.28]: given a strong nominal set Y, an equivariant map $h\colon Y \to Z$ and a supp-nondecreasing quotient $e\colon X \twoheadrightarrow Z$, there exists an equivariant map $g\colon Y \to X$ such that $h = e \cdot g$.

▶ **Proposition 5.2.** *For every Büchi RNNA there exists a Büchi RNNA accepting the same literal ω-language whose states form a strong nominal set.*

Proof sketch. Given a Büchi RNNA viewed as a coalgebra

$$Q \xrightarrow{\gamma} 2 \times \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times Q) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]Q),$$

express the nominal set Q of states as a supp-nondecreasing quotient $e\colon P \twoheadrightarrow Q$ of an orbit-finite strong nominal set P using Remark 5.1. Note that the type functor $F(-) = 2\times\mathcal{P}_{\mathsf{ufs}}(\mathbb{A}\times -)\times\mathcal{P}_{\mathsf{ufs}}([\mathbb{A}](-))$ preserves supp-nondecreasing quotients because all the functors $\mathcal{P}_{\mathsf{ufs}}(-), \mathbb{A}\times -$ and $[\mathbb{A}](-)$ do. Therefore, the right vertical arrow in the square below is supp-nondecreasing, so projectivity of P yields an equivariant map β making it commute:

$$\begin{tikzcd}
P \arrow[r, dashed, "\beta"] \arrow[d, two heads, "e"'] & 2 \times \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times P) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]P) \arrow[d, two heads, "2\times\mathcal{P}_{\mathsf{ufs}}(\mathbb{A}\times e)\times\mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]e)"] \\
Q \arrow[r, "\gamma"] & 2 \times \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times Q) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]Q)
\end{tikzcd}$$

Thus e is a coalgebra homomorphism from (P,β) to (Q,γ). It is not difficult to verify that for every $p \in P$ the states p and $e(p)$ accept the same literal ω-language. In particular, equipping P with an initial state $p_0 \in P$ such that $e(p_0) = q_0$ we see that the Büchi RNNAs P and Q accept the same literal ω-language. ◀

A Büchi RNNA with states $Q = \sum_{i\in I} \mathbb{A}^{\# X_i}$ can be interpreted as an automaton with a finite set I of control states each of which comes equipped with an X_i-indexed set of registers. The following construction shows how to modify such an automaton, preserving the accepted bar ω-language, to become lossy in the sense that after each transition some of the register contents may be nondeterministically erased. Technically, this involves replacing Q with $\widetilde{Q} = \sum_{i\in I} \mathbb{A}^{\$ X_i}$ where $\mathbb{A}^{\$ X_i}$ is a the nominal set of *partial* injective maps from X_i to $\mathbb{A}$, again with the pointwise group action. (Note that while $\mathbb{A}^{\# X_i}$ has only one orbit, $\mathbb{A}^{\$ X_i}$ has one orbit for every subset of X_i.) We represent elements of $\widetilde{Q}$ as pairs (i, r) where $i \in I$ and $r\colon X_i \to \mathbb{A}$ is a partial injective map. The least support of (i, r) is given by

$$\mathsf{supp}(i, r) = \mathsf{supp}(r) = \{r(x) : x \in \mathsf{dom}(r)\},$$

where $\mathsf{dom}(r)$ is the *domain* of r, i.e. the set of all $x \in X_i$ for which $r(x)$ is defined. We say that a partial map $\overline{r}\colon X_i \to \mathbb{A}$ *extends* r if $\mathsf{dom}(r) \subseteq \mathsf{dom}(\overline{r})$ and $r(x) = \overline{r}(x)$ for all $x \in \mathsf{dom}(r)$.

▶ **Definition 5.3** (Name-dropping modification). Let A be a Büchi RNNA whose states form a strong nominal set $Q = \sum_{i\in I} \mathbb{A}^{\# X_i}$ (with I and all X_i finite). The *name-dropping modification* of A is the Büchi RNNA $\widetilde{A}$ defined as follows:

1. The states are given by $\widetilde{Q} = \sum_{i\in I} \mathbb{A}^{\$X_i}$.
2. The initial state of $\widetilde{A}$ is equal to the initial state of A.
3. (i, r) is final in $\widetilde{A}$ iff some (equivalently, every[1]) state $(i, -)$ in A is final.
4. $(i, r) \xrightarrow{a} (j, s)$ in $\widetilde{A}$ iff $\mathsf{supp}(s) \cup \{a\} \subseteq \mathsf{supp}(r)$ and $(i, \overline{r}) \xrightarrow{a} (j, \overline{s})$ in A for some $\overline{r}, \overline{s}$ extending r, s.
5. $(i, r) \xrightarrow{|a} (j, s)$ in $\widetilde{A}$ iff $\mathsf{supp}(s) \subseteq \mathsf{supp}(r) \cup \{a\}$ and $(i, \overline{r}) \xrightarrow{|b} (j, \overline{s})$ in A for some $b \# s$ and some $\overline{r}, \overline{s}$ extending $r, (a\, b)s$.

▶ **Theorem 5.4.** *For every Büchi RNNA A, the literal ω-language of $\widetilde{A}$ is the closure of that of A under α-equivalence.*

Proof sketch.
1. One first verifies that the literal ω-language of $\widetilde{A}$ is closed under α-equivalence. To this end, one proves more generally that given α-equivalent infinite bar strings $w = \sigma_1\sigma_2\sigma_3 \cdots$ and $w' = \sigma_1'\sigma_2'\sigma_3' \cdots$ and a run

$$(i_0, r_0) \xrightarrow{\sigma_1} (i_1, r_1) \xrightarrow{\sigma_2} (i_2, r_2) \xrightarrow{\sigma_3} \cdots$$

for w in $\widetilde{A}$ there exists a run of the form

$$(i_0, r_0') \xrightarrow{\sigma_1'} (i_1, r_1') \xrightarrow{\sigma_2'} (i_2, r_2') \xrightarrow{\sigma_3'} \cdots .$$

for w' in $\widetilde{A}$. In particular, by definition of the final states of $\widetilde{A}$, the first run is accepting iff the second one is. Similar to the proof of Proposition 4.7, one first establishes the corresponding statement for finite runs by induction on their length, and then extends to the infinite case via Kőnig's lemma.
2. In view of part 1 it remains to prove that A and $\widetilde{A}$ accept the same bar ω-language. Again, this follows from a more general observation: for every infinite run

$$(i_0, r_0) \xrightarrow{\sigma_1} (i_1, r_1) \xrightarrow{\sigma_2} (i_2, r_2) \xrightarrow{\sigma_3} \cdots$$

in A there exists an infinite run of the form

$$(i_0, r_0') \xrightarrow{\sigma_1'} (i_1, r_1') \xrightarrow{\sigma_2'} (i_2, r_2') \xrightarrow{\sigma_3'} \cdots$$

in $\widetilde{A}$ such that $\sigma_1'\sigma_2'\sigma_3' \cdots =_\alpha \sigma_1\sigma_2\sigma_3 \cdots$. Moreover, the symmetric statement holds where the roles of A and $\widetilde{A}$ are swapped. As before, one first shows the corresponding statement for finite bar strings and then invokes Kőnig's lemma. ◀

6 Decidability of Inclusion

With the preparations given in the previous sections, we arrive at our main result: language inclusion of Büchi RNNA is decidable, both under bar language semantics and data language semantics. The main ingredients of our proofs below are the name-dropping modification (Theorem 5.4) and the name restriction property stated in Proposition 4.7. We will employ them to show that the language inclusion problems for Büchi RNNA reduce to the inclusion problem for ordinary Büchi automata over finite alphabets. The latter is well-known to be decidable with elementary complexity; in fact, it is PSPACE-complete [18].

[1] This is due to the equivariance of the subset $F \subseteq Q$ of final states.

▶ **Remark 6.1.** For algorithms deciding properties of Büchi RNNAs, a finite representation of the underlying nominal sets of states and transitions is required. A standard representation of a single orbit X is given by picking an arbitrary element $x \in X$ with $\mathsf{supp}(x) = \{a_1, \dots, a_m\}$ and forming the subgroup $G_X \subseteq \mathsf{Perm}(\{a_1, \dots, a_m\})$ of all permutations $\pi\colon \{a_1, \dots, a_m\} \to \{a_1, \dots, a_m\}$ such that $\pi \cdot x = x$. The finite group G_X determines X up to isomorphism [25, Thm. 5.13]. More generally, an orbit-finite nominal set consisting of the orbits $X_1, \dots, X_n$ is represented by a list of n finite permutation groups $G_{X_1}, \dots, G_{X_n}$.

We first consider the bar language semantics:

▶ **Theorem 6.2.** *Bar ω-language inclusion of Büchi RNNAs is decidable in parametrized polynomial space.*

Proof. Let A and B be Büchi RNNAs; the task is to decide whether $L_{\alpha,\omega}(A) \subseteq L_{\alpha,\omega}(B)$. Put $m = \mathsf{deg}(A)$, and choose a set $S \subseteq \mathbb{A}$ of $m+1$ distinct names containing $\mathsf{supp}(q_0)$, where q_0 is the initial state of A. Moreover, form the finite alphabet $\overline{S} = S \cup \{|s : s \in S\}$.

1. We claim that

$$L_{\alpha,\omega}(A) \subseteq L_{\alpha,\omega}(B) \qquad \text{iff} \qquad L_{0,\omega}(A) \cap \overline{S}^{\omega} \subseteq L_{0,\omega}(\widetilde{B}) \cap \overline{S}^{\omega}, \tag{6.1}$$

where $\widetilde{B}$ is the name-dropping modification of B.

($\Rightarrow$) Suppose that $L_{\alpha,w}(A) \subseteq L_{\alpha,\omega}(B)$. Then $L_{0,\omega}(A) \subseteq L_{0,\omega}(\widetilde{B})$ because $L_{0,\omega}(\widetilde{B})$ is the closure of $L_{0,\omega}(B)$ under α-equivalence by Theorem 5.4. In particular, $L_{0,\omega}(A) \cap \overline{S}^{\omega} \subseteq L_{0,\omega}(\widetilde{B}) \cap \overline{S}^{\omega}$.

($\Leftarrow$) Suppose that $L_{0,\omega}(A) \cap \overline{S}^{\omega} \subseteq L_{0,\omega}(\widetilde{B}) \cap \overline{S}^{\omega}$, and let $[w]_\alpha \in L_{\alpha,\omega}(A)$. Thus, $w =_\alpha v$ for some $v \in L_{0,\omega}(A)$. By Proposition 4.7 we know that there exists $v' \in \overline{S}^{\omega}$ accepted by A such that $v' =_\alpha v$. Then

$$v' \in L_{0,\omega}(A) \cap \overline{S}^{\omega} \subseteq L_{0,\omega}(\widetilde{B}) \cap \overline{S}^{\omega} \subseteq L_{0,\omega}(\widetilde{B}).$$

Thus $[w]_\alpha = [v]_\alpha = [v']_\alpha \in L_{\alpha,\omega}(\widetilde{B}) = L_{\alpha,\omega}(B)$, proving $L_{\alpha,\omega}(A) \subseteq L_{\alpha,\omega}(B)$.

2. Now observe that both $L_{0,\omega}(A) \cap \overline{S}^{\omega}$ and $L_{0,\omega}(\widetilde{B}) \cap \overline{S}^{\omega}$ are ω-regular languages over the alphabet $\overline{S}$: they are accepted by the Büchi automata obtained by restricting the Büchi RNNAs A and B, respectively, to the finite set of states with support contained in S and transitions labeled by elements of $\overline{S}$. Inclusion of ω-regular languages is decidable in polynomial space [18]. Let k_A/k_B and m_A/m_B denote the number of orbits and the degree of A/B. Since the support of every state of the Büchi automaton derived from A is contained in the set S and the latter has $(m_A+1)!$ permutations, there are at most

$$k_A \cdot (m_A+1)! \in \mathcal{O}(k_A \cdot 2^{(m_A+1)\log(m_A+1)})$$

states. The Büchi automaton derived from $\widetilde{B}$ has at most

$$k_B \cdot 2^{m_B} \cdot (m_A+1)! \in \mathcal{O}(k_B \cdot 2^{m_B+(m_A+1)\log(m_A+1)})$$

states since $\widetilde{B}$ has at most $k_B \cdot 2^{m_B}$ orbits. Thus, the space required for the inclusion check is polynomial in k_A, k_B and exponential in $m_B + (m_A+1)\log(m_A+1)$. ◀

Next, we turn to the data language semantics:

▶ **Notation 6.3.** Given infinite bar strings $w = \sigma_1\sigma_2\sigma_3\cdots$ and $w' = \sigma_1'\sigma_2'\sigma_3'\cdots$ we write $w \sqsubseteq w'$ if, for all $a \in \mathbb{A}$ and $n \in \mathbb{N}$,

$$\sigma_n = a \quad \text{implies} \quad \sigma_n' \in \{a, |a\} \qquad \text{and} \qquad \sigma_n = |a \quad \text{implies} \quad \sigma_n' = |a.$$

Thus, w' arises from w by arbitrarily putting bars in front of letters in w.

▶ **Lemma 6.4.** *If $v =_\alpha w$ and $v \sqsubseteq v'$, then there exists $w' \in \overline{\mathbb{A}}^\omega$ such that $w \sqsubseteq w'$ and $v' =_\alpha w'$.*

Proof. Let $v' = \sigma'_1\sigma'_2\sigma'_3\cdots$ and $w = \rho_1\rho_2\rho_3\cdots$. We define w' to be the following modification of w: for every $n \in \mathbb{N}$, if $\rho_n = a$ and $\sigma'_n = |b$ for some $a, b \in \mathbb{A}$, replace ρ_n by $|a$. Then $w \sqsubseteq w'$ and $v' =_\alpha w'$ as required. ◀

In the following, we write $D(w)$ for $D(\{[w]_\alpha\})$; thus, $D(w)$ is the set of all $\mathsf{ub}(v) \in \mathbb{A}^\omega$ where $v =_\alpha w$ (cf. Notation 3.5).

▶ **Lemma 6.5.** *Let L be a bar ω-language accepted by some Büchi RNNA and let $w \in \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$. Then $D(w) \subseteq D(L)$ if and only if there exists $w' \sqsupseteq w$ such that $[w']_\alpha \in L$.*

▶ **Corollary 6.6.** *Let K and L be bar ω-languages accepted by Büchi RNNA. Then $D(K) \subseteq D(L)$ if and only if for every $w \in \overline{\mathbb{A}}^\omega_{\mathsf{fs}}$ with $[w]_\alpha \in K$ there exists $w' \sqsupseteq w$ such that $[w']_\alpha \in L$.*

Proof. This follows from Lemma 6.5 using that $D(K) = \bigcup_{[w]_\alpha \in K} D(w)$ and that every α-equivalence class $[w]_\alpha \in K$ has a finitely supported representative by Proposition 4.7. ◀

▶ **Theorem 6.7.** *Data ω-language inclusion of Büchi RNNAs is decidable in parametrized polynomial space.*

Proof. Let A and B be Büchi RNNAs; the task is to decide whether $D_\omega(A) \subseteq D_\omega(B)$. Put $m = \mathsf{deg}(A)$, and choose a set $S \subseteq \mathbb{A}$ of $m+1$ distinct names containing $\mathsf{supp}(q_0)$, where q_0 is the initial state of A. Moreover, form the finite alphabet $\overline{S} = S \cup \{|s : s \in S\}$.

1. For every language $L \subseteq \overline{S}^\omega$ let $L{\downarrow}$ denote the downward closure of L with respect to $\sqsubseteq$:

   $$L{\downarrow} = \{v \in \overline{S}^\omega : \text{there exists } w \in L \text{ such that } v \sqsubseteq w\}.$$

   Every Büchi automaton accepting L can be turned into a Büchi automaton accepting $L{\downarrow}$ by adding the transition $q \xrightarrow{a} q'$ for every transition $q \xrightarrow{|a} q'$ where $a \in S$.
2. We claim that

   $$D_\omega(A) \subseteq D_\omega(B) \qquad \text{iff} \qquad L_{0,\omega}(A) \cap \overline{S}^\omega \subseteq (L_{0,\omega}(\widetilde{B}) \cap \overline{S}^\omega){\downarrow}, \tag{6.2}$$

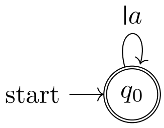
   where $\widetilde{B}$ is the name-dropping modification of B.

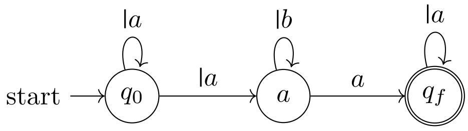
   ($\Rightarrow$) Suppose that $D_\omega(A) \subseteq D_\omega(B)$, that is $D(L_{\alpha,w}(A)) \subseteq D(L_{\alpha,\omega}(B))$, and let $w \in L_{0,\omega}(A) \cap \overline{S}^\omega$. Then $[w]_\alpha \in L_{\alpha,\omega}(A)$, hence by Corollary 6.6 there exists $w' \sqsupseteq w$ such that $[w']_\alpha \in L_{\alpha,\omega}(B)$. Then $w' \in L_{0,\omega}(\widetilde{B}) \cap \overline{S}^\omega$ because $L_{0,\omega}(\widetilde{B})$ is the closure of $L_{0,\omega}(B)$ under α-equivalence by Theorem 5.4. Thus $w \in (L_{0,\omega}(\widetilde{B}) \cap \overline{S}^\omega){\downarrow}$, which proves $L_{0,\omega}(A) \cap \overline{S}^\omega \subseteq (L_{0,\omega}(\widetilde{B}) \cap \overline{S}^\omega){\downarrow}$.

   ($\Leftarrow$) Suppose that $L_{0,\omega}(A) \cap \overline{S}^\omega \subseteq (L_{0,\omega}(\widetilde{B}) \cap \overline{S}^\omega){\downarrow}$. To prove $D_\omega(A) \subseteq D_\omega(B)$, that is $D(L_{\alpha,\omega}(A)) \subseteq D(L_{\alpha,\omega}(B))$, we use Corollary 6.6. Thus, let $[w]_\alpha \in L_{\alpha,\omega}(A)$. Then $w =_\alpha v$ for some $v \in L_{0,\omega}(A)$. By Proposition 4.7 we may assume that $v \in \overline{S}^\omega$. By hypothesis this implies $v \in (L_{0,\omega}(\widetilde{B}) \cap \overline{S}^\omega){\downarrow}$, that is, there exists $v' \sqsupseteq v$ such that $v' \in L_{0,\omega}(\widetilde{B})$. By Lemma 6.4, there exists $w' \sqsupseteq w$ such that $w' =_\alpha v'$. Then $[w']_\alpha = [v']_\alpha \in L_{\alpha,\omega}(\widetilde{B}) = L_{\alpha,\omega}(B)$, as required.
3. The decidability of $D_\omega(A) \subseteq D_\omega(B)$ now follows from part 1 and (6.2) in complete analogy to the proof of Theorem 6.2. ◀

▶ **Remark 6.8.** In the above theorem it is crucial to admit non-finitely supported data words: it is an open problem whether the inclusion $D_\omega(A) \cap \mathbb{A}^\omega_{\mathsf{fs}} \subseteq D_\omega(B) \cap \mathbb{A}^\omega_{\mathsf{fs}}$ is decidable. In fact, our decidability proof relies on Corollary 6.6 as a key ingredient, and the latter fails if the condition $D(K) \subseteq D(L)$ is replaced by the weaker condition $D(K) \cap \mathbb{A}^\omega_{\mathsf{fs}} \subseteq D(L) \cap \mathbb{A}^\omega_{\mathsf{fs}}$.

To see this, let K and L be the bar ω-languages accepted by the two Büchi RNNAs displayed below, where a, b range over names in $\mathbb{A}$ and $a \neq b$:





We have $D(K) = \mathbb{A}^\omega$ and $D(L)$ consists of all data words in $\mathbb{A}^\omega$ in which some name occurs at least twice. Thus, $D(K) \cap \mathbb{A}^\omega_{\mathsf{fs}} = D(L) \cap \mathbb{A}^\omega_{\mathsf{fs}} = \mathbb{A}^\omega_{\mathsf{fs}}$, in particular the inclusion $D(K) \cap \mathbb{A}^\omega_{\mathsf{fs}} \subseteq D(L) \cap \mathbb{A}^\omega_{\mathsf{fs}}$ holds. Consider the infinite bar string $w = {|}a{|}a{|}a \cdots$ and note that $w' \sqsupseteq w$ implies $w' = w$. Then $[w]_\alpha \in K$ but $[w]_\alpha \notin L$ since every bar string accepted by the right-hand automaton contains some letter from $\mathbb{A}$. Thus, "only if" fails in Corollary 6.6.

7 Relation to Other Automata Models

We conclude this paper by comparing Büchi RNNA with two related automata models over infinite words. Ciancia and Sammartino [5] consider deterministic nominal automata accepting data ω-languages $L \subseteq \mathbb{A}^\omega$ via a Muller acceptance condition. More precisely, a *nominal deterministic Muller automaton* (*nDMA*) $A = (Q, \delta, q_0, \mathcal{F})$ is given by an orbit-finite nominal set Q of states, and equivariant map $\delta \colon Q \times \mathbb{A} \to Q$ representing transitions, an initial state $q_0 \in Q$, and a set $\mathcal{F} \subseteq \mathcal{P}(\mathsf{orb}(Q))$ where $\mathsf{orb}(Q)$ is the finite set of orbits of Q. Every input word $w = a_1 a_2 a_3 \cdots \in \mathbb{A}^\omega$ has a unique run $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots$ where $q_{i+1} = \delta(q_i, a_{i+1})$ for $i = 0, 1, 2, \ldots$. The word w is *accepted* if the set $\{O \in \mathsf{orb}(Q) : q_n \in O$ for infinitely many $n\}$ lies in $\mathcal{F}$. The data ω-language *accepted* by the automaton is the set of all words $w \in \mathbb{A}^\omega$ whose run is accepting.

As for Büchi RNNA, language inclusion is decidable for nDMA [5, Thm. 4]. In terms of expressive power the two models are incomparable, as witnessed by the data ω-languages

$$K = \{w \in \mathbb{A}^\omega : \text{some } a \in \mathbb{A} \text{ occurs infinitely often in } w\},$$
$$L = \{w \in \mathbb{A}^\omega : w \text{ does not have the suffix } a^\omega \text{ for any } a \in \mathbb{A}\}.$$

▶ **Proposition 7.1.**
1. *The language K is accepted by a Büchi RNNA but not by any nDMA.*
2. *The language L is accepted by an nDMA but not by any Büchi RNNA.*

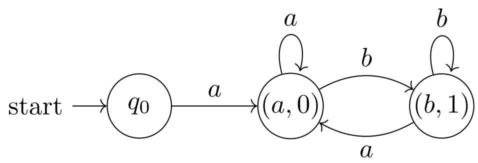
Proof.
1. We have seen in Example 4.4 that the language K is accepted by a Büchi RNNA. We claim that K is not accepted by any nDMA. Since the class of languages accepted by nDMA is closed under complement, it suffices to show that the language

$$\overline{K} = \{w \in \mathbb{A}^\omega : \text{each } a \in \mathbb{A} \text{ occurs only finitely often in } w\}$$

is not accepted by any nDMA. Suppose towards a contradiction that $A = (Q, \delta, q_0, \mathcal{F})$ is an nDMA accepting $\overline{K}$. Let m be the maximum of all $|\mathsf{supp}(q)|$ where $q \in Q$. Fix $m+1$ pairwise distinct names $a_1, \ldots, a_{m+1} \in \mathbb{A}$ and an arbitrary word $w_0 \in \overline{K}$.
Choose a factorization $w_0 = v_1 w_1$ ($v_1 \in \mathbb{A}^*, w_1 \in \mathbb{A}^\omega$) such that all occurrences of $a_1, \ldots, a_{m+1}$ in w_0 lie in the finite prefix v_1. Let q_1 be the state reached from q_0 on input v_1. Then $\mathsf{supp}(q_1)$ does not contain all of the names $a_1, \ldots, a_{m+1}$, say $a_i \mathrel{\#} q_1$. Choose any name $a \in \mathbb{A}$ occurring in w_1 such that $a \# q_1$. Then $q_1 = (a\, a_i) q_1$ accepts $(a\, a_i) w_1$ since by equivariance the run of $(a\, a_i) w_1$ from q_1 visits the same orbits as the run of w_1.
Now repeat the same process with q_1 and $(a\, a_i) w_1$ in lieu of q_0 and w_0, and let $(a\, a_i) w_1 = v_2 w_2$ denote the corresponding factorization; note that v_2 is nonempty because $(a\, a_i) w_1$ contains the name a_i. Continuing in this fashion yields an infinite word

$v = v_1v_2v_3\ldots$ such that each v_i $(i > 1)$ contains at least one of the letters $a_1, \ldots, a_{m+1}$, and the run of v traverses the same orbits (in the same order) as the run of w. Thus v is accepted by the nDMA A although $v \notin \overline{K}$, a contradiction.

2. The language L is accepted by the nDMA with states $\{q_0\} \cup \mathbb{A} \times \{0, 1\}$ and transitions as displayed below, where a, b range over distinct names in $\mathbb{A}$. The acceptance condition is given by $\mathcal{F} = \{\{\mathbb{A} \times \{0\}, \mathbb{A} \times \{1\}\}\}$.



We claim that L is not accepted by any Büchi RNNA. Suppose to the contrary that $A = (Q, R, q_0, F)$ is a Büchi RNNA with $D_\omega(A) = L$. By Theorem 5.4 we may assume that $L_{0,\omega}(A)$ is closed under α-equivalence. Fix an arbitrary word $w = a_1a_2a_3\cdots$ whose names are pairwise distinct and not contained in $\mathsf{supp}(q_0)$. Then $w \in L$, so there exists $v \in L_{0,\omega}(A)$ such that $\mathsf{ub}(v) = w$. We claim that $v = |a_1|a_2|a_3\cdots$. Indeed, if the nth letter of v is a_n, then in an accepting run for v in A the state q reached before reading a_n must have $a_n \in \mathsf{supp}(q)$ by Lemma 4.6.3. But this is impossible because $\mathsf{supp}(q) \subseteq \mathsf{supp}(q_0) \cup \{a_1, \ldots, a_{n-1}\}$ again by Lemma 4.6.

Thus $v =_\alpha |a|a|a\cdots$. This implies $a^\omega \in D_\omega(A)$ although $a^\omega \notin L$, a contradiction. ◀

Let us note that the above result does not originate in weakness of the Büchi acceptance condition. One may generalize Büchi RNNA to *Muller RNNA* where the final states $F \subseteq Q$ are replaced by a set $\mathcal{F} \subseteq \mathcal{P}(\mathsf{orb}(Q))$, and a bar string $w \in \overline{\mathbb{A}}^\omega$ is said to be *accepted* if there exists a run for w such that the set of orbits visited infinitely often lies in $\mathcal{F}$. However, as for classical nondeterministic Büchi and Muller automata, this does not affect expressivity:

▶ **Proposition 7.2.** *A literal ω-language is accepted by a Büchi RNNA if and only if it is accepted by a Muller RNNA.*

Finally, we mention a tight connection between Büchi RNNA and *session automata* [4]. The data (ω-)language associated to a (Büchi) RNNA uses a *local freshness* semantics in the sense that its definition considers possibly non-clean bar strings accepted by A. In some applications, e.g. nonce generation, a more suitable semantics is given by *global freshness* where only clean bar strings are admitted, i.e. one associates to A the data languages

$$D_\#(A) = \{\mathsf{ub}(w) : w \in \overline{\mathbb{A}}^* \text{ clean and } [w]_\alpha \in L_\alpha(A)\},$$
$$D_{\omega,\#}(A) = \{\mathsf{ub}(w) : w \in \overline{\mathbb{A}}^\omega \text{ clean and } [w]_\alpha \in L_{\alpha,\omega}(A)\}.$$

For instance, for the Büchi RNNA A from Example 4.4 the language $D_{\omega,\#}(A)$ consists of all infinite words $w \in \mathbb{A}^\omega$ where exactly one name $a \in \mathbb{A}$ occurs infinitely often and every name $b \neq a$ occurs at most once.

Under global freshness semantics, it has been observed in previous work [26] that a data language $L \subseteq \mathbb{A}^*$ is accepted by some session automaton iff $L = D_\#(A)$ for some RNNA A whose initial state q_0 has empty support. An analogous correspondence holds for Büchi RNNA and data ω-languages $L \subseteq \mathbb{A}^\omega$ if the original notion of session automata is generalized to infinite words with a Büchi acceptance condition.

8 Conclusions and Future Work

We have introduced *Büchi regular nondeterministic nominal automata* (Büchi RNNAs), an automaton model for languages of infinite words over infinite alphabets. Büchi RNNAs allow for inclusion checking in elementary complexity (parametrized polynomial space) despite the fact that they feature full nondeterminism and do not restrict the number of registers (contrastingly, even for register automata over finite words [16], inclusion checking becomes decidable only if the number of registers is bounded to at most 2).

An important further step will be to establish a logic-automata correspondence of Büchi RNNAs with a suitable form of linear temporal logic on infinite data words.

A natural direction for generalization is to investigate RNNAs over infinite trees with binders, modeled as coalgebras of type $\mathcal{P}_{\mathsf{ufs}}F$ for a functor F associated to a binding signature and equipped with the ensuing notion of α-equivalence due to Kurz et al. [20].

Finally, we would like to explore the bar language and data language semantics of Büchi RNNA from the perspective of coalgebraic trace semantics [29], where infinite behaviours emerge as solutions of (nested) fixed point equations in Kleisli categories.

References

1 Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1985.

2 Mikołaj Bojańczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data words. *ACM Trans. Comput. Log.*, 12(4):27:1–27:26, 2011. doi:10.1145/1970398.1970403.

3 Mikołaj Bojańczyk, Bartek Klin, and Sławomir Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10(3), 2014. doi:10.2168/LMCS-10(3:4)2014.

4 Benedikt Bollig, Peter Habermehl, Martin Leucker, and Benjamin Monmege. A robust class of data languages and an application to learning. *Log. Meth. Comput. Sci.*, 10(4), 2014. doi:10.2168/LMCS-10(4:19)2014.

5 Vincenzo Ciancia and Matteo Sammartino. A class of automata for the verification of infinite, resource-allocating behaviours. In *Trustworthy Global Computing, TGC 2014*, volume 8902 of *LNCS*, pages 97–111. Springer, 2014. doi:10.1007/978-3-662-45917-1_7.

6 Thomas Colcombet and Amaldev Manuel. Generalized data automata and fixpoint logic. In *Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014*, volume 29 of *LIPIcs*, pages 267–278. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014. doi:10.4230/LIPIcs.FSTTCS.2014.267.

7 Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary. *J. ACM*, 68(1):7:1–7:28, 2021. doi:10.1145/3422822.

8 Wojciech Czerwiński and Lukasz Orlikowski. Reachability in vector addition systems is Ackermann-complete. *CoRR*, 2021. arXiv:2104.13866.

9 Stéphane Demri and Ranko Lazić. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Log.*, 10(3):16:1–16:30, 2009. doi:10.1145/1507244.1507246.

10 Murdoch J. Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bull. Symb. Log.*, 17(2):161–229, 2011. doi:10.2178/bsl/1305810911.

11 Murdoch J. Gabbay and Vincenzo Ciancia. Freshness and name-restriction in sets of traces with names. In *Foundations of Software Science and Computation Structures, FOSSACS 2011*, volume 6604 of *LNCS*, pages 365–380. Springer, 2011. doi:10.1007/978-3-642-19805-2.

12 Murdoch J. Gabbay and Andrew M. Pitts. A new approach to abstract syntax involving binders. In *Logic in Computer Science, LICS 1999*, pages 214–224. IEEE Computer Society, 1999. doi:10.1109/LICS.1999.782617.

13 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002. doi:10.1007/3-540-36387-4.

14 Radu Grigore, Dino Distefano, Rasmus Petersen, and Nikos Tzevelekos. Runtime verification based on register automata. In *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2013*, volume 7795 of *LNCS*, pages 260–276. Springer, 2013. doi:10.1007/978-3-642-36742-7_19.

15 Orna Grumberg, Orna Kupferman, and Sarai Sheinvald. Variable automata over infinite alphabets. In *Language and Automata Theory and Applications, LATA 2010*, volume 6031 of *LNCS*, pages 561–572. Springer, 2010. doi:10.1007/978-3-642-13089-2.

16 Michael Kaminski and Nissim Francez. Finite-memory automata. *Theor. Comput. Sci.*, 134(2):329–363, 1994. doi:10.1016/0304-3975(94)90242-9.

17 Ahmet Kara, Thomas Schwentick, and Tony Tan. Feasible automata for two-variable logic with successor on data words. In *Language and Automata Theory and Applications, LATA 2012*, volume 7183 of *LNCS*, pages 351–362. Springer, 2012. doi:10.1007/978-3-642-28332-1_30.

18 Orna Kupferman and Moshe Y. Vardi. Verification of fair transisiton systems. In *Computer Aided Verification, CAV 1996*, volume 1102 of *LNCS*, pages 372–382. Springer, 1996. doi:10.1007/3-540-61474-5_84.

19 Klaas Kürtz, Ralf Küsters, and Thomas Wilke. Selecting theories and nonce generation for recursive protocols. In *Formal methods in security engineering, FMSE 2007*, pages 61–70. ACM, 2007. doi:10.1145/1314436.1314445.

20 Alexander Kurz, Daniela Petrisan, Paula Severi, and Fer-Jan de Vries. Nominal coalgebraic data types with applications to lambda calculus. *Log. Methods Comput. Sci.*, 9(4), 2013. doi:10.2168/LMCS-9(4:20)2013.

21 Ranko Lazić. Safely freezing LTL. In *Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2006*, volume 4337 of *LNCS*, pages 381–392. Springer, 2006. doi:10.1007/11944836_35.

22 Jérôme Leroux. The reachability problem for Petri nets is not primitive recursive. *CoRR*, 2021. arXiv:2104.12695.

23 Stefan Milius and Henning Urbat. Equational axiomatization of algebras with structure. In *Foundations of Software Science and Computation Structures, FOSSACS 2019*, volume 11425 of *LNCS*, pages 400–417. Springer, 2019. Full version available at arXiv:1812.02016. doi:10.1007/978-3-030-17127-8_23.

24 Daniela Petrişan. *Investigations into Algebra and Topology over Nominal Sets*. PhD thesis, University of Leicester, 2011.

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

26 Lutz Schröder, Dexter Kozen, Stefan Milius, and Thorsten Wißmann. Nominal automata with name binding. In *Foundations of Software Science and Computation Structures, FOSSACS 2017*, volume 10203 of *LNCS*, pages 124–142, 2017. doi:10.1007/978-3-662-54458-7_8.

27 Christoph Stockhusen and Till Tantau. Completeness results for parameterized space classes. In *Parameterized and Exact Computation, IPEC 2013*, volume 8246 of *LNCS*, pages 335–347. Springer, 2013. doi:10.1007/978-3-319-03898-8.

28 Nikos Tzevelekos. *Nominal Game Semantics*. PhD thesis, University of Oxford, 2008.

29 Natsuki Urabe, Shunsuke Shimizu, and Ichiro Hasuo. Coalgebraic trace semantics for buechi and parity automata. In *Concurrency Theory, CONCUR 2016*, volume 59 of *LIPIcs*, pages 24:1–24:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CONCUR.2016.24.