Rule Formats for Nominal Process Calculi*

Luca Aceto[1], Ignacio Fábregas[1,2], Álvaro García-Pérez[1,3], Anna Ingólfsdóttir[1], and Yolanda Ortega-Mallén[2]

1 **ICE-TCS, School of Computer Science, Reykjavik University (Iceland)**
{luca,annai}@ru.is
2 **Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid (Spain)**
{fabregas,yolanda}@ucm.es
3 **IMDEA Software Institute, Madrid (Spain)**
alvaro.garcia.perez@imdea.org

Abstract

The nominal transition systems (NTS) of Parrow et al. describe the operational semantics of nominal process calculi. We study NTSs in terms of the nominal residual transition systems (NRTS) that we introduce. We provide rule formats for the specifications of NRTSs that ensure that the associated NRTS is an NTS and apply them to the operational specification of the early π-calculus. Our study stems from the recent Nominal SOS of Cimini et al. and from earlier works in nominal sets and nominal logic by Gabbay, Pitts and their collaborators.

1998 ACM Subject Classification D.3.1 Formal Definitions and Theory, F.1.1 Models of Computation, F.1.2 Modes of Computation, F.3.1 Specifying and Verifying and Reasoning about Programs, F.3.2 Semantics of Programming Languages,

Keywords and phrases nominal sets, nominal structural operational semantics, process algebra, nominal transition systems, scope opening, rule formats

1 Introduction

The goal of this paper is to develop the foundations of a framework for studying the meta-theory of structural operational semantics (SOS) [27] for process calculi with names and name-binding operations, such as the π-calculi [29]. To this end, we build on the large body of work on rule formats for SOS, as surveyed in [2, 23], and on the nominal techniques of Gabbay, Pitts and their co-workers [8, 15, 25, 30].

Rule formats provide syntactic templates guaranteeing that the models of the calculi, whose semantics they specify, enjoy some desirable properties. A first design decision that has to be taken in developing a theory of rule formats for a class of languages is therefore the choice of the semantic objects specified by the rules. The target semantic model we adopt in our study is that of *nominal transition systems* (NTSs), which have been introduced by Parrow *et al.* in [24] as a uniform model to describe the operational semantics of a variety of calculi with names and name-binding operations. Based on this choice, a basic sanity criterion for a collection of rules describing the operational semantics of a nominal

* Research partially supported by the project Nominal SOS (nr. 141558-051) of the Icelandic Research Fund, the project 001-ABEL-CM-2013 within the NILS Science and Sustainability Programme, the Spanish projects N-Greens Software (S2013/ICE-2731), TRACES (TIN2015-67522-C3-3-R) and StrongSoft (TIN2012-39391-C04), and the project RACCOON (H2020-EU 714729) of the European Research Council.

© L. Aceto and I. Fábregas and A. García-Pérez and A. Ingólfsdóttir and Y. Ortega-Mallén;
licensed under Creative Commons License CC-BY

calculus is that they specify an NTS, and we present a rule format guaranteeing this property (Thm. 5.9).

As a first stepping stone in our study, we introduce *nominal residual transition systems* (NRTSs), and study NTSs in terms of NRTSs (Section 2). More specifically, NRTSs enjoy one desirable property in the setting of nominal calculi, namely that their transition relation is equivariant (which means that it treats names uniformly). NTSs are NRTSs that, in addition to having an equivariant transition relation, satisfy a property Parrow *et al.* call *alpha-conversion of residuals* (see Def. 2.3 for the details). The latter property formalises a key aspect of calculi in which names can be scoped to represent local resources. To wit, one crucial feature of the π-calculus is scope opening [22]. Consider a transition $p \xrightarrow{\overline{a}(\nu b)} p'$ in which a process p exports a private/local channel name b along channel a. Since the name b is local, it 'can be subject to alpha-conversion' [24] and the transitions $p \xrightarrow{\overline{a}(\nu c)} p\{b/c\}$ should also be present for each 'fresh name' c.

We specify an NRTS by means of a nominal residual transition system specification (NRTSS), which describes the syntax of a nominal calculus in terms of a nominal signature (Section 3) and its semantics by means of a set of inference rules (Section 4). We develop the basic theory of the NRTS/NRTSS framework, building on the nominal algebraic datatypes of Pitts [25] and the nominal rewriting framework of Fernandez and Gabbay [9]. Based on this framework, we provide rule formats [2,23] for NRTSSs that ensure that the associated NRTS specifies a transition relation that is equivariant (Thm. 5.3) and enjoys alpha-conversion of residuals (Thm. 5.9), and is therefore an NTS. Section 6 presents an example of application of our rule formats to the setting of the π-calculus, and Section 7 discusses avenues for future work, discusses related work and concludes.

The appendix accompanying the paper collects all the proofs of our results, and provides further definitions and examples.

2 Preliminaries

Nominal sets

We follow earlier foundational work by Gabbay and Pitts on nominal sets in [16, 25, 26]. We assume a countably infinite set $\mathbb{A}$ of *atoms* and consider Perm $\mathbb{A}$ as the group of *finite permutations of atoms* (hereafter *permutations*) ranged over by π, where we write ι for the *identity*, $\circ$ for *composition* and π^{-1} for the *inverse* of permutation π. We are particularly interested in *transpositions* of two atoms: $(a\,b)$ stands for the permutation that swaps a with b and leaves all other atoms fixed. Every permutation π is equal to the composition of a finite number of transpositions, i.e. $\pi = (a_1\,b_1) \circ \ldots \circ (a_n\,b_n)$ with $n \geq 0$.

An *action* of the group Perm $\mathbb{A}$ on a set S is a binary operation mapping each $\pi \in$ Perm $\mathbb{A}$ and $s \in S$ to an element $\pi \cdot s \in S$, and satisfying the identity law $\iota \cdot s = s$ and the composition law $(\pi_1 \circ \pi_2) \cdot s = \pi_1 \cdot (\pi_2 \cdot s)$. A Perm $\mathbb{A}$-*set* is a set equipped with an action of Perm $\mathbb{A}$.

We say that a set of atoms A *supports* an object s iff $\pi \cdot s = s$ for every permutation π that leaves each element $a \in A$ invariant. In particular, we are interested in sets all of whose elements have finite support (Def. 2.2 of [25]).

▶ **Definition 2.1** (Nominal sets). A *nominal set* is a Perm $\mathbb{A}$-set all of whose elements are finitely supported.

For each element s of a nominal set, we write $\mathrm{supp}(s)$ for the least set that supports s, called the *support* of s. (Intuitively, the action of permutations on a set S determines

that a finitely supported $s \in S$ only depends on atoms in $\mathrm{supp}(s)$, and no others.) The set $\mathbb{A}$ of atoms is a nominal set when $\pi \cdot a = \pi a$ since $\mathrm{supp}(a) = \{a\}$ for each atom $a \in \mathbb{A}$. The set Perm $\mathbb{A}$ of finite permutations is also a nominal set where the permutation action on permutations is given by conjugation, i.e. $\pi \cdot \pi' = \pi \circ \pi' \circ \pi^{-1}$, and the support of a permutation π is $\mathrm{supp}(\pi) = \{a \mid \pi a \neq a\}$.

Given two Perm $\mathbb{A}$-sets S and T and a function $f : S \to T$, the action of permutation π on function f is given by conjugation, i.e. $(\pi \cdot f)(s) = \pi \cdot f(\pi^{-1} \cdot s)$ for each $s \in S$. We say that a function $f : S \to T$ is *equivariant* iff $\pi \cdot f(s) = f(\pi \cdot s)$ for every $\pi \in$ Perm $\mathbb{A}$ and every $s \in S$. The intuition is that an equivariant function f is atom-blind, in that f does not treat any atom preferentially. It turns out that a function f is equivariant iff $\mathrm{supp}(f) = \emptyset$ (Rem. 2.13 of [25]). The function supp is equivariant (Prop. 2.11 of [25]).

Let S be a Perm $\mathbb{A}$-set, we write S_{fs} for the nominal set that contains the elements in S that are finitely supported. The *nominal function set* between nominal sets S and T is the nominal set $S \to_{\mathrm{fs}} T$ of finitely supported functions from S to T—be they equivariant or not.

Let S_1 and S_2 be nominal sets. The product $S_1 \times S_2$ is a nominal set (Prop. 2.14 of [25]). An element $s_1 \in S_1$ *is fresh in* $s_2 \in S_2$, written $s_1 \# s_2$, iff $\mathrm{supp}(s_1) \cap \mathrm{supp}(s_2) = \emptyset$. The freshness relation is equivariant (Eq. (3.2) of [25]).

Finally, we consider *atom abstractions* (Sec. 4 of [25]), which represent alpha-equivalence classes of elements.

▶ **Definition 2.2** (Atom abstraction)**.** Given a nominal set S, the *atom abstraction* of atom a in element $s \in S$, written $\langle a \rangle s$, is the set $\langle a \rangle s = \{(b, (b\,a) \cdot s) \mid b = a \vee b \# s\}$, whose permutation action and support are $\pi \cdot \langle a \rangle s = \{(\pi \cdot b, \pi \cdot ((b\,a) \cdot s)) \mid \pi \cdot b = \pi \cdot a \vee \pi \cdot b \# \pi \cdot s\}$, and $\mathrm{supp}(\langle a \rangle s) = \mathrm{supp}(s) \setminus \{a\}$.

We write $[\mathbb{A}]S$ for the set of *atom abstractions* in elements of S, which is a nominal set (Def. 4.4 of [25]).

Nominal Transition Systems

Nominal transitions systems adopt the state/residual presentation for transitions of [4], where a residual is a pair consisting of an action and a state. In [24], Parrow *et al.* develop modal logics for process algebras à la Hennessy-Milner. Here we are mainly interested in the transition relation and we adapt Definition 1 in [24] by removing the predicates. We write $\mathcal{P}_\omega(\mathbb{A})$ for the *finite power set* of $\mathbb{A}$.

▶ **Definition 2.3** (Nominal transition system)**.** A *nominal transition system* (NTS) is a quadruple $(S, Act, \mathrm{bn}, \longrightarrow)$ where S and Act are nominal sets of *states* and *actions* respectively, $\mathrm{bn} : Act \to \mathcal{P}_\omega(\mathbb{A})$ is an equivariant function that delivers the *binding names* in an action, and $\longrightarrow \subseteq S \times (Act \times S)$ is an equivariant binary transition relation from states to *residuals* (we let $Act \times S$ be the set of residuals). The function bn is such that $\mathrm{bn}(\ell) \subseteq \mathrm{supp}(\ell)$ for each $\ell \in Act$. We often write $p \longrightarrow (\ell, p')$ in lieu of $(p, (\ell, p')) \in \longrightarrow$.

Finally, the transition relation $\longrightarrow$ must satisfy *alpha-conversion of residuals*, that is, if $a \in \mathrm{bn}(\ell)$, $b \# (\ell, p')$ and $p \longrightarrow (\ell, p')$ then also $p \longrightarrow ((a\,b) \cdot \ell, (a\,b) \cdot p')$—the permutation action for products of nominal sets is given componentwise (Eq (1.12) of [25]) and from now on we may write $p \longrightarrow (a\,b) \cdot (\ell, p')$.

We will consider an NTS (without its associated binding-names function bn) as a particular case of a nominal residual transition system, which we introduce next.

▶ **Definition 2.4** (Nominal residual transition system)**.** A *nominal residual transition system* (NRTS) is a triple $(S, R, \longrightarrow)$ where S and R are nominal sets, and where $\longrightarrow \subseteq S \times R$ is an equivariant binary transition relation. We say S is the set of *states* and R is the set of *residuals*.

The connection between NTSs and NRTSs will be explained in more detail in Section 5.

3 Nominal terms

This section is devoted to the notion of nominal terms, which are syntactic objects that make use of the atom abstractions of Definition 2.2 and represent terms up to alpha-equivalence. As a first step, we introduce raw terms, which do not represent terms up to alpha-equivalence. Our raw terms resemble those from the literature, mainly [8, 9, 25, 30], but with some important differences. In particular, our terms include both variables (i.e. unknowns) and moderated terms (i.e. explicit permutations over raw terms), and we consider atom and abstraction sorts. (The raw terms of [25] do not include moderated terms, and the ones in [9, 30] only consider moderated variables. In [8] the authors consider neither atom nor abstraction sorts.) We also adopt the classic presentation of free algebras and term algebras in [6, 18] in a different way from that in [8, 25]. The raw terms correspond to the standard notion of free algebra over a signature generated by a set of variables. We then adapt the Σ-structures of [8] to our sorting schema. Finally, the nominal terms are the interpretations of the ground terms in the initial Σ-structure, and we show that they coincide with the nominal algebraic terms of [25].

▶ **Definition 3.1** (Nominal signature and nominal sort)**.** A *nominal signature* (or simply a *signature*) Σ is a triple (Δ, A, F) where $\Delta = \{\delta_1, \ldots, \delta_n\}$ is a finite set of *base sorts*, A is a countable set of *atom sorts*, and F is a finite set of *function symbols*. The *nominal sorts* over Δ and A are given by the grammar $\sigma ::= \delta \mid \alpha \mid [\alpha]\sigma \mid \sigma_1 \times \ldots \times \sigma_k$, with $k \geq 0$, $\delta \in \Delta$ and $\alpha \in A$. The sort $[\alpha]\sigma$ is the *abstraction sort*. Symbol $\times$ denotes the *product sort*, which is associative; $\sigma_1 \times \ldots \times \sigma_k$ stands for the sort of the empty product when $k = 0$, which we may write as $\mathbf{1}$. We write S for the set of nominal sorts.

The arity of each function symbol $f_{ij} \in F$ with $1 \leq i \leq n$ and $1 \leq j \leq m_i$ is $\sigma_{ij} \to \delta_i$, where δ_i is a base sort and m_i is the number of different ways of constructing data of that sort.

We let $\mathcal{V}$ be a set that contains a countably infinite collection of *variable names* (variables for short) x_σ, y_σ, z_σ, ... for each sort σ, such that the sets of variables $\mathcal{V}_\sigma$ of each sort are mutually disjoint. We also assume that $\mathcal{V}$ is disjoint from $\mathbb{A}$.

▶ **Definition 3.2** (Raw terms)**.** Let $\Sigma = (\Delta, A, F)$ be a signature. The set of *raw terms over signature* Σ *and set of variables* $\mathcal{V}$ (*raw terms* for short) is given by the grammar

$$t_\sigma ::= \quad x_\sigma \mid a_\alpha \mid (\pi \bullet t_\sigma)_\sigma \mid ([a_\alpha]t_\sigma)_{[\alpha]\sigma} \mid (t_{\sigma_1}, \ldots, t_{\sigma_k})_{\sigma_1 \times \ldots \times \sigma_k} \mid (f_{ij}(t_{\sigma_{ij}}))_{\delta_i},$$

where term x_σ is a *variable* of sort σ, term a_α is an *atom* of sort α, term $(\pi \bullet t_\sigma)_\sigma$ is a *moderated term* (i.e. the explicit, or delayed, permutation π over term t_σ), term $([a_\alpha]t_\sigma)_{[\alpha]\sigma}$ is the *abstraction of atom* a_α *in term* t_σ, term $(t_{\sigma_1}, \ldots, t_{\sigma_k})_{\sigma_1 \times \ldots \times \sigma_k}$ is the *product of terms* t_{σ_1}, ..., t_{σ_k}, and term $(f_{ij}(t_{\sigma_{ij}}))_{\delta_i}$ is the *datum of base sort* δ_i *constructed from term* $t_{\sigma_{ij}}$ *and function symbol* $f_{ij} : \sigma_{ij} \to \delta_i$. When they are clear from the context or immaterial, we leave the arities and sorts implicit and write x, a, $\pi \bullet t$, $[a]t$, $(t_1, \ldots, t_k)$, $f(t)$, etc.

The raw terms are the inhabitants of the carrier of the free algebra over the set of variables $\mathcal{V}$ and over the S-sorted conventional signature that consists of the function symbols in F, together with a constant symbol for each atom a_α, a unary symbol that produces moderated terms for each permutation π and each sort σ, a unary symbol that produces abstractions for each atom a_α and sort σ, and a k-ary symbol that produces a product of sort $\sigma_1 \times \ldots \times \sigma_k$ for each sequence of sorts $\sigma_1, \ldots, \sigma_k$. (See [18] for a classic presentation of term algebras, initial algebra semantics and free algebras.)

We write $\mathbb{T}(\Sigma, \mathcal{V})_\sigma$ for the set of raw terms of sort σ. A raw term t is *ground* iff no variables occur in t. We write $\mathbb{T}(\Sigma)_\sigma$ for the set of ground terms of sort σ. The sets of raw terms (resp. ground terms) of each sort are mutually disjoint as terms carry sort information. Therefore we sometimes identify the family $(\mathbb{T}(\Sigma, \mathcal{V})_\sigma)_{\sigma \in \mathsf{S}}$ of S-indexed raw terms and the family $(\mathbb{T}(\Sigma)_\sigma)_{\sigma \in \mathsf{S}}$ of S-indexed ground terms with their respective ranges $\bigcup_{\sigma \in \mathsf{S}} \mathbb{T}(\Sigma, \mathcal{V})_\sigma$ and $\bigcup_{\sigma \in \mathsf{S}} \mathbb{T}(\Sigma)_\sigma$, which we abbreviate as $\mathbb{T}(\Sigma, \mathcal{V})$ and $\mathbb{T}(\Sigma)$ respectively.

The set $\mathbb{T}(\Sigma, \mathcal{V})$ of raw terms is a nominal set, with the Perm $\mathbb{A}$-action and the support of a raw term given by:

$$
\begin{array}{rclcrcl}
\pi \cdot x & = & x & \qquad & \mathrm{supp}(x) & = & \emptyset \\
\pi \cdot a & = & \pi\, a & & \mathrm{supp}(a) & = & \{a\} \\
\pi \cdot (\pi_1 \bullet t) & = & (\pi \cdot \pi_1) \bullet (\pi \cdot t) & & \mathrm{supp}(\pi \bullet t) & = & \mathrm{supp}(\pi) \cup \mathrm{supp}(t) \\
\pi \cdot [a]t & = & [\pi\, a](\pi \cdot t) & & \mathrm{supp}([a](t)) & = & \{a\} \cup \mathrm{supp}(t) \\
\pi \cdot (t_1, \ldots, t_k) & = & (\pi \cdot t_1, \ldots, \pi \cdot t_k) & & \mathrm{supp}((t_1, \ldots, t_k)) & = & \mathrm{supp}(t_1) \cup \ldots \cup \mathrm{supp}(t_k) \\
\pi \cdot (f(t)) & = & f(\pi \cdot t), & & \mathrm{supp}(f(t)) & = & \mathrm{supp}(t).
\end{array}
$$

The set $\mathbb{T}(\Sigma)$ of ground terms is also a nominal set since it is closed with respect to the Perm $\mathbb{A}$-action given above.

▶ **Example 3.3** (π-calculus)**.** Consider a signature Σ for the π-calculus [7, 29] given by a single atom sort ch of channel names, and base sorts pr and ac for processes and actions respectively. The function symbols (adapted from [29]) are the following:

$$
\begin{array}{lll}
F = \{\ \mathit{null} : \mathbf{1} \to \mathsf{pr}, & \mathit{par} : (\mathsf{pr} \times \mathsf{pr}) \to \mathsf{pr}, & \mathit{tauA} : \mathbf{1} \to \mathsf{ac}, \\
\quad \mathit{tau} : \mathsf{pr} \to \mathsf{pr}, & \mathit{sum} : (\mathsf{pr} \times \mathsf{pr}) \to \mathsf{pr}, & \mathit{inA} : (\mathsf{ch} \times \mathsf{ch}) \to \mathsf{ac}, \\
\quad \mathit{in} : (\mathsf{ch} \times [\mathsf{ch}]\mathsf{pr}) \to \mathsf{pr}, & \mathit{rep} : \mathsf{pr} \to \mathsf{pr}, & \mathit{outA} : (\mathsf{ch} \times \mathsf{ch}) \to \mathsf{ac}, \\
\quad \mathit{out} : (\mathsf{ch} \times \mathsf{ch} \times \mathsf{pr}) \to \mathsf{pr}, & \mathit{new} : [\mathsf{ch}]\mathsf{pr} \to \mathsf{pr}, & \mathit{boutA} : (\mathsf{ch} \times \mathsf{ch}) \to \mathsf{ac} \quad \}.
\end{array}
$$

Recalling terminology from [7, 29], *null* stands for inaction, $\mathit{tau}(p)$ for the internal action after which process p follows, $\mathit{in}(a, [b]p)$ for the input at channel a where the input name is bound to b in the process p that follows, $\mathit{out}(a, b, p)$ for the output of name b through channel a after which process p follows, $\mathit{par}(p, q)$ for parallel composition, $\mathit{sum}(p, q)$ for nondeterministic choice, $\mathit{rep}(p)$ for parallel replication, and $\mathit{new}([a]p)$ for the restriction of channel a in process p (a is private in p). Actions and processes belong to different sorts. We use tauA, $\mathit{outA}(a, b)$, $\mathit{inA}(a, b)$ and $\mathit{boutA}(a, b)$ respectively for the internal action, output action, the input action and the bound output action.

The set of terms of the π-calculus corresponds to the subset of ground terms over Σ of sort pr and ac in which no moderated (sub-)terms occur. For instance, the process $(\nu b)(\overline{a}b.0)$ corresponds to the ground term $\mathit{new}([b](\mathit{out}(a, b, \mathit{null})))$, whose support is $\{a, b\}$. Both free and bound channel names (such as the a and b respectively in the example process) are represented by atoms.

The set of ground terms also contains generalised processes and actions with moderated (sub-)terms $\pi \bullet p$, which stand for a delayed permutation π that ought to be applied to a term p, e.g. $\mathit{new}(\pi \bullet ([b](\mathit{out}(a, b, \mathit{null}))))$. ◀

Raw terms allow variables to occur in the place of any ground subterm. The variables represent *unknowns*, and should be mistaken neither with free nor bound channel names. For instance, the raw term $new([b](out(a,b,x)))$ represents a process $(\nu b)(\overline{a}b.P)$ where the x is akin to the meta-variable P, which stands for some unknown process. The process $(\nu b)(\overline{a}b.P)$ unifies with $(\nu b)(\overline{a}b.0)$ by replacing P with 0. In the nominal setting, the raw term $new([b](out(a,b,x)))$ unifies with ground term $new([b](out(a,b,null)))$, by means of a *substitution* φ such that $\varphi(x) = null$. Formally, substitutions are defined below.

▶ **Definition 3.4** (Substitution). A *substitution* $\varphi : \mathcal{V} \to_{\mathrm{fs}} \mathbb{T}(\Sigma, \mathcal{V})$ is a sort-preserving, finitely supported function from variables to raw terms.

The *domain* $\mathrm{dom}(\varphi)$ *of a substitution* φ is the set $\{x \mid \varphi(x) \neq x\}$. A substitution φ is *ground* iff $\varphi(x) \in \mathbb{T}(\Sigma)$ for every variable $x \in \mathrm{dom}(\varphi)$.

The set of substitutions is a nominal set. The *extension to raw terms* $\overline{\varphi}$ *of substitution* φ is the unique homomorphism induced by φ from the free algebra $\mathbb{T}(\Sigma, \mathcal{V})$ to itself, which coincides with the function given by:

$$\begin{array}{rclcrcl} \overline{\varphi}(x) & = & \varphi(x) & & \overline{\varphi}([a]t) & = & [a](\overline{\varphi}(t)) \\ \overline{\varphi}(a) & = & a & & \overline{\varphi}(t_1,\ldots,t_k) & = & (\overline{\varphi}(t_1),\ldots,\overline{\varphi}(t_k)) \\ \overline{\varphi}(\pi \bullet t) & = & \pi \bullet \overline{\varphi}(t) & & \overline{\varphi}(f(t)) & = & f(\overline{\varphi}(t)). \end{array}$$

Given substitutions φ and γ we write $\varphi \circ \gamma$ for their composition, which is defined as follows: For every variable x, $(\varphi \circ \gamma)(x) = \overline{\varphi}(t)$ where $\gamma(x) = t$. It is straightforward to check that $(\overline{\varphi \circ \gamma})(t) = \overline{\varphi}(\overline{\gamma}(t))$ (we sometimes write $\varphi(\gamma(t))$ instead of $\overline{\varphi}(\overline{\gamma}(t))$ for short).

We note that our definition of substitution is different form those in both [8, 30], where the authors consider a function that performs the delayed permutations of the moderated terms *on-the-fly*.

▶ **Lemma 3.5** (Extension to raw terms is equivariant). *Let φ be a substitution and π a permutation. Then, $\pi \cdot \overline{\varphi} = \overline{\pi \cdot \varphi}$.*

It is straightforward to check that the support of $\overline{\varphi}$ coincides with the support of φ. By the above lemma, the set of extended substitutions is also a nominal set, since it is closed with respect to the Perm $\mathbb{A}$-action. Hereafter we sometimes write $\varphi(t)$, where t is a raw term, instead of $\overline{\varphi}(t)$. We may also write φ^π instead of $\pi \cdot \overline{\varphi}$ or $\overline{\pi \cdot \varphi}$ for short.

The following result highlights the relation between substitution and the permutation action.

▶ **Lemma 3.6** (Substitution and permutation action). *Let φ be a substitution, π a permutation and t a raw term. Then, $\pi \cdot \varphi(t) = \varphi^\pi(\pi \cdot t)$.*

Our goal is to give meaning to ground terms in nominal sets. To this end, we need a suitable class of algebraic structures that can be used to give an *interpretation* of those ground terms.

▶ **Definition 3.7** (Σ-structure). Let $\Sigma = (\Delta, A, F)$ be a signature. A Σ-*structure* M consists of a nominal set $M[\![\sigma]\!]$ for each sort σ defined as follows

$$\begin{array}{rcl} M[\![\alpha]\!] & = & \mathbb{A}_\alpha \\ M[\![[\alpha]\sigma]\!] & = & [\mathbb{A}_\alpha](M[\![\sigma]\!]) \\ M[\![\sigma_1 \times \ldots \times \sigma_k]\!] & = & M[\![\sigma_1]\!] \times \ldots \times M[\![\sigma_k]\!], \end{array}$$

where the $M[\![\delta_i]\!]$ with $\delta_i \in \Delta$ are given, as well as an equivariant function $M[\![f_{ij}]\!] : M[\![\sigma_{ij}]\!] \to M[\![\delta_i]\!]$ for each symbol $(f_{ij})_{\sigma_{ij} \to \delta_i} \in F$.

The notion of Σ-structure adapts that of Σ-structure in [8] to our sorting convention with atom and abstraction sorts. The Σ-structures characterise a range of interpretations of ground terms into elements of nominal sets, such that any sort σ gives rise to the expected nominal set, i.e. atom sorts give rise to sets of atoms, abstraction sorts give rise to sets of atom abstractions, and product sorts give rise to finite products of nominal sets.

Next we define the *interpretation of a ground term in a Σ-structure*, which resembles the *value of a term* in [8].

▶ **Definition 3.8** (Interpretation of ground terms in a Σ-structure)**.** Let Σ be a signature and M be a Σ-structure. The *interpretation $M[\![p]\!]$ of a ground term p in M* is given by:

$$\begin{array}{rclcrcl} M[\![a]\!] & = & a & & M[\![(p_1,\ldots,p_k)]\!] & = & (M[\![p_1]\!],\ldots,M[\![p_k]\!]) \\ M[\![\pi \bullet p]\!] & = & \pi \cdot M[\![p]\!] & & M[\![f(p)]\!] & = & M[\![f]\!](M[\![p]\!]). \\ M[\![[a]p]\!] & = & \langle a \rangle (M[\![p]\!]) & & & & \end{array}$$

The next lemma states that interpretation in a Σ-structure is equivariant and highlights the relation between interpretation and moderated terms.

▶ **Lemma 3.9** (Interpretation and moderated terms)**.** *Let M be a Σ-structure. Interpretation in M is equivariant, that is, $\pi \cdot M[\![p]\!] = M[\![\pi \cdot p]\!]$ for every ground term p and permutation π. Moreover, $M[\![\pi \bullet p]\!] = M[\![\pi \cdot p]\!]$.*

Finally, we introduce the Σ-structure NT, which formalises the set of *nominal terms*.

▶ **Definition 3.10** (Σ-structure for nominal terms)**.** Let Σ be a signature. The *Σ-structure NT for nominal terms* is given by the least tuple $(NT[\![\delta_1]\!],\ldots,NT[\![\delta_n]\!])$ satisfying

$$NT[\![\delta_i]\!] = NT[\![\sigma_{i1}]\!] + \ldots + NT[\![\sigma_{im_i}]\!] \quad \text{for each base sort } \delta_i \in \Delta, \text{ and}$$

and $NT[\![f_{ij}]\!] = \mathrm{inj}_j : NT[\![\sigma_{ij}]\!] \to NT[\![\delta_i]\!]$, for each function symbol $f_{ij} \in F$.

In the conditions above, the 'less than or equal to' relation for tuples is pointwise set inclusion. The $NT[\![f_{ij}]\!]$ is the jth injection of the ith component in $(NT[\![\delta_1]\!],\ldots,NT[\![\delta_n]\!])$.

Nominal terms represent alpha-equivalence classes of raw terms by using the atom abstractions of Definition 2.2.

▶ **Definition 3.11** (Nominal terms)**.** Let Σ be a signature. The set $\mathbb{N}(\Sigma)_\sigma$ of *nominal terms over Σ of sort σ* is the domain of interpretation of the ground terms of sort σ in the Σ-structure NT, that is, $\mathbb{N}(\Sigma)_\sigma = NT[\![\sigma]\!]$.

We write $\mathbb{N}(\Sigma)_\sigma$ for the set of nominal terms of sort σ. Since the sets of nominal terms of each sort may not be mutually disjoint, we sometimes identify the family $(\mathbb{N}(\Sigma)_\sigma)_{\sigma \in \mathsf{S}}$ of S-indexed nominal terms with the disjoint union $\sum_{\sigma \in \mathsf{S}} \mathbb{N}(\Sigma)_\sigma$, which we abbreviate as $\mathbb{N}(\Sigma)$ (see Appendix B).

We sometimes write p, ℓ instead of $NT[\![p]\!]$, $NT[\![\ell]\!]$ when it is clear from the context that we are referring to the interpretation into nominal terms of ground terms p and ℓ.

The following proposition states the connection between our nominal terms and the algebraic datatypes of [25].

▶ **Proposition 3.12.** *The nominal sets $\mathbb{N}(\Sigma)_\sigma$ coincide (up to isomorphism) with the nominal algebraic datatypes of Definition 8.9 in [25].*

4 Specifications of NRTSs

The NRTSs of Definition 2.4 are meant to be a model of computation for transition systems with name-binding operators and state/residual presentation. In this section we present syntactic specifications for NRTSs. We start by defining nominal residual signatures.

▶ **Definition 4.1** (Nominal residual signature)**.** A *nominal residual signature* (a residual signature for short) is a quintuple $\Sigma = (\Delta, A, \sigma, \rho, F)$ such that (Δ, A, F) is a nominal signature and σ and ρ are distinguished nominal sorts over Δ and A, which we call *state sort* and *residual sort* respectively. We say that $\mathbb{N}(\Sigma)_\sigma$ is the set of *states* and $\mathbb{N}(\Sigma)_\rho$ is the set of *residuals*.

Let $\mathcal{T} = (S, R, \longrightarrow)$ be an NRTS and $\Sigma = (\Delta, A, \sigma, \rho, F)$ be a residual signature. We say that $\mathcal{T}$ is an NRTS *over signature* Σ iff the sets of states S and residuals R coincide with the sets of nominal terms of state sort $\mathbb{N}(\Sigma)_\sigma$ and residual sort $\mathbb{N}(\Sigma)_\rho$ respectively.

Our next goal is to introduce syntactic specifications of NRTSs, which we call nominal residual transition system specifications. To this end, we will make use of residual formulas and freshness assertions over raw terms, which are defined below.

▶ **Definition 4.2** (Residual formula and freshness assertion)**.** A *residual formula* (a *formula* for short) over a residual signature Σ is a pair (s, r), where $s \in \mathbb{T}(\Sigma, \mathcal{V})_\sigma$ and $r \in \mathbb{T}(\Sigma, \mathcal{V})_\rho$. We use the more suggestive $s \longrightarrow r$ in lieu of (s, r). A formula $s \longrightarrow r$ is *ground* iff s and r are ground terms.

A *freshness assertion* (an *assertion* for short) over a signature Σ is a pair (a, t) where $a \in \mathbb{A}$ and $t \in \mathbb{T}(\Sigma, \mathcal{V})$. We will write $a \not\# t$ in lieu of (a, t). An assertion is *ground* iff t is a ground term.

▶ Remark. Formulas and assertions are raw syntactic objects, similar to raw terms, which will occur in the rules of the nominal residual transition system specifications to be defined, and whose purpose is to represent respectively transitions and freshness relations involving nominal terms. A formula $s \longrightarrow r$ (resp. an assertion $a \not\# t$) unifies with a ground formula $\varphi(s) \longrightarrow \varphi(r)$ (resp. a ground assertion $a \not\# \varphi(t)$), which in turn represents a transition $NT[\![\varphi(s)]\!] \longrightarrow NT[\![\varphi(r)]\!]$ (resp. a freshness relation $a\#NT[\![\varphi(t)]\!]$). For the assertions, notice how the symbols $\not\#$, $\#$ and $NT[\![\ ]\!]$ interact. The ground assertion $a \not\# [a]a$ represents the freshness relation $a\#NT[\![[a]a]\!]$, which is true. On the other hand, the freshness relation $a\#[a]a$ is false because $a \in \mathrm{supp}([a]a)$.

Permutation action and substitution extend to formulas and assertions in the expected way.

Formulas and assertions are elements of nominal sets. Their support is the union of the supports of the raw terms in them, hence we write $\mathrm{supp}(t \longrightarrow t')$ and $\mathrm{supp}(a \not\# t)$. We will also write $b\#(t \longrightarrow t')$ and $b\#(a \not\# t)$ for freshness relations involving formulas and assertions respectively.

▶ **Definition 4.3** (Nominal residual transition system specification)**.** Let Σ be a residual signature $(\Delta, A, \sigma, \rho, F)$. A *transition rule over* Σ (a *rule*, for short) is of the form

$$\frac{\{u_i \longrightarrow u'_i \mid i \in I\} \quad \{a_j \not\# v_j \mid j \in J\}}{t \longrightarrow t'}$$

(abbreviated as $H, \nabla / t \longrightarrow t'$) where $H = \{u_i \longrightarrow u'_i \mid i \in I\}$ is a finitely supported set of formulas over Σ (we call H the set of *premisses*) and where $\nabla = \{a_j \not\# v_j \mid j \in J\}$ is a finite

set of assertions over Σ (we call ∇ the *freshness environment*). We say formula $t \longrightarrow t'$ over Σ is the *conclusion*, where t is the *source* and t' is the *target*. A rule is an *axiom* iff it has an empty set of premisses. Note that axioms might have a non-empty freshness environment.

A *nominal residual transition system specification over* Σ (abbreviated to NRTSS) is a set of transition rules over Σ.

Permutation action and substitution extend to rules in the expected way as well. Permutation action and substitution are applied to each of the formulas and freshness assertions in the rule.

Notice that the rules of an NRTSS are elements of a nominal set. The support of a rule is the union of the supports of its premisses, freshness assertions and conclusion. In the sequel we write supp(Ru) for the support of rule Ru, and $a\#$Ru for a freshness relation involving atom a and rule Ru. Observe that the set H of premisses of a rule may be infinite, but its support must be finite. However, the freshness environment ∇ must be finite in order to make the simplification rules of Definition 5.4 to follow terminating. These simplification rules will be used in Section 5 to define the rule format in Definition 5.8.

Let $\mathcal{R}$ be an NRTSS. We say that the formula $s \longrightarrow r$ *unifies* with rule Ru in $\mathcal{R}$ iff Ru has conclusion $t \longrightarrow t'$ and $s \longrightarrow r$ is a substitution instance of $t \longrightarrow t'$. If s and r are ground terms, we also say that transition $NT[\![s]\!] \longrightarrow NT[\![r]\!]$ unifies with Ru.

▶ **Definition 4.4** (Proof tree)**.** Let Σ be a residual signature and $\mathcal{R}$ be an NRTSS over Σ. A proof tree in $\mathcal{R}$ of a transition $NT[\![s]\!] \longrightarrow NT[\![r]\!]$ is an upwardly branching rooted tree without paths of infinite length whose nodes are labelled by transitions such that

(i) the root is labelled by $NT[\![s]\!] \longrightarrow NT[\![r]\!]$, and
(ii) if $K = \{NT[\![q_i]\!] \longrightarrow NT[\![q_i']\!] \mid i \in I\}$ is the set of labels of the nodes directly above a node with label $NT[\![p]\!] \longrightarrow NT[\![p']\!]$, then there exist a rule

$$\frac{\{u_i \longrightarrow u_i' \mid i \in I\} \qquad \{a_j \not\# v_j \mid j \in J\}}{t \longrightarrow t'}$$

in $\mathcal{R}$ and a ground substitution φ such that $\varphi(t \longrightarrow t') = p \longrightarrow p'$ and, for each $i \in I$ and for each $j \in J$, $\varphi(u_i \longrightarrow u_i') = q_i \longrightarrow q_i'$ and $a_j\#NT[\![\varphi(v_j)]\!]$ holds.

We say that $NT[\![s]\!] \longrightarrow NT[\![r]\!]$ is *provable* in $\mathcal{R}$ iff it has a proof tree in $\mathcal{R}$. The transition relation specified by $\mathcal{R}$ consists of all the transitions that are provable in $\mathcal{R}$.

The nodes of a proof tree are labelled by transitions, which contain nominal terms (i.e. syntactic objects that use the atom abstractions of Definition 2.2). The use of nominal terms captures the convention in typical nominal calculi of considering terms 'up to alpha-equivalence'. (See Appendix C for an example explaining this point.)

The fact that the nodes of a proof tree are labelled by nominal terms is the main difference between our approach and previous work in nominal structural operational semantics [1], nominal rewriting [9, 30] and nominal algebra [15]. In all these works, the 'up-to-alpha-equivalence' transitions are explicitly instrumented within the model of computation by adding to the specification system inference rules that perform alpha-conversion of raw terms.

5 Rule formats for NRTSSs

This section defines rule formats for NRTSSs. The two main rule formats ensure that: (i) an NRTSS induces an equivariant transition relation, and thus an NRTS of Definition 2.4;

(ii) an NRTSS induces a transition relation which, together with an equivariant function bn, corresponds to an NTS of Definition 2.3 [24]. For the latter, we need to ensure that the induced transition relation is equivariant and satisfies *alpha-conversion of residuals* (recall, if $p \longrightarrow (\ell, p')$ is provable in $\mathcal{R}$ and a is in the set of binding names of ℓ, then for every atom b that is fresh in (ℓ, p') the transition $p \longrightarrow (a\,b) \cdot (\ell, p')$ is also provable).

As a first step, we introduce a rule format ensuring equivariance of the induced transition relation.

▶ **Definition 5.1** (Equivariant format)**.** Let $\mathcal{R}$ be an NRTSS. $\mathcal{R}$ is in *equivariant format* iff the rule $(a\,b) \cdot \textsc{Ru}$ is in $\mathcal{R}$, for every rule $\textsc{Ru}$ in $\mathcal{R}$ and for each $a, b \in \mathbb{A}$.

▶ **Lemma 5.2.** *Let $\mathcal{R}$ be an NRTSS in equivariant format. For every rule $\textsc{Ru}$ in $\mathcal{R}$ and for every permutation π, the rule $\pi \cdot \textsc{Ru}$ is in $\mathcal{R}$.*

▶ **Theorem 5.3** (Rule format for NRTSs)**.** *Let $\mathcal{R}$ be an NRTSS. If $\mathcal{R}$ is in equivariant format then $\mathcal{R}$ induces an NRTS.*

Before introducing a rule format ensuring alpha-conversion of residuals, we adapt to our freshness environments the simplification rules and the entailment relation of Definition 10 and Lemma 15 in [9], which we will use in the definition of the rule format.

▶ **Definition 5.4** (Simplification of freshness environments)**.** Consider a signature Σ. The following rules, where a, b are assumed to be distinct atoms and ∇ is a freshness environment over Σ, define *simplification of freshness environments*:

$$\begin{array}{rcl@{\qquad}rcl}
\{a \not\# b\} \cup \nabla & \Longrightarrow & \nabla & \{a \not\# [a]p\} \cup \nabla & \Longrightarrow & \nabla \\
\{a \not\# \pi \bullet t\} \cup \nabla & \Longrightarrow & \{\pi^{-1} \cdot a \not\# t\} \cup \nabla & \{a \not\# f(p)\} \cup \nabla & \Longrightarrow & \{a \not\# p\} \cup \nabla \\
\{a \not\# [b]p\} \cup \nabla & \Longrightarrow & \{a \not\# p\} \cup \nabla & \{a \not\# (p_1, \ldots, p_k)\} \cup \nabla & \Longrightarrow & \{a \not\# p_i, \ldots, a \not\# p_k\} \cup \nabla.
\end{array}$$

The rules define a reduction relation on freshness environments. We write $\nabla \Longrightarrow \nabla'$ when ∇' is obtained from ∇ by applying one simplification rule, and $\Longrightarrow^*$ for the reflexive and transitive closure of $\Longrightarrow$.

▶ **Lemma 5.5.** *The relation $\Longrightarrow$ is confluent and terminating.*

A freshness assertion is *reduced* iff it is of the form $a \not\# a$ or $a \not\# x$. We say that $a \not\# a$ is *inconsistent* and $a \not\# x$ is *consistent*. An environment ∇ is *reduced* iff it consists only of reduced assertions. An environment containing a freshness assertion that is not reduced can always be simplified using one of the rules in Definition 5.4. Therefore, by Lemma 5.5, an environment ∇ reduces by $\Longrightarrow^*$ to a unique reduced environment, which we call the *normal form* of ∇, written $\langle \nabla \rangle_{nf}$. An environment ∇ is *inconsistent* iff $\langle \nabla \rangle_{nf}$ contains some inconsistent assertion. We say ∇ *entails* ∇' (written $\nabla \vdash \nabla'$) iff either ∇ is an inconsistent environment, or $\langle \nabla' \rangle_{nf} \subseteq \langle \nabla \rangle_{nf}$. We write $\vdash \nabla$ iff $\emptyset \vdash \nabla$.

▶ **Lemma 5.6.** *Let ∇ be an environment over Σ. Then, for every ground substitution φ, the conjunction of the freshness relations represented by $\varphi(\langle \nabla \rangle_{nf})$ holds iff the conjunction of the freshness relations represented by $\varphi(\nabla)$ hold.*

In particular, if $\vdash \nabla$ then for every ground substitution φ the freshness relations represented by $\varphi(\nabla)$ hold.

We are interested in NTS and from now on we consider signatures with base sorts ac and pr, with a single atom sort ch and with source and residual sorts pr and $\mathsf{ac} \times \mathsf{pr}$ respectively. We let Σ_{NTS} be any such signature parametric on a set F of function symbols that we keep

implicit. We let $\mathrm{bn} : \mathbb{N}(\Sigma)_{\mathsf{ac}} \to \mathcal{P}_\omega(\mathbb{A}_{\mathsf{ch}})$ be the binding-names function of a given NTS. From now on we require that the rules of an NRTSS only contain ground actions ℓ and therefore function bn is always defined over $NT[\![\ell]\!]$. (Recall that we write $\mathrm{bn}(\ell)$ instead of $\mathrm{bn}(NT[\![\ell]\!])$ since it is clear in this context that the ℓ stands for a nominal term.) The rule format that we introduce in Definition 5.8 relies on identifying the rules that give rise to transitions with actions ℓ such that $\mathrm{bn}(\ell)$ is non-empty. To this end, we adapt the notion of strict stratification from [3, 14].

▶ **Definition 5.7** (Partial strict stratification)**.** Let $\mathcal{R}$ be an NRTSS over a signature Σ_{NTS} and bn be a binding-names function. Let S be a partial map from pairs of processes and ground actions to ordinal numbers. S is a *partial strict stratification of* $\mathcal{R}$ iff

(i) $S(\varphi(t), \ell) \neq \bot$, for every rule in $\mathcal{R}$ with conclusion $t \longrightarrow (\ell, t')$ such that $bn(\ell)$ is non-empty and for every ground substitution φ, and
(ii) $S(\varphi(u_i), \ell_i) < S(\varphi(t), \ell)$, for every rule in $\mathcal{R}$ with conclusion $t \longrightarrow (\ell, t')$ and for every premiss $u_i \longrightarrow (\ell_i, u_i')$ of $\mathcal{R}$, and for every ground substitution φ such that $S(\varphi(t), \ell) \neq \bot$ and $S(\varphi(u_i), \ell_i) \neq \bot$.

We say a pair (p, ℓ) of ground process and action *has order* $S(p, \ell)$.

The choice of S determines which rules will be considered by the rule format for NRTSSs of Definition 5.8 below, which guarantees that the induced transition relation satisfies alpha-conversion of residuals and, therefore, the associated transition relation together with function bn are indeed an NTS. We will intend the S to be such that the only rules with defined order are those that may take part in proof trees of transitions with some binding atom in the action.

▶ **Definition 5.8** (Alpha-conversion-of-residuals format)**.** Let $\mathcal{R}$ be an NRTSS over a signature Σ_{NTS} and bn be a binding-names function. Assume that all the actions occurring in the rules of $\mathcal{R}$ are ground. Let

$$\frac{\{u_i \longrightarrow (\ell_i, u_i') \mid i \in I\} \qquad \nabla}{t \longrightarrow (\ell, t')}\ \textsc{Ru}$$

be a rule in $\mathcal{R}$ such that there exists a partial strict stratification S and a ground substitution φ such that $S(\varphi(t), \ell) \neq \bot$. Let D be the set of variables that occur in the source t of Ru but do not occur in the premisses $u_i \longrightarrow (\ell_i, u_i')$ with $i \in I$, the environment ∇ or the target t' of the rule. The rule Ru is in *alpha-conversion-of-residuals format* (*ACR format* for short) iff there exists a ground substitution γ such that $\mathrm{dom}(\gamma) \subseteq D$, and for every atom a in the set $\mathbb{A} \setminus \{c \in \mathrm{supp}(t) \mid \langle\{c \# t\}\rangle_{nf} = \emptyset\}$ and for every atom $b \in \mathrm{bn}(\ell)$, the following conditions hold:

(i) $\{a \# t'\} \cup \nabla \vdash \{a \# u_i' \mid i \in I\}$,
(ii) $\{a \# t'\} \cup \nabla \cup \{a \# u_i \mid i \in I\} \vdash \{a \# \gamma(t)\}$, and
(iii) $\nabla \cup \{b \# u_i \mid i \in I \wedge b \in \mathrm{bn}(\ell_i)\} \vdash \{b \# \gamma(t)\}$.

An NTRSS $\mathcal{R}$ is in *ACR format* iff $\mathcal{R}$ is in equivariant format, has a partial strict stratification S and all the rules in $\mathcal{R}$ are in ACR format.

Let us explain the intuition behind the rule format. Given a transition $p \longrightarrow (\ell, q)$ that unifies with the conclusion of Ru, the rule format ensures that the binding atom b is fresh in p, and also that any atom a fresh in (ℓ, q) is also fresh in p. Constraints (i) and (ii) of Definition 5.8 are not required for atoms a that for sure are fresh in p. This is the case for the atoms in $\{c \in \mathrm{supp}(t) \mid \langle\{c \# t\}\rangle_{nf} = \emptyset\}$. For instance, take rule ResB from Section 6.

Condition $\{c \not\# (boutA(a,b), new([c]y)), c \not\# boutA(a,b)\} \vdash \{c \not\# (boutA(a,b), y)\}$ does not hold because $c \not\# [c]y$ does not entail that $c \not\# y$. However, for a transition $NT[\![new([c]p)]\!] \longrightarrow NT[\![(\ell, new([c]p'))]\!]$, c is fresh in $NT[\![new([c]p)]\!]$ even if c is not fresh in $NT[\![p]\!]$.

The purpose of substitution γ is to ignore the variables that occur in the source of a rule but are dropped everywhere else in the rule. Consider rule SumBL from Section 6 and variable x_2 occurring only in its source. Condition $\{c \not\# (boutA(a,b), y_1), c \not\# x_1\} \vdash \{c \not\# sum(x_1, x_2)\}$ does not hold because neither the premisses nor the environment of the rule provide information about c being fresh in the dropped x_2. However, this is immaterial when showing that the transition system enjoys alpha-conversion of residuals. For a transition $NT[\![sum(p,q)]\!] \longrightarrow NT[\![(\ell, p')]\!]$, the existence of transition $NT[\![sum(p,q)]\!] \longrightarrow NT[\![(b\,c) \cdot (\ell, p')]\!]$ for each atom $c \# NT[\![(\ell, p')]\!]$ does not depend on the dropped q, and constraint $\{c \not\# (boutA(a,b), y_1), c \not\# x_1\} \vdash \{c \not\# \gamma(sum(x_1, x_2))\}$ does hold with $\gamma(x_2) = null$.

▶ **Theorem 5.9** (Rule format for NTSs). *Let $\mathcal{R}$ be an NRTSS. If $\mathcal{R}$ is in ACR format then the NRTS induced by $\mathcal{R}$, together with the binding-names function* bn*, constitute an NTS. That is, the transition relation induced by $\mathcal{R}$ is equivariant and satisfies alpha-conversion of residuals.*

6 Example of application to the early π-calculus

Consider the NRTSS $\mathcal{R}$ for the early π-calculus [22] over a signature Σ_{NTS} where F is the set of function symbols from Example 3.3. Below we collect an excerpt of the rules (see Figure 1 in Appendix E for the whole NRTSS):

$$\frac{b \not\# [c]x}{in(a,[c]x) \longrightarrow (inA(a,b), (c\,b) \bullet x)}\ \textsc{In} \qquad \frac{x_1 \longrightarrow (boutA(a,b), y_1)}{sum(x_1,x_2) \longrightarrow (boutA(a,b), y_1)}\ \textsc{SumBL}$$

$$\frac{}{out(a,b,x) \longrightarrow (outA(a,b), x)}\ \textsc{Out} \qquad \frac{x \longrightarrow (boutA(a,b), y)}{rep(x) \longrightarrow (boutA(a,b), (par(y, rep(x))))}\ \textsc{RepB}$$

$$\frac{x_1 \longrightarrow (boutA(a,b), y_1) \qquad b \not\# x_2}{par(x_1,x_2) \longrightarrow (boutA(a,b), (par(y_1,x_2)))}\ \textsc{ParResL}$$

$$\frac{x_1 \longrightarrow (boutA(a,b), y_1) \qquad x_2 \longrightarrow (inA(a,b), y_2) \qquad b \not\# x_2}{par(x_1,x_2) \longrightarrow (tauA, new([b](par(y_1,y_2))))}\ \textsc{CloseL}$$

$$\frac{x \longrightarrow (outA(a,b), y) \qquad b \not\# a}{new([b]x) \longrightarrow (boutA(a,b), y)}\ \textsc{Open} \qquad \frac{x \longrightarrow (boutA(a,b), y) \qquad c \not\# boutA(a,b)}{new([c]x) \longrightarrow (boutA(a,b), new([c]y))}\ \textsc{ResB}$$

where $a, b, c \in \mathbb{A}_{\mathsf{ch}}$.

According to the early semantics, an input process $NT[\![in(a,[c]p)]\!]$ can perform a transition to a process $NT[\![(c\,b) \cdot p]\!]$ that is obtained by substituting a channel name b received through channel a for channel name c in p. In the rule In, the moderated term $(c\,b) \bullet x$ needs to be used in order to indicate that permutation $(c\,b)$ will be performed over the term substituted for variable x.

The rule CloseL specifies the interaction of a process such as $NT[\![new([b](out(a,b,p)))]\!]$, which exports a private channel name b through channel a, composed in parallel with an

input process such as $NT[\![in(a,[c]q)]\!]$ that reads through channel a. The private name b is exported and the resulting process $NT[\![new([b](par(p,(c\,b)\cdot q)))]\!]$ is the parallel composition of processes p and q where atom b is restricted. For illustration, transition $NT[\![new([b](out(a,b,p)))]\!] \longrightarrow NT[\![(boutA(a,b),p)]\!]$ is provable in $\mathcal{R}$ by the following proof tree:

$$\dfrac{\dfrac{d\#NT[\![(outA(a,b),p)]\!]}{NT[\![out(a,b,p)]\!] \longrightarrow NT[\![(outA(a,b),p)]\!]}\ \textsc{Out} \qquad b\#a}{NT[\![new([b](out(a,b,p)))]\!] \longrightarrow NT[\![(boutA(a,b),p)]\!]}\ \textsc{Open}.$$

We use the rule format of Definition 5.8 to show that $\mathcal{R}$, together with equivariant function $\mathrm{bn}(\ell) = \{b \mid \ell = boutA(a,b)\}$ specifies an NTS. We consider the following partial strict stratification:

$$\begin{array}{rcll}
S(out(a,b,p), outA(a,b)) & = & 0 & \\
S(par(p,q), boutA(a,b)) & = & 1+\max\{S(p, boutA(a,b)), S(q, boutA(a,b))\} & \\
S(sum(p,q), boutA(a,b)) & = & 1+\max\{S(p, boutA(a,b)), S(q, boutA(a,b))\} & \\
S(rep(p), boutA(a,b)) & = & 1+S(p, boutA(a,b)) & \\
S(new([b]p), boutA(a,b)) & = & 1+S(p, outA(a,b)) & \\
S(new([c]p), boutA(a,b)) & = & 1+S(p, outA(a,b)) & c \notin \{a,b\} \\
S(p,\ell) & = & \bot & \text{o.w.}
\end{array}$$

We check that $\mathcal{R}$ is in ACR format as follows. The only rules in $\mathcal{R}$ whose sources and actions unify with pairs of processes and actions that have defined order are Out, ParResL, SumBL, RepB, Open and ResB, and the symmetric versions of rules ParResL and SumBL, which are omitted in the excerpt. We will only check the ACR-format for rules Out, SumBL, Open and ResB (see Appendix E for further details).

For rule Out, we have an empty set of premisses and the set D of atoms that are in $\mathrm{supp}(out(a,b,x))$ but are not in $\mathrm{supp}(outA(a,b),x)$ is empty. Therefore we can give away with substitution γ. There is no atom a such that $\langle\{a \not\# out(a,b,x)\} = \emptyset\rangle_{nf}$ and the set $\mathrm{bn}(outA(a,b))$ is empty. We only need to check that for every atom c, $\{c \not\# (outA(a,b),x)\} \vdash \{c \not\# out(a,b,x)\}$. For atoms $c \in \mathrm{supp}(outA(a,b),x)$ the obligation of the rule format vacuously holds, and therefore it is enough to pick an atom c fresh in the rule and check that $\{c \not\# (outA(a,b),x)\} \vdash \{c \not\# out(a,b,x)\}$, which is straightforward.

For rule SumBL, we have premiss $x_1 \longrightarrow (boutA(a,b), y_1)$ and the set D contains x_2. We pick γ such that $\gamma(x_2) = null$. There is no atom a such that $\langle\{a \not\# sum(x_1,x_2)\}\rangle_{nf} = \emptyset$ and the set $\mathrm{bn}(boutA(a,b))$ contains atom b. Again, it is enough to pick atom c fresh in the rule and check that

$$\begin{array}{c}
\{c \not\# (boutA(a,b),y_1)\} \vdash \{c \not\# (boutA(a,b),y_1)\} \quad \text{and} \\
\{c \not\# (boutA(a,b),y_1), c \not\# x_1\} \vdash \{c \not\# \gamma(sum(x_1,x_2))\} \quad \text{and} \\
\{b \not\# x_1\} \vdash \{b \not\# \gamma(sum(x_1,x_2))\},
\end{array}$$

which holds since $\gamma(sum(x_1,x_2)) = sum(x_1, null)$ and $b \not\# null$ reduces to the empty set.

For rule Open the set D is empty and $\langle\{b \not\# new([b]x)\}\rangle_{nf} = \emptyset$. It is enough to pick atom c fresh in the rule (and therefore different from b) and check that

$$\begin{array}{c}
\{c \not\# (boutA(a,b),y), b \not\# a\} \vdash \{c \not\# (boutA(a,b),y)\} \quad \text{and} \\
\{c \not\# (boutA(a,b),y), b \not\# a, c \not\# x\} \vdash \{c \not\# new([b]x))\} \quad \text{and} \\
\{b \not\# x, b \not\# a\} \vdash \{b \not\# new([b]x)\},
\end{array}$$

which holds because $b \not\# new([b]x)$ reduces to the empty set.

For rule ResB the set D is empty and $\langle\{c \not\# new([c]x)\}\rangle_{nf} = \emptyset$. It is enough to pick atom d fresh in the rule (and therefore different from c) and check that

$$\{d \not\# (boutA(a,b), new([c]y)), c \not\# boutA(a,b)\} \vdash \{d \not\# (boutA(a,b), y)\} \quad \text{and}$$
$$\{d \not\# (boutA(a,b), new([c]y)), c \not\# boutA(a,b), d \not\# x\} \vdash \{d \not\# new([c]x))\} \quad \text{and}$$
$$\{b \not\# x, c \not\# boutA(a,b)\} \vdash \{b \not\# new([c]x)\},$$

which holds because $d \not\# x$ and $b \not\# x$ entail $d \not\# new([c]x)$ and $b \not\# new([c]x)$ respectively.

Atoms a, b and c in the specification of $\mathcal{R}$ range over $\mathbb{A}_{\mathsf{ch}}$, and thus $\mathcal{R}$ is in equivariant format. Therefore $\mathcal{R}$ is in ARC format. By Theorem 5.9 the NRTS induced by $\mathcal{R}$, together with function bn, constitute an NTS of Definition 2.3.

7 Conclusions and future work

The work we have presented in this paper stems from the recently proposed Nominal SOS (NoSOS) framework [7] and from earlier proposals for nominal logic in [8,15,30]. It is by no means the only approach studied so far in the literature that aims at a uniform treatment of binders and names in programming and specification languages. Other existing approaches that accommodate variables and binders within the SOS framework are those proposed by Fokkink and Verhoef in [13], by Middelburg in [20, 21], by Bernstein in [5], by Ziegler, Miller and Palamidessi in [31] and by Fiore and Staton in [10] (originally, by Fiore and Turi in [11]). The aim of all of the above-mentioned frameworks is to establish sufficient syntactic conditions guaranteeing the validity of a semantic result (congruence in the case of [5,10,20,31] and conservativity in the case of [13,21]). In addition, Gabbay and Mathijssen present a nominal axiomatization of the λ-calculus in [17].

Our current proposal aims at following closely the spirit of the seminal work on nominal techniques by Gabbay, Pitts and their co-workers, and paves the way for the development of results on rule formats akin to those presented in the aforementioned references. Amongst those, we consider the development of a congruence format for the notion of bisimilarity presented in [24, Def. 2] to be of particular interest. The logical characterisation of bisimilarity given in [24] opens the intriguing possibility of employing the divide-and-congruence approach from [12] to obtain an elegant congruence format and a compositional proof system for the logic.

In the NTSs of Parrow *et al.* [24], scope opening is modelled by the property of alpha-conversion of residuals. We are currently exploring an alternative in which scope opening is encoded by a *residual abstraction* of sort $[\mathsf{ch}](\mathsf{ac} \times \mathsf{pr})$. We have developed mutual, one-to-one translations between the NTSs and the NRTSs with residual abstractions. The generality of our NRTSs also allows for neat specifications of variants of the π-calculus such as Sangiorgi's internal π-calculus [28].

Developing rule formats for SOS is always the result of a trade-off between ease of application and generality. Our rule format for alpha-conversion of residuals in Definition 5.8 is no exception and might be generalised in various ways. For instance, the quantification on atom a in conditions (i) and (ii), and the use of substitution γ might be made more general by a finer analysis of the variable flow in a rule. Another generalisation of the rule format would consider possibly open raw actions.

Finally, we are developing other rule formats for properties other than alpha-conversion of residuals. One such rule format ensures a *non-dropping property* for NRTSs to the effect that, in each transition, the support of a state is a subset of the support of its derivative.

References

1 L. Aceto, M. Cimini, M. J. Gabbay, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. Nominal structural operational semantics. In preparation.

2 L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, chapter 3, pages 197–292. Elsevier, 2001.

3 L. Aceto, I. Fábregas, Á. García-Pérez, and A. Ingólfsdóttir. A unified rule format for bounded nondeterminism in SOS with terms as labels. *Journal of Logical and Algebraic Methods in Programming*, 2017. In press.

4 J. Bengtson and J. Parrow. Formalising the pi-calculus using nominal logic. *Logical Methods in Computer Science*, 5(2), 2009.

5 K. L. Bernstein. A congruence theorem for structured operational semantics of higher-order languages. In *13th Annual IEEE Symposium on Logic in Computer Science*, pages 153–164. IEEE Computer Society, 1998.

6 S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra: The Millennium Edition*. Springer Verlag, 2000.

7 M. Cimini, M. R. Mousavi, M. A. Reniers, and M. J. Gabbay. Nominal SOS. *Electronic Notes in Theoretical Computer Science*, 286:103–116, 2012.

8 R. Clouston and A. Pitts. Nominal equational logic. *Electronic Notes in Theoretical Computer Science*, 172:223–257, 2007.

9 M. Fernández and M. J. Gabbay. Nominal rewriting. *Information and Computation*, 205(6):917–965, 2007.

10 M. P. Fiore and S. Staton. A congruence rule format for name-passing process calculi. *Information and Computation*, 207(2):209–236, 2009.

11 M. P. Fiore and D. Turi. Semantics of name and value passing. In *16th Annual IEEE Symposium on Logic in Computer Science*, pages 93–104. IEEE Computer Society, 2001.

12 W. Fokkink, R. J. van Glabbeek, and P. de Wind. Compositionality of Hennessy-Milner logic by structural operational semantics. *Theoretical Computer Science*, 354(3):421–440, 2006.

13 W. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Information and Computation*, 146(1):24–54, 1998.

14 W. Fokkink and T. D. Vu. Structural operational semantics and bounded nondeterminism. *Acta Informatica*, 39(6-7):501–516, 2003.

15 M. J. Gabbay and A. Mathijssen. Nominal (universal) algebra: Equational logic with names and binding. *Journal of Logic and Computation*, 19(6):1455–1508, 2009.

16 M. J. Gabbay and A. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13(3–5):341–363, 2002.

17 Murdoch James Gabbay and Aad Mathijssen. A nominal axiomatization of the lambda calculus. *Journal of Logic and Computation*, 20(2):501–531, 2010.

18 J. A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright. Initial algebra semantics and continuous algebras. *Journal of the ACM*, 24(1):68–95, 1977.

19 J. Lambek. A fixed point theorem for complete categories. *Mathematische Zeitung*, 103:151–161, 1968.

20 C. A. Middelburg. Variable binding operators in transition system specifications. *Journal of Logic and Algebraic Programming*, 47(1):15–45, 2001.

21 C. A. Middelburg. An alternative formulation of operational conservativity with binding terms. *Journal of Logic and Algebraic Programming (JLAP)*, 55(1-2):1–19, 2003.

22 R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (Parts I and II). *Information and Computation*, 100(1):1–77, 1992.

23 M. R. Mousavi, M. A. Reniers, and J. F. Groote. SOS formats and meta-theory: 20 years after. *Theoretical Computer Science*, 373(3):238–272, 2007.

24 J. Parrow, J. Borgström, L.-H. Eriksson, R. Gutkovas, and T. Weber. Modal logics for nominal transition systems. In *26th International Conference on Concurrency Theory*, volume 42 of *LIPIcs*, pages 198–211. Schloss Dagstuhl, 2015.

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

26 A. Pitts. Nominal techniques. *ACM SIGLOG News*, 3(1):57–72, 2016.

27 Gordon D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.

28 D. Sangiorgi. pi-Calculus, internal mobility, and agent-passing calculi. *Theoretical Computer Science*, 167(1&2):235–274, 1996.

29 D. Sangiorgi and D. Walker. *The π-calculus — A Theory of Mobile Processes*. Cambridge University Press, 2001.

30 C. Urban, A. Pitts, and M. J. Gabbay. Nominal unification. *Theoretical Computer Science*, 323(1–3):473–497, 2004.

31 A. Ziegler, D. Miller, and C. Palamidessi. A congruence format for name-passing calculi. *Electronic Notes in Theoretical Computer Science*, 156(1):169–189, 2006.

A Preliminaries

A.1 Category Nom

Nominal sets are the objects of a category **Nom** whose morphisms are the equivariant functions. The category **Nom** is closed under finite products and both finite and infinite coproducts.[1] We write $s = \mathrm{inj}_i s'$ with $i \in I$ and $s' \in S_i$ for an element s in a coproduct $\sum_{i\in I}(S_i)$. (For a finite coproduct $S_1 + \ldots + S_n$ we let $I = \{1, \ldots, n\}$.) For other set-theoretical operations (i.e. infinite products, functions, partial functions, power sets) the following caveat applies. The category of nominal sets is closed under the variant of each operation that restricts any universal quantification that is involved in the operation to quantify only over finitely supported elements (see Sections 2.2 to 2.5 of [25]).

The *nominal function set* between nominal sets S and T (Definition 2.18 of [25]) is the nominal set $(T^S)_{\mathrm{fs}}$ of finitely supported functions from S to T—equivariant or not, recall that an equivariant function has empty support. (We may write $S \to_{\mathrm{fs}} T$ as well.) The application and currying functions can be respectively restricted to equivariant functions $\mathrm{app} : (Y \to_{\mathrm{fs}} X) \times X \to Y$ and $\mathrm{curry} : (Z \times X \to_{\mathrm{fs}} Y) \to Z \to (X \to_{\mathrm{fs}} Y)$ such that the nominal function set coincides with the *exponential object* in **Nom**, i.e. there is a bijection between hom-sets $\mathbf{Nom}(Z \times X, Y)$ and $\mathbf{Nom}(Z, X \to_{\mathrm{fs}} Y)$ given by sending $f \in \mathbf{Nom}(Z \times X, Y)$ to $\mathrm{curry}(f) \in \mathbf{Nom}(Z, X \to_{\mathrm{fs}} Y)$. (Section 2.4 in [25] spells out all the details on this isomorphism.)

The category **Nom** is Cartesian closed (Theorem 2.19 in [25]), i.e., **Nom** admits all the finite products (including the empty product **1** which is the terminal object) and all the exponentials.

Now we present some additional definitions and results on nominal sets.

▶ **Definition A.1** (Permutation action on a function (Sec. 1.4 [25])). Given $f : X \to Y$ a function, and $\pi \in \mathrm{Perm}\ \mathbb{A}$. For each $x \in X$, the action of permutation π over f is defined as: $\pi \cdot f(x) = \pi \cdot (f(\pi^{-1} \cdot x))$.

▶ Remark. Conjugation yields that for every $\mathrm{Perm}\ \mathbb{A}$-set S the action of π on $s \in S$ is equivariant. Indeed,

$$\pi \cdot (\pi_1 \cdot s) = (\pi \circ \pi_1) \cdot s = (\pi \circ \pi_1 \circ \pi^{-1} \circ \pi) \cdot s = ((\pi \cdot \pi_1) \circ \pi) \cdot s = (\pi \cdot \pi_1) \cdot (\pi \cdot s).$$

It is also straightforward to show that composition of permutations is equivariant. In fact,

$$\pi \cdot (\pi_1 \circ \pi_2) = \pi \circ (\pi_1 \circ \pi_2) \circ \pi^{-1} = (\pi \circ \pi_1 \circ \pi^{-1})(\pi \circ \pi_2 \circ \pi^{-1}) = (\pi \cdot \pi_1) \circ (\pi \cdot \pi_2).$$

B Nominal Terms

The function $\mathrm{fa} : \mathbb{T}(\Sigma, \mathcal{V}) \to \mathcal{P}_\omega(\mathbb{A})$ delivers the set of free atoms in a raw term:

$$\begin{array}{rclcrcl} \mathrm{fa}(x) & = & \emptyset & & \mathrm{fa}([a]t) & = & \mathrm{fa}(t) \setminus \{a\} \\ \mathrm{fa}(a) & = & \{a\} & & \mathrm{fa}(t_1, \ldots, t_k) & = & \mathrm{fa}(t_1) \cup \ldots \cup \mathrm{fa}(t_k) \\ \mathrm{fa}(\pi \bullet t) & = & \pi \cdot (\mathrm{fa}(t)) & & \mathrm{fa}(f(t)) & = & \mathrm{fa}(t). \end{array}$$

Notice that the set of free atoms in a raw term differs from the support of the term. For instance, $\mathrm{fa}([a](a, b)) = \{b\}$, but $\mathrm{supp}([a](a, b)) = \{a, b\}$.

[1] In **Nom**, coproducts correspond to disjoint unions.

▶ **Lemma B.1.** *Let t be a raw term, then* $\mathrm{fa}(t) \subseteq \mathrm{supp}(t)$.

Proof. By induction on the structure of t. The only non-trivial case is $t = \pi \bullet t'$. By the induction hypothesis, $\mathrm{fa}(t') \subseteq \mathrm{supp}(t')$. If $\mathrm{supp}(\pi)$ and $\mathrm{supp}(t')$ are disjoint sets, then $\pi \cdot \mathrm{fa}(t') = \mathrm{fa}(t')$ and the lemma holds. Otherwise, $\pi \cdot \mathrm{fa}(t') \subseteq \mathrm{supp}(\pi) \cup \mathrm{supp}(t') = \mathrm{supp}(\pi \bullet t')$ because every atom that is in $\pi \cdot \mathrm{fa}(t')$ and not in $\mathrm{fa}(t')$ has to be in $\mathrm{supp}(\pi)$. ◀

Proof of Lemma 3.5. We prove $(\pi \cdot \overline{\varphi})(t) = \overline{\pi \cdot \varphi}(t)$ by induction on the structure of raw term t. By conjugation,

$$(\pi \cdot \overline{\varphi})(x) = \pi \cdot \overline{\varphi}(\pi^{-1} \cdot x) = \pi \cdot \overline{\varphi}(x) = \pi \cdot \varphi(x)$$
$$= \pi \cdot \varphi(\pi^{-1} \cdot x) = (\pi \cdot \varphi)(x) = \overline{\pi \cdot \varphi}(x)$$

and the lemma holds for the base case $t = x$. Similarly,

$$(\pi \cdot \overline{\varphi})(a) = \pi \cdot \overline{\varphi}(\pi^{-1} \cdot a) = \pi \cdot (\pi^{-1} \cdot a) = a = \overline{\pi \cdot \varphi}(a)$$

and the lemma holds for the base case $t = a$. The rest of the cases are straightforward by the induction hypothesis. ◀

Proof of Lemma 3.6. By definition of φ^π, we have that $\varphi^\pi(\pi \cdot t) = \pi \cdot \varphi(\pi^{-1} \cdot (\pi \cdot t)) = \pi \cdot \varphi(t)$ and we are done. ◀

Proof of Lemma 3.9. We first prove that $M[\![\pi \bullet p]\!] = M[\![\pi \cdot p]\!]$. We proceed by induction on the structure of p. The base case $p = a$ is trivial. If $p = \pi_1 \bullet q$, then

$$M[\![\pi \bullet (\pi_1 \bullet q)]\!] = \pi \cdot M[\![\pi_1 \bullet q]\!] = \pi \cdot (\pi_1 \cdot M[\![q]\!]) = (\pi \circ \pi_1) \cdot M[\![q]\!]$$
$$= (\pi \circ \pi_1 \circ \pi^{-1} \circ \pi) \cdot M[\![q]\!] = (\pi \circ \pi_1 \circ \pi^{-1}) \cdot M[\![\pi \bullet q]\!],$$

which by the induction hypothesis equals

$$(\pi \circ \pi_1 \circ \pi^{-1}) \cdot (\pi \cdot M[\![q]\!]) = (\pi \cdot \pi_1) \cdot M[\![\pi \cdot q]\!]$$
$$= M[\![(\pi \cdot \pi_1) \bullet (\pi \cdot q)]\!] = M[\![\pi \cdot (\pi_1 \bullet q)]\!].$$

If $p = [a]q$, then

$$M[\![\pi \bullet ([a]q)]\!] = \pi \cdot M[\![[a]q]\!] = \pi \cdot (\langle a \rangle (M[\![q]\!]))$$
$$= \langle \pi \cdot a \rangle (\pi \cdot (M[\![q]\!])) = \langle \pi \cdot a \rangle ((M[\![\pi \bullet q]\!])),$$

which by the induction hypothesis equals to

$$\langle \pi \cdot a \rangle ((M[\![\pi \cdot q]\!])) = M[\![[\pi \cdot a](\pi \cdot q)]\!] = M[\![\pi \cdot ([a]q)]\!].$$

The rest of the cases are straightforward by the induction hypothesis.

Now, equivariance of $M[\![\ ]\!]$ follows by Definition 3.8, i.e. $\pi \cdot M[\![p]\!] = M[\![\pi \bullet p]\!] = M[\![\pi \cdot p]\!]$. ◀

Connection between our nominal terms and the nominal algebraic datatypes of [25]

We first illustrate the nominal terms by the means of the signature Σ for π-calculus in Example 3.3.

▶ **Example B.2.** The Σ-structure NT is given by the least pair $(NT[\![\mathsf{pr}]\!], NT[\![\mathsf{ac}]\!])$ of nominal sets satisfying the following set equations

$$\begin{array}{rcl} NT[\![\mathsf{pr}]\!] & = & NT[\![\mathbf{1}]\!] + NT[\![\mathsf{pr}]\!] + NT[\![\mathsf{ch} \times [\mathsf{ch}]\mathsf{pr}]\!] + NT[\![\mathsf{ch} \times \mathsf{ch} \times \mathsf{pr}]\!] \\ & & + NT[\![\mathsf{pr} \times \mathsf{pr}]\!] + NT[\![\mathsf{pr} \times \mathsf{pr}]\!] + NT[\![\mathsf{pr}]\!] + NT[\![[\mathsf{ch}]\mathsf{pr}]\!] \\ & = & \{()\} + NT[\![\mathsf{pr}]\!] + (\mathbb{A}_{\mathsf{ch}} \times [\mathbb{A}_{\mathsf{ch}}](NT[\![\mathsf{pr}]\!])) + (\mathbb{A}_{\mathsf{ch}} \times \mathbb{A}_{\mathsf{ch}} \times NT[\![\mathsf{pr}]\!]) \\ & & + (NT[\![\mathsf{pr}]\!] \times NT[\![\mathsf{pr}]\!]) + (NT[\![\mathsf{pr}]\!] \times NT[\![\mathsf{pr}]\!]) + NT[\![\mathsf{pr}]\!] + [\mathbb{A}_{\mathsf{ch}}](NT[\![\mathsf{pr}]\!]), \\ NT[\![\mathsf{ac}]\!] & = & NT[\![\mathbf{1}]\!] + NT[\![\mathsf{ch} \times \mathsf{ch}]\!] + NT[\![\mathsf{ch} \times \mathsf{ch}]\!] + NT[\![\mathsf{ch} \times \mathsf{ch}]\!] \\ & = & \{()\} + (\mathbb{A}_{\mathsf{ch}} \times \mathbb{A}_{\mathsf{ch}}) + (\mathbb{A}_{\mathsf{ch}} \times \mathbb{A}_{\mathsf{ch}}) + (\mathbb{A}_{\mathsf{ch}} \times \mathbb{A}_{\mathsf{ch}}), \end{array}$$

together with an equivariant function for each function symbol in F (we only show a few)

$$\begin{array}{rcl} NT[\![null]\!] = \mathrm{inj}_1 & : & \{()\} \to NT[\![\mathsf{pr}]\!] \\ NT[\![tau]\!] = \mathrm{inj}_2 & : & NT[\![\mathsf{pr}]\!] \to NT[\![\mathsf{pr}]\!] \\ NT[\![out]\!] = \mathrm{inj}_4 & : & \mathbb{A}_{\mathsf{ch}} \times [\mathbb{A}_{\mathsf{ch}}](NT[\![\mathsf{pr}]\!]) \to NT[\![\mathsf{pr}]\!] \\ NT[\![new]\!] = \mathrm{inj}_8 & : & [\mathbb{A}_{\mathsf{ch}}](NT[\![\mathsf{pr}]\!]) \to NT[\![\mathsf{pr}]\!] \\ NT[\![tauA]\!] = \mathrm{inj}_1 & : & \{()\} \to NT[\![\mathsf{ac}]\!] \\ NT[\![boutA]\!] = \mathrm{inj}_4 & : & \mathbb{A}_{\mathsf{ch}} \times \mathbb{A}_{\mathsf{ch}} \to NT[\![\mathsf{ac}]\!]. \end{array}$$

Consider the process $(\nu b)(\overline{a}b.0)$ encoded as the ground term $new([b](out(a, b, null)))$, whose interpretation in NT is $\mathrm{inj}_8(\langle b\rangle(\mathrm{inj}_4(a, b, \mathrm{inj}_1())))$. ◀

▶ Remark. Recall that the constructor inj_j for disjoint union has the polymorphic type

$$\mathrm{inj}_j : \forall(S_1 + \ldots + S_m).S_j \to S_1 + \ldots + S_m, \qquad \text{where } j \leq m.$$

Therefore, a nominal term may have 'polymorphic sort' and the sets of nominal terms of each sort may not be mutually disjoint. For instance, both ground terms *null* and *tauA* have the same interpretation $\mathrm{inj}_1()$ in NT. However, each of the $NT[\![null]\!]$ and $NT[\![tauA]\!]$ live in different components of the carrier $(NT[\![\mathsf{pr}]\!], NT[\![\mathsf{ac}]\!])$ of the T-algebra induced by the Σ-structure NT and, by all means, the sort information is never lost. Here we are not concerned with this technical subtlety and, at any rate, we can always determine the 'monomorphic sort' of a given nominal term by using implicit type parameters (within curly braces) that fix the set $S_1+\ldots+S_n$ over which each constructor inj_j is universally quantified, i.e. $NT[\![null]\!] = \mathrm{inj}_1\{NT[\![\mathsf{pr}]\!]\}()$.

The nominal term with implicit type parameters that corresponds to process $(\nu b)(\overline{a}b.0)$ is $\mathrm{inj}_8\{NT[\![\mathsf{pr}]\!]\}(\langle b\rangle(\mathrm{inj}_4\{NT[\![\mathsf{pr}]\!]\}(a, b, \mathrm{inj}_1\{NT[\![\mathsf{pr}]\!]\}())))$.

The remainder shows that the nominal terms are connected to the elements of the nominal algebraic data types of Definition 8.9 in [25]. We follow closely the exposition on initial algebraic semantics for nominal algebraic data types in [25]. The reader is advised to read Sections 8.3 and 8.4 of [25] alongside.

Let $\mathbf{Nom}^n = \mathbf{Nom}\times\ldots_{n \text{ times}}\ldots\times\mathbf{Nom}$ be the n-product category and let $T : \mathbf{Nom}^n \to \mathbf{Nom}^n$ be the nominal algebraic functor induced by a signature Σ (see Section 8.3 of [25]), which we describe next. Given an n-tuple $S = (S_1, \ldots, S_n)$ of nominal sets, each sort σ gives rise to a nominal set $[\![\sigma]\!]S$ defined by:

$$\begin{array}{rcl} [\![\delta_i]\!]S & = & S_i \\ [\![\alpha]\!]S & = & \mathbb{A}_\alpha \\ [\![[\alpha]\sigma]\!]S & = & [\mathbb{A}_\alpha]([\![\sigma]\!]S) \\ [\![\sigma_1, \ldots, \sigma_k]\!]S & = & [\![\sigma_1]\!]S \times \ldots \times [\![\sigma_k]\!]S. \end{array}$$

Let the sorts σ_{ij} be such that $f_{ij} : \sigma_{ij} \to \delta_i$ are the function symbols of signature Σ. The nominal algebraic functor T has components $T_i : \mathbf{Nom}^n \to \mathbf{Nom}$ mapping each $S = (S_1, \ldots, S_m) \in \mathbf{Nom}^n$ to

$$T_i S = [\![\sigma_{i1}]\!]S + \ldots + [\![\sigma_{im_i}]\!]S,$$

and similarly for n-tuples of equivariant functions.

A Σ-structure M gives rise to a T-algebra with carrier the n-tuple of nominal sets $S = (M[\![\delta_1]\!], \ldots, M[\![\delta_n]\!])$ and with morphism the n-tuple of equivariant functions $F = (F_1, \ldots, F_n)$ where $F_i(\mathrm{inj}_j s) = M[\![f_{ij}]\!](s)$ for each $s \in S_i$.

Proof of Proposition 3.12. Let $D = (NT[\![\delta_1]\!], \ldots, NT[\![\delta_n]\!])$ together with morphism $I = (I_1, \ldots, I_n)$ such that $I_i(\mathrm{inj}_j s) = NT[\![f_{ij}]\!](s)$ be the T-algebra induced by Σ-structure NT. It is routine to check that I maps $T(D)$ to D, where T is the nominal algebraic functor induced by signature Σ, and that the morphism I coincides with the identity. Since D is the least tuple satisfying this condition, then the tuple coincides with the least fixed point of functor T. By a well known result by Lambek [19], (D, I) constitutes the initial T-algebra. The proposition follows by Theorem 8.15 in [25]. ◀

C Specifications of NRTSs

Permutation action, substitution and the function fa defined on page 17 extend to residual formulas and freshness assertions in the expected way, i.e.

$$\begin{array}{rcl}
\pi \cdot (s \longrightarrow r) & = & \pi \cdot s \longrightarrow \pi \cdot r \\
\pi \cdot (a \fresh t) & = & \pi \cdot a \fresh \pi \cdot t \\
\\
\varphi(s \longrightarrow r) & = & \varphi(s) \longrightarrow \varphi(r) \\
\varphi(a \fresh t) & = & a \fresh \varphi(t) \\
\\
\mathrm{fa}(r \longrightarrow s) & = & \mathrm{fa}(r) \cup \mathrm{fa}(s) \\
\mathrm{fa}(a \fresh t) & = & \{a\} \cup \mathrm{fa}(t).
\end{array}$$

Residual formulas and freshness assertions are elements of nominal sets. The support of a residual formula (respectively a freshness assertion) is the union of the supports of the raw terms in it. We write $\mathrm{supp}(t \longrightarrow t')$ and $\mathrm{supp}(a \fresh t)$ for the supports of residual formula $t \longrightarrow t'$ and of freshness assertion $a \fresh t$ respectively. We write $b\#(t \longrightarrow t')$ and $b\#(a \fresh t)$ for the freshness relations that involve atom b and residual formula $t \longrightarrow t'$ and freshness assertion $a \fresh t$ respectively.

▶ **Example C.1.** Consider the residual signature with base sort b, atom sort a, two function symbols f, g with arity $[\mathsf{a}]\mathsf{a} \to \mathsf{b}$ and state and residual sorts equal to b. Let $\mathcal{R}$ be the NRTSS defined by the rules:

$$\frac{}{g(x) \longrightarrow g(x)}\ \textsc{Ax} \qquad \frac{g([a]a) \longrightarrow g([b]b) \quad a \fresh b}{f([a]a) \longrightarrow f([b]b)}\ \textsc{Ru}\ , \quad \text{where } a, b \in \mathbb{A}_{\mathsf{a}}.$$

The nominal term $NT[\![f([a]a)]\!]$ is equal to $NT[\![f([b]b)]\!]$, and $NT[\![g([a]a)]\!]$ is equal to $NT[\![g([b]b)]\!]$, so the transition $NT[\![f([a]a)]\!] \longrightarrow NT[\![f([a]a)]\!]$ is provable with the following proof tree,

where rule Ax is instantiated using a ground substitution φ such that $\varphi(x) = [a]a$:[2]

$$\dfrac{\dfrac{}{NT[\![g([a]a)]\!] \longrightarrow NT[\![g([b]b)]\!]}\ \text{Ax} \qquad a\#b}{NT[\![f([a]a)]\!] \longrightarrow NT[\![f([b]b)]\!]}\ \text{Ru}.$$

Intuitively, the freshness assertion $a \not\# b$ in rule Ru is superfluous because it references atoms a and b, which do not occur free in the rule (i.e. $a, b \notin \mathrm{fa}(f([a]a) \longrightarrow f([b]b))$ and $a, b \notin \mathrm{fa}(g([a]a) \longrightarrow g([b]b))$). ◀

D Rule formats for NRTSSs

Proof of Lemma 5.2. The claim follows straightforwardly by Definition 5.1, since each permutation π is equivalent to a composition of transpositions $(a_1\, b_1) \circ \ldots \circ (a_n\, b_n)$ with $n \geq 0$. ◀

Proof of Theorem 5.3. We prove that the transition relation induced by $\mathcal{R}$ is equivariant. That is, if $NT[\![p]\!] \longrightarrow NT[\![p']\!]$ then $\pi \cdot NT[\![p]\!] \longrightarrow \pi \cdot NT[\![p']\!]$ for every permutation π. We proceed by induction on the height of the proof tree of $NT[\![p]\!] \longrightarrow NT[\![p']\!]$. Assume that the last rule used in this proof is

$$\dfrac{\{u_i \longrightarrow u_i' \mid i \in I\} \qquad \{a_j \not\# v_j \mid j \in J\}}{t \longrightarrow t'}\ \text{Ru}$$

and that for some ground substitution φ the premisses $NT[\![\varphi(u_i)]\!] \longrightarrow NT[\![\varphi(u_i')]\!]$, $i \in I$ are provable in $\mathcal{R}$, the freshness relations $a_j \# NT[\![\varphi(v_j)]\!]$, $j \in J$ hold, and $\varphi(t) \longrightarrow \varphi(t') = p \longrightarrow p'$. Since $\mathcal{R}$ is in equivariant format, by Lemma 5.2 $\mathcal{R}$ contains the rule

$$\dfrac{\{\pi \cdot u_i \longrightarrow \pi \cdot u_i' \mid i \in I\} \qquad \{\pi \cdot a_j \not\# \pi \cdot v_j \mid j \in J\}}{\pi \cdot t \longrightarrow \pi \cdot t'}\ \text{Ru}_\pi.$$

Our goal now is to show that the transition $\pi \cdot NT[\![p]\!] \longrightarrow \pi \cdot NT[\![p']\!]$ is provable using rule Ru_π and substitution φ^π defined on page 6. Let $j \in J$. By Lemma 3.6 we know that $\pi \cdot \varphi(v_j) = \varphi^\pi(\pi \cdot v_j')$. Moreover, since $\#$ is equivariant, by Lemma 3.9, the freshness relation $\pi \cdot a_j \# NT[\![\varphi^\pi(\pi \cdot t_j)]\!]$ holds. Assume now that $i \in I$. We know that the premiss $\pi \cdot NT[\![\varphi(u_i)]\!] \longrightarrow \pi \cdot NT[\![\varphi(u_i')]\!]$ is provable in $\mathcal{R}$ by the induction hypothesis ($I = \emptyset$ corresponds to the base case, i.e. a rule without premisses). By Lemmas 3.6 and 3.9, this premiss is equal to $NT[\![\varphi^\pi(\pi \cdot u_i)]\!] \longrightarrow NT[\![\varphi^\pi(\pi \cdot u_i')]\!]$. Therefore, the transition $\pi \cdot NT[\![p]\!] \longrightarrow \pi \cdot NT[\![p']\!]$ is provable using rule Ru_π and substitution φ^π because it is equal to $NT[\![\varphi^\pi(\pi \cdot t)]\!] \longrightarrow NT[\![\varphi^\pi(\pi \cdot t')]\!]$ by Lemmas 3.6 and 3.9. ◀

▶ Remark. It is straightforward to check that the proof tree of transition $(a\, b) \cdot p \longrightarrow (a\, b) \cdot p'$ obtained in the proof above coincides with the proof tree of $p \longrightarrow p'$ with atoms a and b transposed, and that both proof trees have the same height.

[2] Extending the existing convention to our notion of proof tree, we depict proof trees as trees of inference rules where the conclusion and premisses in each rule are replaced by *the transitions denoted by* their substitution instances, and where the freshness assertions in each rule are replaced by *the freshness relations denoted by* their substitution instances.

The proofs of the lemmas and theorems to come use induction on the size of a freshness environment. We let the *size of a freshness environment* ∇ be the sum of the sizes of the raw terms in its assertions, where the size of a term is the number of nodes of its abstract syntax tree.

Proof of Lemma 5.5. The proof goes along the same lines of the proof of Lemma 11 in [9]. Since the simplification rules do not overlap, there are no critical pairs and confluence holds trivially. Moreover, as the size of the environment decreases with each simplification rule, the reduction relation is terminating. ◀

Proof of Lemma 5.6. By straightforward induction on the size of ∇. We provide the proof for the cases $\nabla = \{a \not\# \pi \bullet t\} \cup \nabla'$ and $\nabla = \{a \not\# [a]t\} \cup \nabla'$ for illustration. If $\nabla = \{a \not\# \pi \bullet t\} \cup \nabla'$, then $a\#NT[\![\pi \bullet \varphi(t)]\!]$ iff $\pi^{-1} \cdot a\#NT[\![\varphi(t)]\!]$ by Lemmas 3.6 and 3.9. The assertion $a \not\# \pi \bullet t$ simplifies to $\pi^{-1} \cdot a \not\# t$ and the lemma follows by the induction hypothesis. If $\nabla = \{a \not\# [a]t\} \cup \nabla'$, then $a\#NT[\![[a]\varphi(t)]\!]$ trivially holds. The freshness assertion $a \not\# [a]t$ vanishes when simplifying the environment and the lemma follows by the induction hypothesis because $\langle\nabla\rangle_{nf} = \langle\nabla'\rangle_{nf}$. ◀

▶ Remark. Notice that if ∇ in the lemma above is inconsistent, then the lemma follows trivially since no substitution φ exists such that all the freshness assertions denoted by $\varphi(\nabla)$ hold. This is so because $\langle\nabla\rangle_{nf}$ contains some freshness assertion of the form $a \not\# a$, and neither the conjunction of the freshness relations denoted by $\varphi(\nabla)$ holds, nor the conjunction of the ones denoted by $\varphi(\langle\nabla\rangle_{nf})$ does.

▶ **Lemma D.1.** *Let $\mathcal{R}$ be an NRTSS and* Ru *be a rule*

$$\frac{\{u_i \longrightarrow (\ell_i, u_i') \mid i \in I\} \qquad \{a_j \not\# v_j \mid j \in J\}}{t \longrightarrow (\ell, t')}\ \textsc{Ru}$$

in $\mathcal{R}$. Let D be the set of variables that occur in the source t of Ru *but do not occur in the premisses $u_i \longrightarrow (\ell_i, u_i')$ with $i \in I$, the environment ∇ or the target t' of the rule. For every $\gamma : D \to \mathbb{T}(\Sigma_{\mathrm{NTS}})$ and every substitution φ, a proof tree for transition $NT[\![\varphi(\gamma(t))]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!]$ that uses* Ru *as last rule exists iff a proof tree for transition $NT[\![\varphi(t)]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!]$ that uses* Ru *as last rule exists.*

Proof. Since the domain D of γ contains variables neither in the premisses nor in the target of rule Ru, $NT[\![\varphi(\gamma(t))]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!]$ is equal to $NT[\![\varphi(\gamma(t))]\!] \longrightarrow NT[\![\varphi(\gamma(\ell, t'))]\!]$. Consider a proof tree of transition $NT[\![\varphi(\gamma(t))]\!] \longrightarrow NT[\![\varphi(\gamma(\ell, t'))]\!]$ that uses Ru as last rule, if it exists. Since none of the variables occurring in the premisses and in the environment are in the domain of γ, $\varphi(w) = \varphi(\gamma(w))$ for each w in $\{u_i, u_i' \mid i \in I\} \cup \{v_j \mid j \in J\}$. Hence, the sub-trees that prove the premisses $NT[\![\varphi(\gamma(u_i))]\!] \longrightarrow NT[\![\varphi(\gamma(\ell_i, u_i'))]\!]$, with $i \in I$, also prove transitions $NT[\![\varphi(u_i)]\!] \longrightarrow NT[\![\varphi(\ell_i, u_i')]\!]$; and also all $a_j\#\varphi(\gamma(v_j))$ and $a_j\#\varphi(v_j)$ with $j \in J$ hold. Therefore, in the case they exist, the proof trees for transitions $NT[\![\varphi(\gamma(t))]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!]$ and $NT[\![\varphi(t)]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!]$ that use Ru as last rule share the same sub-trees for the premisses, the freshness assertions hold, and the only difference between the proof trees is the root node. ◀

Proof of Theorem 5.9. Since $\mathcal{R}$ is in equivariant format, $\mathcal{R}$ induces an NRTS with an equivariant transition relation (Theorem 5.3). We prove that this transition relation also enjoys alpha-conversion of residuals. That is, if $NT[\![p]\!] \longrightarrow NT[\![(\ell, p')]\!]$ is provable in $\mathcal{R}$ and $b \in \mathrm{bn}(\ell)$, then $NT[\![p]\!] \longrightarrow (a\,b) \cdot NT[\![(\ell, p')]\!]$ for every atom a that is fresh in $NT[\![(\ell, p')]\!]$.

To this end, observe, first of all, that $S(p,\ell)$ is defined because $\mathrm{bn}(\ell)$ is non-empty. Assume that the last rule used in the proof of $NT[\![p]\!] \longrightarrow NT[\![(\ell,p')]\!]$ is

$$\frac{\{u_i \longrightarrow (\ell_i, u_i') \mid i \in I\} \qquad \{a_j \not\# v_j \mid j \in J\}}{t \longrightarrow (\ell, t')}\ \textsc{Ru}$$

and therefore that for some ground substitution φ

- the premisses $NT[\![\varphi(u_i)]\!] \longrightarrow NT[\![\varphi(\ell_i, u_i')]\!]$ with $i \in I$ are provable in $\mathcal{R}$,
- the freshness relations $a_j \# NT[\![\varphi(v_j)]\!]$ with $j \in J$ hold, and
- $NT[\![\varphi(t)]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!] = NT[\![p]\!] \longrightarrow NT[\![(\ell, p')]\!]$.

As rule Ru is in ACR format, there is a ground substitution γ whose domain is contained in the set of variables D occurring in t but nowhere else in the rule, meeting conditions (i)-(iii) in Definition 5.8 for each atom a in the set $\mathbb{A} \setminus \{c \in \mathrm{supp}(t) \mid \langle\{c \not\# t\}\rangle_{nf} = \emptyset\}$ and each atom b in $\mathrm{bn}(\ell)$.

We first show that transition $NT[\![\varphi(t)]\!] \longrightarrow (a\,b) \cdot NT[\![\varphi(\ell, t')]\!]$ is provable under the assumption that $a \# NT[\![\varphi(\gamma(t))]\!]$ and $b \# NT[\![\varphi(\gamma(t))]\!]$. We will then show that those assumptions hold. By Lemma D.1 we know that a proof tree of $NT[\![\varphi(\gamma(t))]\!] \longrightarrow NT[\![\varphi(\ell, t')]\!]$ that uses Ru as last rule exists, and since $\mathcal{R}$ is in equivariant format then a proof tree of $(a\,b) \cdot NT[\![\varphi(\gamma(t))]\!] \longrightarrow (a\,b) \cdot NT[\![\varphi(\ell, t')]\!]$ that uses $(a\,b) \cdot \textsc{Ru}$ as last rule exists. By our assumptions, we have that $(a\,b) \cdot NT[\![\varphi(\gamma(t))]\!] = NT[\![\varphi(\gamma(t))]\!]$ and therefore $NT[\![\varphi(\gamma(t))]\!] \longrightarrow (a\,b) \cdot NT[\![\varphi(\ell, t')]\!]$. Again by Lemma D.1, a proof tree of $NT[\![\varphi(t)]\!] \longrightarrow (a\,b) \cdot NT[\![\varphi(\ell, t')]\!]$ that uses $(a\,b) \cdot \textsc{Ru}$ as last rule exists, and the theorem holds.

In the remainder we prove the assumptions $a \# NT[\![\varphi(\gamma(t))]\!]$ and $b \# NT[\![\varphi(\gamma(t))]\!]$. Since $\mathcal{R}$ is in ACR format, then for atoms $a \in (\mathbb{A} \setminus \{c \in \mathrm{supp}(t) \mid \langle\{c \not\# t\}\rangle_{nf} = \emptyset\})$

- $\{a \not\# t'\} \cup \nabla \vdash \{a \not\# u_i' \mid i \in I\}$ and
- $\{a \not\# t'\} \cup \nabla \cup \{a \not\# u_i \mid i \in I\} \vdash \{a \not\# \gamma(t)\}$.

We prove first $a \# NT[\![\varphi(\gamma(t))]\!]$. We distinguish two cases. If $a \in \{c \in \mathrm{supp}(t) \mid \langle\{c \not\# t\}\rangle_{nf} = \emptyset\}$ then $\vdash \{a \not\# t\}$ and by Lemma 5.6, $a \# NT[\![\overline{\varphi \circ \gamma}(t))]\!] = NT[\![\varphi(\gamma(t))]\!]$ holds. Otherwise, we use Lemma 5.6 to obtain the implications

(1) $(\ a \# NT[\![\varphi(t')]\!] \ \wedge \bigwedge_{j\in J}(a_j \# NT[\![\varphi(v_j)]\!])\ ) \Longrightarrow \bigwedge_{i\in I}(a \# NT[\![\varphi(u_i')]\!])$ and
(2) $(\ a \# NT[\![\varphi(t')]\!] \ \wedge \bigwedge_{j\in J}(a_j \# NT[\![\varphi(v_j)]\!]) \ \wedge \bigwedge_{i\in I}(a \# NT[\![\varphi(u_i)]\!])\ ) \Longrightarrow a \# NT[\![\varphi(\gamma(t))]\!]$.
Since the set D does not contain any variable occurring in t' it follows that $\varphi(\gamma(t')) = \varphi(t')$. Now we prove the statement $a \# NT[\![\varphi(t')]\!] \Longrightarrow a \# NT[\![\varphi(\gamma(t))]\!]$ by induction on $S(\varphi(\gamma(t)), \ell)$ (this is enough to show the claim since $a \# NT[\![\varphi(t')]\!]$ holds by assumption). Since all $a_j \# NT[\![\varphi(v_j)]\!]$ with $j \in J$ hold, then all $a \# NT[\![\varphi(u_i')]\!]$ with $i \in I$ hold by (1). By the induction hypothesis we obtain the implications $a \# NT[\![\varphi(u_i')]\!] \implies a \# NT[\![\varphi(\gamma(u_i))]\!]$, with $i \in I$. For each $i \in I$, since the variables that occur in u_i are not in $\mathrm{dom}(\gamma)$ then $a \# NT[\![\varphi(u_i')]\!] \implies a \# NT[\![\varphi(u_i)]\!]$. And now by (2) we know that $a \# NT[\![\varphi(\gamma(t))]\!]$ which is what was to be shown. The base case is when the set I is empty, which makes $\bigwedge_{i\in I}(a \# NT[\![\varphi(u_i)]\!])$ trivially true and what we were proving holds by (2).

Now we prove the statement $b \# NT[\![\varphi(\gamma(t))]\!]$ by induction on $S(\varphi(\gamma(t)), \ell)$. Since $\mathcal{R}$ is in ACR format we have that $\nabla \cup \{b \not\# u_i \mid i \in I \wedge b \in \mathrm{bn}(\ell_i)\} \vdash \{b \not\# \gamma(t)\}$. We use Lemma 5.6 to obtain the implication

$$\Big(\ \bigwedge_{j\in J}(a_j \# NT[\![\varphi(v_j)]\!]) \ \wedge \bigwedge_{i\in I \wedge b\in \mathrm{bn}(\ell_i)} (b \# NT[\![\varphi(u_i)]\!])\ \Big) \Longrightarrow b \# NT[\![\varphi(\gamma(t))]\!].$$

By the induction hypothesis, $b\#NT[\![\varphi(\gamma(u_i))]\!]$ for each $i \in I$ such that $b \in \mathrm{bn}(\ell_i)$. For each $i \in I$, since the set D does not contain any variable occurring in u_i we know that $\varphi(\gamma(u_i)) = \varphi(u_i)$. In particular this holds for $i \in I$ such that $b \in \mathrm{bn}(\ell_i)$. By the implication above we know that $b\#NT[\![\varphi(\gamma(t))]\!]$. The base case is when $\{i \mid i \in I \wedge b \in \mathrm{bn}(\ell_i)\}$ is empty, in which case $\bigwedge_{i\in I\wedge b\in \mathrm{bn}(\ell_i)}(b\#NT[\![\varphi(u_i)]\!])$ is trivially true and we are done. ◀

E Example of application to the early π-calculus

Consider the NRTSS $\mathcal{R}$ in Figure 1 over the residual signature $\Sigma = (\{\mathsf{pr}, \mathsf{ac}\}, \{\mathsf{ch}\}, F, \mathsf{pr}, \mathsf{ac} \times \mathsf{pr})$, where F is the set of function symbols from Example 3.3. (Omitted rules ParR, ParResR, CommR, CloseR, SumR and SumBR, are respectively the symmetric version of rules ParL, ParResL, CommL, CloseL, SumL and SumBL.)

We use the rule format of Definition 5.8 to show that $\mathcal{R}$, together with equivariant function $\mathrm{bn}(\ell) = \{b \mid \ell = \mathit{boutA}(a, b)\}$ specifies an NTS. We consider the following partial strict stratification:

$$\begin{array}{rcll}
S(\mathit{out}(a,b,p), \mathit{outA}(a,b)) &=& 0 & \\
S(\mathit{par}(p,q), \mathit{boutA}(a,b)) &=& 1 + \max\{S(p, \mathit{boutA}(a,b)), S(q, \mathit{boutA}(a,b))\} & \\
S(\mathit{sum}(p,q), \mathit{boutA}(a,b)) &=& 1 + \max\{S(p, \mathit{boutA}(a,b)), S(q, \mathit{boutA}(a,b))\} & \\
S(\mathit{rep}(p), \mathit{boutA}(a,b)) &=& 1 + S(p, \mathit{boutA}(a,b)) & \\
S(\mathit{new}([b]p), \mathit{boutA}(a,b)) &=& 1 + S(p, \mathit{outA}(a,b)) & \\
S(\mathit{new}([c]p), \mathit{boutA}(a,b)) &=& 1 + S(p, \mathit{outA}(a,b)) & c \notin \{a,b\} \\
S(p, \ell) &=& \bot & \text{o.w.}
\end{array}$$

We check that $\mathcal{R}$ is in ACR format as follows. The only rules in $\mathcal{R}$ whose sources and actions unify with pairs of processes and actions that have defined order are Out, ParResL, SumBL, RepB, Open and ResB, and the symmetric versions of rules ParResL and SumBL (we will not check the ACR-format for the symmetric version of the rules).

For rule Out, we have an empty set of premisses and the set D of atoms that are in $\mathrm{supp}(\mathit{out}(a,b,x))$ but are not in $\mathrm{supp}(\mathit{outA}(a,b), x)$ is empty. Therefore we can give away with substitution γ. There is no atom a such that $\langle\{a \not\approx \mathit{out}(a,b,x)\} = \emptyset\rangle_{nf}$ and the set $\mathrm{bn}(\mathit{outA}(a,b))$ is empty. We only need to check that for every atom c, $\{c \not\approx (\mathit{outA}(a,b), x)\} \vdash \{c \not\approx \mathit{out}(a,b,x)\}$. For atoms $c \in \mathrm{supp}(\mathit{outA}(a,b), x)$ the obligation of the rule format vacuously holds, and therefore it is enough to pick an atom c fresh in the rule and check that $\{c \not\approx (\mathit{outA}(a,b), x)\} \vdash \{c \not\approx \mathit{out}(a,b,x)\}$, which is straightforward.

For ParResL we have premiss $x_1 \longrightarrow (\mathit{boutA}(a,b), y_1)$ and the set D is empty. There is no atom a such that $\langle\{a \not\approx \mathit{par}(x_1,x_2)\} = \emptyset\rangle_{nf}$ and the set $\mathrm{bn}(\mathit{boutA}(a,b))$ contains atom b. It is enough to pick atom c, which is fresh in the rule, and check that

$$\{c \not\approx (\mathit{boutA}(a,b), \mathit{par}(y_1,x_2)), b \not\approx x_2\} \vdash \{c \not\approx (\mathit{boutA}(a,b), y_1)\} \quad \text{and}$$
$$\{c \not\approx (\mathit{boutA}(a,b), \mathit{par}(y_1,x_2)), b \not\approx x_2, c \not\approx x_1\} \vdash \{c \not\approx \mathit{par}(x_1,x_2)\} \quad \text{and}$$
$$\{b \not\approx x_1, b \not\approx x_2\} \vdash \{b \not\approx \mathit{par}(x_1,x_2)\},$$

which is straightforward.

For rule SumBL, we have premiss $x_1 \longrightarrow (\mathit{boutA}(a,b), y_1)$ and the set D contains x_2. We pick γ such that $\gamma(x_2) = \mathit{null}$. There is no atom a such that $\langle\{a \not\approx \mathit{sum}(x_1,x_2)\}\rangle_{nf} = \emptyset$ and the set $\mathrm{bn}(\mathit{boutA}(a,b))$ contains atom b. Again, it is enough to pick atom c fresh in the rule and check that

$$\{c \not\approx (\mathit{boutA}(a,b), y_1)\} \vdash \{c \not\approx (\mathit{boutA}(a,b), y_1)\} \quad \text{and}$$
$$\{c \not\approx (\mathit{boutA}(a,b), y_1), c \not\approx x_1\} \vdash \{c \not\approx \gamma(\mathit{sum}(x_1,x_2))\} \quad \text{and}$$
$$\{b \not\approx x_1\} \vdash \{b \not\approx \gamma(\mathit{sum}(x_1,x_2))\},$$

which holds since $\gamma(sum(x_1, x_2)) = sum(x_1, null)$ and $b \not\# null$ reduces to the empty set.

For rule RepB the set D is empty and $\langle\{b \not\# rep(x)\}\rangle_{nf} = \emptyset$. It is enough to pick atom c fresh in the rule and check that

$$\{c \not\# (boutA(a,b), par(y, rep(x)))\} \vdash \{c \not\# (boutA(a,b), y)\} \quad \text{and}$$
$$\{c \not\# (boutA(a,b), par(y, rep(x))), c \not\# x\} \vdash \{c \not\# rep(x))\} \quad \text{and}$$
$$\{b \not\# x\} \vdash \{b \not\# rep(x)\},$$

which is straightforward.

For rule Open the set D is empty and $\langle\{b \not\# new([b]x)\}\rangle_{nf} = \emptyset$. It is enough to pick atom c fresh in the rule (and therefore different from b) and check that

$$\{c \not\# (boutA(a,b), y), b \not\# a\} \vdash \{c \not\# (boutA(a,b), y)\} \quad \text{and}$$
$$\{c \not\# (boutA(a,b), y), b \not\# a, c \not\# x\} \vdash \{c \not\# new([b]x))\} \quad \text{and}$$
$$\{b \not\# x, b \not\# a\} \vdash \{b \not\# new([b]x)\},$$

which holds because $b \not\# new([b]x)$ reduces to the empty set.

For rule ResB the set D is empty and $\langle\{c \not\# new([c]x)\}\rangle_{nf} = \emptyset$. It is enough to pick atom d fresh in the rule (and therefore different from c) and check that

$$\{d \not\# (boutA(a,b), new([c]y)), c \not\# boutA(a,b)\} \vdash \{d \not\# (boutA(a,b), y)\} \quad \text{and}$$
$$\{d \not\# (boutA(a,b), new([c]y)), c \not\# boutA(a,b), d \not\# x\} \vdash \{d \not\# new([c]x))\} \quad \text{and}$$
$$\{b \not\# x, c \not\# boutA(a,b)\} \vdash \{b \not\# new([c]x)\},$$

which holds because $d \not\# x$ and $b \not\# x$ entail $d \not\# new([c]x)$ and $b \not\# new([c]x)$ respectively.

Atoms a, b and c in the specification of $\mathcal{R}$ range over $\mathbb{A}_{\mathsf{ch}}$, and thus $\mathcal{R}$ is in equivariant format. Therefore $\mathcal{R}$ is in ACR format. By Theorem 5.9 the NRTS induced by $\mathcal{R}$, together with function bn, constitute an NTS of Definition 2.3.

$$\frac{c \not\approx [b]x}{in(a,[b]x) \longrightarrow (inA(a,c),(b\,c) \bullet x)}\ \textsc{In} \qquad \frac{}{out(a,b,x) \longrightarrow (outA(a,b),x)}\ \textsc{Out}$$

$$\frac{}{tau(x) \longrightarrow (\tau,x)}\ \textsc{Tau} \qquad \ell \notin \{boutA(a,b)\}\ \frac{x_1 \longrightarrow (\ell,y_1)}{par(x_1,x_2) \longrightarrow (\ell,(par(y_1,x_2)))}\ \textsc{ParL}$$

$$\frac{x_1 \longrightarrow (boutA(a,b),y_1) \qquad b \not\approx x_2}{par(x_1,x_2) \longrightarrow (boutA(a,b),(par(y_1,x_2)))}\ \textsc{ParResL}$$

$$\frac{x_1 \longrightarrow (outA(a,b),y_1) \qquad x_2 \longrightarrow (inA(a,b),y_2)}{par(x_1,x_2) \longrightarrow (tauA,(par(y_1,y_2)))}\ \textsc{CommL}$$

$$\frac{x_1 \longrightarrow (boutA(a,b),y_1) \qquad x_2 \longrightarrow (inA(a,b),y_2) \qquad b \not\approx x_2}{par(x_1,x_2) \longrightarrow (tauA,new([b](par(y_1,y_2))))}\ \textsc{CloseL}$$

$$\ell \notin \{boutA(a,b)\}\ \frac{x_1 \longrightarrow (\ell,y_1)}{sum(x_1,x_2) \longrightarrow (\ell,y_1)}\ \textsc{SumL} \qquad \frac{x_1 \longrightarrow (boutA(a,b),y_1)}{sum(x_1,x_2) \longrightarrow (boutA(a,b),y_1)}\ \textsc{SumBL}$$

$$\ell \notin \{boutA(a,b)\}\ \frac{x \longrightarrow (\ell,y)}{rep(x) \longrightarrow (\ell,(par(y,rep(x))))}\ \textsc{Rep}$$

$$\frac{x \longrightarrow (boutA(a,b),y)}{rep(x) \longrightarrow (boutA(a,b),(par(y,rep(x))))}\ \textsc{RepB}$$

$$\frac{x \longrightarrow (outA(a,b),y_1) \qquad x \longrightarrow (inA(a,b),y_2)}{rep(x) \longrightarrow (tauA,par(par(y_1,y_2),rep(x)))}\ \textsc{RepComm}$$

$$\frac{x \longrightarrow (boutA(a,b),y_1) \qquad x \longrightarrow (inA(a,b),y_2) \qquad b \not\approx x}{rep(x) \longrightarrow (tauA,par(new([b](par(y_1,y_2))),rep(x)))}\ \textsc{RepClose}$$

$$\frac{x \longrightarrow (outA(a,b),y) \qquad b \not\approx a}{new([b]x) \longrightarrow (boutA(a,b),y)}\ \textsc{Open} \qquad \ell \notin \{boutA(a,b)\}\ \frac{x \longrightarrow (\ell,y) \qquad c \not\approx \ell}{new([c]x) \longrightarrow (\ell,new([c]y))}\ \textsc{Res}$$

$$\frac{x \longrightarrow (boutA(a,b),y) \qquad c \not\approx boutA(a,b)}{new([c]x) \longrightarrow (boutA(a,b),new([c]y))}\ \textsc{ResB}$$

where $a,b,c \in \mathbb{A}_{\mathsf{ch}}$.

Figure 1 NRTSS $\mathcal{R}$ for the early π-calculus.