Graded Semantics of Nominal Systems

Hannes Schulze
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

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

Üsame Cengiz
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract

Nominal automata models and transition systems serve as formalisms for languages and processes carrying data, and as such relate closely to classical register-based models. The paradigm of name allocation in nominal systems helps alleviate the pervasive computational hardness of register-based models in a tradeoff between expressiveness and computational tractability. For instance, regular nondeterministic nominal automata (RNNAs) correspond, under their *local freshness* semantics, to a form of lossy register automata. Unlike the full register automaton model, RNNAs allow for inclusion checking in elementary complexity (parametrized PSpace); similarly, trace inclusion in the underlying nominal transition systems is in parametrized PSpace. In the present work, we develop a unified algebraic treatment of spectra of behavioural equivalences on nominal systems in the framework of graded monads, working in the setting of universal coalgebra. In particular, we extend the associated notion of graded algebraic theory to the nominal setting, and use this to give an algebraic axiomatization of the (linear-time) global and local freshness semantics of nominal systems with name allocation. As an illustration of the benefits of graded monads, we develop a nominal version of the generic game characterization of graded semantics, which we instantiate to obtain trace equivalence games for nominal transition systems under global and local freshness semantics.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Concurrency

Keywords and phrases Nominal transition systems, trace semantics, graded monads, coalgebra, nominal algebra

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2026.47

Related Version *Full Version*: `https://arxiv.org/abs/2510.24353` [38]

Funding *Lutz Schröder*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 517924115.
Üsame Cengiz: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 419850228.

1 Introduction

In processes and formal languages, infinite alphabets serve as an abstraction of data types that are either infinite or prohibitively large. Typical examples include data values in XML documents [31], object identities [20], channel names [22], process identifiers [5], parameters of method calls [23], or nonces in cryptographic protocols [26]. Transition systems and automata models processing infinite alphabets are conveniently described as systems based on nominal sets [4, 32], in which, roughly speaking, the underlying set of names plays the role of the infinite alphabet, and system states store a bounded number of names in analogy to the standard register paradigm. For instance, non-deterministic orbit-finite automata (NOFA) [4] are equivalent to register automata [24] with non-deterministic update [25].

37th International Conference on Concurrency Theory (CONCUR 2026).
Editors: Ana Sokolova and Patrick Totzke; Article No. 47; pp. 47:1–47:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The nominal setting naturally supports explicit name-allocating transitions, subject to α-conversion of bound names; these are featured, for instance, in *nominal transition systems* [32] and in *regular non-deterministic nominal automata (RNNA)* [37], a nominal automata model that is equivalent to a subclass of register automata characterized by a lossiness condition called *name dropping*. RNNA trade some of the expressiveness of register automata for computational tractability; in particular, they allow inclusion checking in elementary complexity (a problem that is undecidable for the full class of register automata). Indeed, name-allocating transitions support two distinct notions of freshness, determined by the discipline imposed on renaming bound names: Under *global freshness*, as featured, for instance, in the register-based model of *session automata* [5], newly read names are required to be fresh w.r.t. all previous names. In the nominal setting, this is enforced by a discipline of clean renaming (avoiding shadowing of bound names). On the other hand, *local freshness*, i.e. freshness w.r.t. (some of) the names currently stored in the registers, is induced by unrestricted renaming according to the usual rules of α-equivalence familiar from λ-calculus. These concepts have been extended to nominal automata models processing trees [35] and infinite words [41], as well as to alternating nominal automata [14].

Our main concern in the present work is to capture the various semantics of nominal systems in the general categorical framework of *graded semantics* [30], which provides a unified treatment of spectra of semantics of automata models and concurrent systems, such as the well-known linear-time / branching-time spectrum of semantics of labelled transition systems [8]. This is achieved by combining the setting of universal coalgebra [36] with additional algebraic laws introduced via the choice of a *graded monad* [39], a categorical structure that abstractly represents a *graded algebraic theory*. Here, the notion of *grade* intuitively corresponds to depth of look-ahead, measured in (exact) numbers of transition steps. A typical example of a graded algebraic equation is the equation $a(x+y) = a(x)+a(y)$, which occurs in the axiomatization of the trace semantics of labelled transition systems [8] and says that taking an a-labelled step distributes over nondeterministic choice $+$. Here, one assigns depth 1 to the unary operation $a(-)$ representing a-steps and depth 0 to the choice operator $+$, so that the overall depth of the equation is 1. The notion of graded theory has originally been introduced for set-based graded monads [39]; it has been extended to graded monads on the category of posets [10], the category of metric spaces [12], and more generally to graded monads on categories of relational structures [9, 13], while coverage of nominal systems is currently missing. Benefits of casting a given semantics as a graded semantics include instantiation of generic results obtained for graded semantics, such as logical characterizations [30, 8, 10, 12] and game characterizations [11, 13].

In this context, our main contributions in the present work are as follows.

- We introduce a notion of graded nominal algebra, based on a simplification of Gabbay and Mathijsen's framework of nominal algebra [19]. We show that graded nominal theories induce graded monads on the category of nominal sets and, crucially, that graded nominal theories of depth 1 induce graded monads that are depth-1 in an abstract categorical sense [30, 8], a key prerequisite for most technical results in graded semantics (e.g. [30, 8, 11, 13]).
- Besides more basic examples, we then present graded nominal theories capturing the global and local freshness semantics of name allocation in nominal transition systems and RNNA; in particular the case of local freshness turns out to be quite non-trivial even though it is characterized by a single simple additional axiom.
- We transfer the game characterization of graded semantics [11] to nominal sets, obtaining as instances of this result Spoiler-Duplicator games for trace equivalence of nominal transition systems under both global and local freshness. Notably, establishing the

technical conditions for correctness of the game in the case of local freshness requires a maybe unexpected twist to the setup of the corresponding graded nominal-algebraic theory, which as it turns out needs to include name restriction [34] as a depth-0 operation in order to unblock certain α-renamings, in analogy to name-dropping constructions for RNNA [37].

Related work. We have already mentioned previous work on graded semantics [30, 8, 10, 11, 12, 13] and nominal algebra [19]; details on the relationship with the latter are discussed in Remark 5.11. We note that alternative systems of varying expressiveness for universal algebra over nominal sets have been proposed [7, 28]. Some of these systems [28, 29] indeed use standard multisorted universal algebra, which however needs to be restricted to *uniform* theories and terms, and then corresponds to a subclass of (finitary) monads on nominal sets, the so-called fb-monads [29]. While many nominal-algebraic systems derive freshness judgements [18, 19, 7], sometimes in synchrony with equalities [7], our system is entirely concerned with equalities, with freshness implicit in the context of equalities. The global freshness semantics of RNNA has been modelled coalgebraically using categorical distributive laws (in what has been termed Eilenberg-Moore-type [1] and Kleisli-type [21] coalgebraic trace semantics), which by general results [30] entails a modelling as a graded semantics, without however providing a syntactic algebraic presentation [15]. Categorical distributive laws have also been used to obtain up-to techniques for proving language equivalence of NOFAs [6].

2 Preliminaries

We assume basic familiarity with category theory (e.g. [2]). We proceed to recall requisite background on nominal sets and universal coalgebra.

Nominal Sets. serve the principled treatment of *names* and *renaming* in syntax and semantics [34]; a key insight behind the framework is that many issues around renaming are best handled by permuting names. Fix a countably infinite set $\mathbb{A}$ of *names*, and let $\mathsf{Perm}(\mathbb{A})$ be the group of finite permutations on $\mathbb{A}$, which is generated by the permutations $(a\ b)$ that swap $a, b \in \mathbb{A}$. A $\mathsf{Perm}(\mathbb{A})$*-set* $(X, \cdot)$, or just X, is a set X equipped with a left action $(\pi, x) \mapsto \pi \cdot x$ of $\mathsf{Perm}(\mathbb{A})$. Then, X decomposes disjointly into the *orbits* of its elements $x \in X$, i.e. the sets $\{\pi \cdot x \mid \pi \in \mathsf{Perm}(\mathbb{A})\}$. If X has only finitely many orbits, then X is *orbit-finite*.

For $x \in X$, $A \subseteq X$, we write

$$\mathsf{fix}(x) = \{\pi \in \mathsf{Perm}(\mathbb{A}) \mid \pi \cdot x = x\} \quad \text{and} \quad \mathsf{Fix}(A) = \textstyle\bigcap_{x \in A} \mathsf{fix}(x);$$

that is, $\mathsf{fix}(x)$ is the set of permutations fixing the element x, and $\mathsf{Fix}(A)$ is the set of permutations fixing the set A pointwise. Notice that $\mathbb{A}$ itself is a $\mathsf{Perm}(\mathbb{A})$-set with the action given by $\pi \cdot a = \pi(a)$. A set $S \subseteq \mathbb{A}$ is a *support* of $x \in X$ if $\mathsf{Fix}(S) \subseteq \mathsf{fix}(x)$, i.e. if every permutation that fixes all names in S fixes also x; in this case, we think of S as giving an upper bound on the names occurring in x. We say that x is *finitely supported* if x has some finite support. A $\mathsf{Perm}(\mathbb{A})$-set X is a *nominal set* if every $x \in X$ is finitely supported. One can then show that every $x \in X$ has a *least* finite support, denoted $\mathsf{supp}(x)$; we think of $\mathsf{supp}(x)$ as the set of names occurring in x. A name $a \in \mathbb{A}$ is *fresh* for $x \in X$ (in symbols, $a \,\#\, x$) if $a \notin \mathsf{supp}(x)$. Every set X can be made into a *discrete* nominal set by equipping X with the trivial $\mathsf{Perm}(\mathbb{A})$-action $\pi \cdot x = x$. Then, $\mathsf{supp}(x) = \emptyset$ for all $x \in X$. For every $n \in \mathbb{N}$, $\mathbb{A}^n$ is a nominal set under the componentwise action; then, $\mathsf{supp}(a_1, \dots, a_n) = \{a_1, \dots, a_n\}$.

A map $f\colon X \to Y$ between $\mathsf{Perm}(\mathbb{A})$-sets X, Y is *equivariant* if $f(\pi \cdot x) = \pi \cdot f(x)$ for all $x \in X$, $\pi \in \mathsf{Perm}(\mathbb{A})$. (This term extends to subsets $Y \subseteq X$ of a nominal set X by viewing them as predicates; explicitly, Y is equivariant if $y \in Y$ implies $\pi \cdot y \in Y$ for all $\pi \in \mathsf{Perm}(\mathbb{A})$.) Nominal sets and equivariant maps form a category **Nom**.

For every $\mathsf{Perm}(\mathbb{A})$-set X, the finitely supported elements of X form a nominal set. For instance, the powerset $\mathcal{P}(X)$ of a nominal set X is a $\mathsf{Perm}(\mathbb{A})$-set under the action $\pi \cdot A = \{\pi \cdot x \mid x \in A\}$. We obtain a nominal set $\mathcal{P}_{\mathsf{fs}}(X)$, the *finitely supported powerset*, consisting of the finitely supported elements of $\mathcal{P}(X)$. Similarly, for nominal sets X, Y, the set $X \to Y$ of maps from X to Y is a $\mathsf{Perm}(\mathbb{A})$-set under the action $(\pi \cdot f)(x) = \pi \cdot f(\pi^{-1} \cdot x))$. The set $X \to_{\mathsf{fs}} Y$ of finitely supported elements of $X \to Y$ forms a nominal set, the set of *finitely supported maps* $X \to Y$. A set $A \in \mathcal{P}_{\mathsf{fs}}(X)$ is *uniformly finitely supported* if $\mathsf{supp}(x) \subseteq \mathsf{supp}(A)$ for all $x \in A$; the *uniformly finitely supported powerset* $\mathcal{P}_{\mathsf{ufs}}(X)$ consists of the uniformly finitely supported subsets of X. If X is orbit-finite, then the uniformly finitely supported subsets of X are precisely the finite subsets. (Contrastingly, for instance, a subset of $\mathbb{A}$ is finitely supported iff it is either finite or cofinite.)

Given a nominal set X, the *abstraction set* $[\mathbb{A}]X$ is the quotient of $\mathbb{A} \times X$ modulo the equivalence relation $\sim$ defined by

$$(a, x) \sim (b, y) \iff (c\ a) \cdot x = (c\ b) \cdot y \text{ for some fresh } c \in \mathbb{A}.$$

We write $\langle a \rangle x$ for the equivalence class of (a, x) under $\sim$. We think of $\langle a \rangle x$ as arising from x by abstracting the name a in analogy to λ-abstraction in λ-calculus or name abstraction in the π-calculus, i.e. by allowing a to be α-renamed. Indeed we have $\mathsf{supp}(\langle a \rangle x) = \mathsf{supp}(x) \setminus \{a\}$.

Universal Coalgebra. Recall that a *functor* $F\colon \mathbf{C} \to \mathbf{C}$ on a category $\mathbf{C}$ assigns to each $\mathbf{C}$-object C a $\mathbf{C}$-object FC and additionally acts on $\mathbf{C}$-morphisms, compatibly with domains and codomains of morphisms as well as identities and composition. It is useful to think of FC as consisting of structured objects made up of inhabitants of C. We will primarily be interested in the case $\mathbf{C} = \mathbf{Nom}$. Indeed, all constructions of nominal sets recalled above extend to functors. For instance, the construction of the abstraction set $[\mathbb{A}]X$ extends to an *abstraction functor* $[\mathbb{A}](-)$ taking an equivariant map $f\colon X \to Y$ to the equivariant map $[\mathbb{A}]f\colon [\mathbb{A}]X \to [\mathbb{A}]Y$ given by $[\mathbb{A}]f(\langle a \rangle x) = \langle a \rangle f(x)$.

Universal coalgebra [36] provides a unified framework for automata models and concurrent systems by abstracting the type of a system as a functor $F\colon \mathbf{C} \to \mathbf{C}$ on a base category $\mathbf{C}$. An *F-coalgebra* (C, γ) then consists of a $\mathbf{C}$-object C, thought of as an object of *states*, and a $\mathbf{C}$-morphism $\gamma\colon C \to FC$ embodying the *transition* structure of the system. As a first example, we have a functor G on **Nom** defined on objects by $GX = 2 \times \mathcal{P}_{\mathsf{fs}}(\mathbb{A} \times X)$ where $2 = \{\top, \bot\}$ is a two-element discrete nominal set. A G-coalgebra $\gamma\colon C \to GC$ then assigns to each state $c \in C$ a binary value, read as determining finality, and a set of label/successor pairs; it is thus equivalently described as a triple $(C, F, \to)$ where $F \subseteq C$ is an equivariant set of final states and $\to \subseteq C \times \mathbb{A} \times C$ is an equivariant $\mathbb{A}$-labelled transition relation. Thus, orbit-finite G-coalgebras are precisely non-deterministic orbit-finite automata (NOFAs) over the alphabet $\mathbb{A}$ [4].

A $\mathbf{C}$-morphism $f\colon C \to D$ between F-coalgebras (C, γ) and (D, δ) is an *F-coalgebra morphism* if $Ff \cdot \gamma = \delta \cdot f$. Such F-coalgebra morphisms are thought of as behaviour-preserving maps. Correspondingly, universal coalgebra yields a canonical notion of behavioural equivalence: States $c \in C$, $d \in D$ in F-coalgebras (C, γ), (D, δ) are *behaviourally equivalent* if there exists an F-coalgebra (E, ϵ) and F-coalgebra morphisms $f\colon C \to E$, $g\colon D \to E$ such that $f(c) = g(d)$. This notion typically captures branching-time equivalences, such as bisimilarity on labelled transition systems. However, one is often interested in more coarse-grained notions of equivalence, such as trace equivalence on labelled transition systems or language equivalence of automata. There have been quite a number of approaches to modelling such equivalences in universal coalgebra (e.g. [21, 1, 30]); the specific framework of graded semantics is recalled in detail in Section 4.

3 Nominal Systems with Name Allocation

We proceed to recall regular non-deterministic nominal automata (RNNA) [37] and the associated subclass of nominal transition systems [32], discussing their global and local freshness semantics. The main driving principle behind RNNA is to let automata natively accept words with bound names indicating that a fresh name is read from the input string, so-called bar strings, and then generate data words from bar strings by taking all instantiations of the bound letters. Here, instantiation of bound names proceeds in two steps: First, form all α-equivalent variants of the bar string, and then forget that bound names are bound. Bound names are indicated by a preceding "I"; that is, $\mathsf{I}a$ indicates that a fresh letter a is read from the input. Thus, a *bar string* is just a word over the extended alphabet $\overline{\mathbb{A}} = \mathbb{A} \cup \{\mathsf{I}a \mid a \in \mathbb{A}\}$. Formally, an occurrence of $\mathsf{I}a$ in a bar string binds the name a, with the scope of the binding extending to the end of the string. Bound names in this sense can be renamed according to the usual discipline of α-equivalence known from λ-calculus or π-calculus; explicitly:

▶ **Definition 3.1** (α-Equivalence of bar strings)**.** The relation of *α-equivalence* on bar strings is the equivalence $\equiv_\alpha$ generated by

$$w\mathsf{I}av \equiv_\alpha w\mathsf{I}bu \quad \text{if } \langle a\rangle v = \langle b\rangle u \text{ in } [\mathbb{A}]\,\overline{\mathbb{A}}^*. \tag{3.1}$$

The α-equivalence class of a bar string $w \in \overline{\mathbb{A}}^*$ is denoted $[w]_\alpha$. The set $\mathsf{FN}(w)$ of *free names* in w is given recursively by $\mathsf{FN}(\epsilon) = \emptyset$ (where as usual ϵ denotes the empty string), $\mathsf{FN}(aw) = \{a\} \cup \mathsf{FN}(w)$, and $\mathsf{FN}(\mathsf{I}aw) = \mathsf{FN}(w) \setminus \{a\}$. A bar string w is *closed* if $\mathsf{FN}(w) = \emptyset$, and *clean* if all bound names $\mathsf{I}a$ in w are mutually distinct and distinct from all free names in w.

For instance, $\mathsf{I}a\mathsf{I}b \equiv_\alpha \mathsf{I}a\mathsf{I}c \equiv_\alpha \mathsf{I}a\mathsf{I}a$, while $\mathsf{I}a\mathsf{I}ba \equiv_\alpha \mathsf{I}a\mathsf{I}ca \not\equiv_\alpha \mathsf{I}a\mathsf{I}aa$. Crucially, the renaming of $\mathsf{I}c$ into $\mathsf{I}a$ in $\mathsf{I}a\mathsf{I}ca$ is *blocked* by the free occurrence of a in $\mathsf{I}ca$; formally, this is due to the (straightforward) observation that $\mathsf{supp}([w]_\alpha) = \mathsf{FN}(w)$.

We will be interested in three types of languages: *Data languages* are subsets of $\mathbb{A}^*$, whose elements we refer to as *data words*; *literal languages* are subsets of $\overline{\mathbb{A}}^*$; and *bar languages* are subsets of $\overline{\mathbb{A}}^*/\equiv_\alpha$. We convert bar languages into data languages in two alternative ways respectively corresponding to *global* and *local freshness* semantics of name binding. We define the auxiliary operation $\mathsf{ub}\colon \overline{\mathbb{A}}^* \to \mathbb{A}^*$ recursively by $\mathsf{ub}(\epsilon) = \epsilon$, $\mathsf{ub}(aw) = a\mathsf{ub}(w)$, and $\mathsf{ub}(\mathsf{I}aw) = a\mathsf{ub}(w)$; that is, ub removes all bars 'I' from a bar string. Then given a bar language L, we define data languages $N(L)$, $D(L)$ by

$$N(L) = \{\mathsf{ub}(w) \mid [w]_\alpha \in L, w \text{ clean}\} \quad \text{and} \quad D(L) = \{\mathsf{ub}(w) \mid [w]_\alpha \in L\}.$$

We briefly write $D(w) = D(\{w\})$, $N(w) = N(\{w\})$. For instance, $N(\mathsf{I}a\mathsf{I}b) = \{ab \mid a, b \in \mathbb{A}, a \neq b\}$, $D(\mathsf{I}a\mathsf{I}b) = \{ab \mid a, b \in \mathbb{A}\}$, and $D(\mathsf{I}a\mathsf{I}ba) = \{aba \mid a, b \in \mathbb{A}, a \neq b\}$. Roughly speaking, global freshness semantics, embodied by the N operator, enforces instantiations of bound letters to be distinct from all letters seen previously in the input word, while local freshness, embodied by the D operator, is more permissive, enforcing distinctness only from previously seen letters that are expected to be seen again (in an automaton model, this implies in particular that the previous letter must still be in memory). In fact the N operator is injective on languages of closed bar strings, so global freshness semantics is essentially equivalent to a semantics given directly in terms of bar languages.

The automata model of RNNA is designed to accept bar languages, subsequently equipped with global or local freshness semantics as indicated above.

▶ **Definition 3.2.** A *regular nondeterministic nominal automaton (RNNA)* is a tuple $A = (Q, \to, s, F)$ consisting of an orbit-finite set Q of *states*; an equivariant subset $\to \subseteq Q \times \overline{\mathbb{A}} \times Q$, the *transition relation*; an *initial state* $s \in Q$; and an equivariant subset $F \subseteq Q$ of *final states* such that the following conditions are satisfied:

- The relation $\to$ is *α-invariant*: If $s \xrightarrow{|a} q$ and $\langle a\rangle q = \langle a'\rangle q'$, then $s \xrightarrow{|a'} q'$.
- The relation $\to$ is *finitely branching up to α-equivalence*: For every state $q \in Q$, the sets $\{(a, q') \mid q \xrightarrow{a} q'\}$ and $\{\langle a\rangle q' \mid q \xrightarrow{|a} q'\}$ are finite.

The relation $\to$ is extended to bar strings in the usual manner; a bar string w is *accepted* by a state $q \in Q$ if there exists a state $q' \in F$ such that $q \xrightarrow{w} q'$. The *literal language* $L_0(A, q)$ accepted by q is the set of bar strings accepted by q. Correspondingly, the *bar language* $L(A, q)$ accepted by q is the set $L_0(A, q)/{\equiv_\alpha}$ of equivalence classes of bar strings accepted by q. The *data languages* accepted by q under *global* and *local freshness semantics* are the sets $N(L(A, q))$ and $D(L(A, q))$, respectively. When mention of q is omitted, we assume $q = s$; e.g. $L_0(A) = L_0(A, s)$ is the literal language accepted by A, etc. A *regular nominal transition system (RNTS)* is an RNNA A as above with $F = Q$, i.e. with all states final; we then refer to elements of $L_0(A, q)$, $L(A, q)$, $N(L(A, q))$, or $D(L(A, q))$ as *literal traces*, *bar traces*, or *data traces* (under global/local freshness) of q, respectively, and similarly to languages as *trace sets*. States q, q' in regular nominal transitions A, A', respectively, are *trace equivalent under global (local) freshness semantics* if $N(L(A, q)) = N(L(A', q'))$ (resp. $D(L(A, q)) = D(L(A', q'))$).

For readability, we will restrict the technical development to RNTS (see however Remark 4.4), which can be seen as a slight simplification of nominal transition systems in the sense of Parrow et al. [32] given by omitting propositional atoms (*state predicates*), restricting the number of bound names per transition to at most one, and imposing finite branching. RNTS can be viewed as orbit-finite coalgebras (cf. Section 2) for the functor $H\colon \mathbf{Nom} \to \mathbf{Nom}$ given on objects by

$$HX = \mathcal{P}_{\mathsf{ufs}}(\mathbb{A} \times X) \times \mathcal{P}_{\mathsf{ufs}}([\mathbb{A}]X). \tag{3.2}$$

The induced coalgebraic notion of behavioural equivalence can be described concretely as a form of bisimilarity in a fairly straightforward manner, and as such in particular does not coincide with trace equivalence under either local or global freshness semantics. Instead, we discuss in Sections 4 and 6 how these semantics are captured using graded semantics.

An important property of RNTS is that names are obtained only by reading them from the input word; formally, this is reflected in the properties that

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

The following diagram depicts an example RNTS A by picking representatives of orbits:

$$\begin{tikzcd}
 & s() \arrow[dr, "|a"] & \\
v(b) \arrow[ur, "b"] & & t(a) \arrow[dl, "|b"] \\
 & u(a,b) \arrow[ul, "a"] &
\end{tikzcd}$$

The bar trace set $L(A, s())$ of $s()$ consists of the α-equivalence classes of all bar strings of the form $(|a|bab)^n w$ where $w \in \{\epsilon, |a, |a|b, |a|ba\}$. Correspondingly, the data trace set $D(L(A, s()))$ under local freshness consists of the data traces of the form $a_1 b_1 a_1 b_1 \dots a_n b_n a_n b_n w$ where $a_1, \dots, a_n, b_1, \dots, b_n \in \mathbb{A}$, $a_i \neq b_i$ for $i = 1, \dots, n$, and $w \in \mathbb{A}^*$ is a data trace of length at most 3 in which the third letter, if any, equals the first and differs from the second.

4 Graded Monads and Graded Theories

We next briefly summarize the framework of *graded semantics* [30, 8], which serves to unify system semantics of varying granularity, such as found, for example, on the well-known linear-time / branching-time spectrum [42]. The central notion of the framework is the following.

▶ **Definition 4.1.** A *graded monad* [39] on a category $\mathbf{C}$ is a tuple $\mathbb{M} = ((M_n)_{n\in\mathbb{N}}, \eta, (\mu^{nk})_{n,k\in\mathbb{N}})$ consisting of

- a family of endofunctors $M_n \colon \mathbf{C} \to \mathbf{C}$, indexed over $n \in \mathbb{N}$;
- a natural transformation $\eta : \mathsf{Id} \to M_0$, the *unit*; and
- a family of natural transformations $\mu^{nk} : M_n M_k \to M_{n+k}$, indexed over $n, k \in \mathbb{N}$, the *multiplication*

satisfying graded variants of the usual *unit* and *associative* laws ($\mu^{0,n} \cdot \eta M_n = id_{M_n} = \mu^{n,0} \cdot M_n \eta$, $\mu^{n,k+m} \cdot M_n \mu^{k,m} = \mu^{n+k,m} \cdot \mu^{n,k} M_m$). The *graded Kleisli star* operation $(-)^*_k$ of $\mathbb{M}$, indexed over $k \geq 0$, lifts a morphism $f \colon X \to M_n Y$ to the morphism $f^*_k = \mu^{kn} \cdot M_k f \colon M_k X \to M_{k+n} Y$.

We recall how finitary graded monads on the category **Set** of sets and maps can be generated from algebraic presentations [39, 30]; one of our present contributions is to generalize this principle to graded monads on the category **Nom** of nominal sets (Section 5). A *graded signature* Σ is a collection of function symbols f with assigned *arity* $\mathsf{ar}(f)$ and *depth* $d(f)$. This induces a notion of term of uniform depth: Variables are terms of uniform depth 0, and if $d(f) = k$ and $\mathsf{ar}(f) = n$, then $f(t_1, \dots, t_n)$ is a term of uniform depth $m + k$ if all t_i are terms of uniform depth m. (Hence, if $\mathsf{ar}(c) = 0$ and $d(c) = k$, then c is a term of uniform depth m for every $m \geq k$.) A *graded theory* (Σ, E) then consists of a graded signature Σ and a set E of equational axioms of *uniform depth*, i.e. for every equation $s = t$ in E, there is some k such that s and t have uniform depth k. Such a graded theory induces a graded monad $((M_n), \eta, (\mu^{n,k}))$ on **Set** where $M_n X$ consists of terms over X of uniform depth n modulo derivable equality, η converts variables into terms, and $\mu^{n,k}$ collapses layered terms [30].

Generally, we think of $M_n X$ as a set of n-step behaviours ending in poststates in X, and of $M_n 1$ (where 1 is a terminal object) as a set of n-step behaviours. Correspondingly, one has the following notion of *graded semantics*:

▶ **Definition 4.2.** A *graded semantics* $(\mathbb{M}, \beta)$ for a functor G on a category $\mathbf{C}$ with a terminal object 1 consists of a graded monad $\mathbb{M} = ((M_n), \eta, (\mu^{nk}))$ and a natural transformation $\beta : G \to M_1$. Given a G-coalgebra $\gamma \colon X \to GX$, the *behaviour maps* $(\gamma^{(n)} \colon X \to M_n 1)_{n<\omega}$ are given recursively as the composites

$$\gamma^{(0)} = (X \xrightarrow{\eta_X} M_0 X \xrightarrow{M_0 !} M_0 1),$$
$$\gamma^{(n+1)} = (X \xrightarrow{\beta_X \cdot \gamma} M_1 X \xrightarrow{M_1 \gamma^{(n)}} M_1 M_n 1 \xrightarrow{\mu^{1n}_X} M_{n+1} 1).$$

States $x \in X$, $y \in Y$ in G-coalgebras $\gamma \colon X \to GX$ and $\delta \colon Y \to GY$ are *(β)-behaviourally equivalent* if $\gamma^{(n)}(x) = \delta^{(n)}(y)$ for all $n \in \mathbb{N}$.

▶ **Example 4.3** (Trace semantics of LTS [8]). We can view finitely branching $\mathcal{A}$-labelled transition systems as coalgebras for the functor $G\colon \mathbf{Set} \to \mathbf{Set}$ given by $GX = \mathcal{P}_\omega(\mathcal{A} \times X)$ where $\mathcal{P}_\omega$ denotes finite powerset. We have a graded monad $\mathbb{M}$ with functors M_n given on objects by $M_n X = \mathcal{P}_\omega(\mathcal{A}^n \times X)$ and with evident unit and multiplications, and obtain a graded semantics $(\mathbb{M}, \beta)$ for G where β is just the identity on $G = M_1$. This graded semantics captures exactly the trace semantics of LTS; in particular, two states are β-behaviourally equivalent iff they are trace equivalent in the usual sense. The graded monad $\mathbb{M}$ is induced by the graded algebraic theory that has a constant 0 and a binary function $+$ of depth 0, and for every $a \in \mathcal{A}$ a unary function $a(-)$ of depth 1; the equations are the usual join-semilattice axioms for 0 and $+$, and depth-1 equations $a(0) = 0$ and $a(x + y) = a(x) + a(y)$.

▶ Remark 4.4. The coalgebraic modelling of automata models differs from that of transition systems in having an additional output or acceptance component. Graded monads for the language semantics of automata retain a trace component capturing words that have a run; the language semantics is then obtained by applying a canonical projection $M_n 1 \to M_n 0$ to the graded semantics [27, 30]. In the algebraic view, accepted words are captured by a depth-1 constant for acceptance, and the canonical projection substitutes the single variable in the set 1 by the bottom element. In the present discussion, we elide this complication purely in the interest of readability.

One has the following graded analogues of monad algebras:

▶ **Definition 4.5** (Graded algebras). Given $n \in \mathbb{N}$ and a graded monad $\mathbb{M} = ((M_n), \eta, (\mu^{n,k}))$ on $\mathbf{C}$, an M_n*-algebra* A consists of

- a family $(A_k)_{0 \le k \le n}$ of $\mathbf{C}$-objects A_k called *carriers*, and
- a family $(a^{m,k})_{0 \le m+k \le n}$ of morphisms $a^{m,k} : M_m A_k \to A_{m+k}$

satisfying evident graded variants of the usual laws ($a^{0,m} \cdot \eta_{A_m} = id_{A_m}$, $a^{m,r+k} \cdot M_m a^{r,k} = a^{m+r,k} \cdot \mu^{m,r}_{A_k}$). A *homomorphism* $A \to B$ of M_n-algebras A, B is a family $(f_k)_{0 \le k \le n}$ of morphisms $f_k\colon A_k \to B_k$ such that, for $m + k \le n$, $b^{m,k} \cdot M_m f_k = f_{m+k} \cdot a^{m,k}$.

Most results on graded semantics, such as logical and game-based characterizations of behavioural equivalence [30, 8, 11, 12, 13], hinge on the monad being *depth*-1 in the following sense.

▶ **Definition 4.6.** A graded monad $\mathbb{M} = ((M_n), \eta, (\mu^{nk}))$ is *depth-1* if the following diagram (which commutes by the associative law of $\mathbb{M}$) is a coequalizer in the category of M_0-algebras for all X and $n \in \mathbb{N}$:

$$M_1 M_0 M_n X \underset{\mu^{1,0}_{M_n X}}{\overset{M_1 \mu^{0,n}_X}{\rightrightarrows}} M_1 M_n X \xrightarrow{\mu^{1,n}_X} M_{1+n} X. \tag{4.1}$$

On $\mathbf{Set}$, a graded monad is depth-1 iff it is generated by a *depth*-1 *graded theory*, i.e. one whose operations and axioms all have depth at most 1 [30, Proposition 7.3] (where the "only if" direction depends on size considerations, i.e. on either allowing infinitary operations or restricting to finitary graded monads). For instance, the presentation of the graded monad for trace semantics (Example 4.3) implies that the graded monad is depth-1. Similar results (in particular for the "if" direction) have been obtained for graded monads on posets [10], relational structures [9], and metric spaces [12], and indeed we prove the "if" direction for our notion of graded nominal theory, introduced next (Theorem 5.13). In the universal-algebraic view, the coequalizer condition in Definition 4.6 says roughly that all identifications caused by collapsing depth-1 terms over depth-n terms into depth-$(n + 1)$ terms come from shifting depth-0 operations between the depth-1 layer and the depth-n layer, in combination with the depth-0 axioms.

5 Graded Nominal Algebra

We now present our framework of *graded nominal algebra*, a universal-algebraic system in which theories define graded monads on **Nom**. The system builds on *nominal algebra* [18, 19], from which it differs most prominently by adding the grading. Further differences are discussed in Remark 5.11.

▶ **Definition 5.1.** A *nominal graded signature* Σ is a disjoint union of sets Σ_0, Σ_f, and Σ_b of *pure*, *free*, and *bound* operations, respectively. Each operation f is equipped with an *arity* $\mathsf{ar}(f) \in \mathbb{N}$ and a *depth* $d(f) \in \mathbb{N}$. We write $f/n \in \Sigma_0$ if $f \in \Sigma_0$ and $\mathsf{ar}(f) = n$, similarly for Σ_f, Σ_b, and Σ.

▶ **Definition 5.2** (Nominal Terms)**.** Given a nominal set X of *variables* and a graded signature Σ, we define *nominal Σ-terms $t, \dots$ (over X)* by the grammar

$$t ::= x \mid f(t_1, \dots, t_p) \mid a.g(t_1, \dots, t_p) \mid \nu a.\, h(t_1, \dots, t_p)$$

where $x \in X$, $a \in \mathbb{A}$, $f/p \in \Sigma_0$, $g/p \in \Sigma_\mathsf{f}$, and $h/p \in \Sigma_\mathsf{b}$. For brevity, we occasionally unify the three last clauses notationally as $\eta.f(t_1, \dots, t_p)$ for $f \in \Sigma$, where η is either nothing or of the form a or νa, with use of the notation implying the assumption of well-formedness.

Note in particular that we work with given nominal structures on sets of variables; cf. Remark 5.11. In terms $a.g(t_1, \dots, t_p)$, a is a *free* name, while in terms $\nu a.\, h(t_1, \dots, t_p)$, the name a is *bound*. This reading gives rise to an evident notion of *free names* in a term, where we count both explicit occurrences of names and occurrences of names in the support of variables; we write $\mathsf{FN}(t)$ for the set of free names in t. Explicitly, $\mathsf{FN}(x) = \mathsf{supp}(x)$, $\mathsf{FN}(f(t_1, \dots, t_p)) = \mathsf{FN}(t_1) \cup \dots \cup \mathsf{FN}(t_p)$, $\mathsf{FN}(a.g(t_1, \dots, t_p)) = \mathsf{FN}(t_1) \cup \dots \cup \mathsf{FN}(t_p) \cup \{a\}$, and $\mathsf{FN}(\nu a.\, h(t_1, \dots, t_p)) = (\mathsf{FN}(t_1) \cup \dots \cup \mathsf{FN}(t_p)) \setminus \{a\}$.

We have a notion of *uniform depth* of terms defined completely analogously to the set-based case (Section 4). Then, a substitution σ (a map assigning nominal terms $\sigma(x)$ to variables $x \in X$) has *uniform depth* if all $\sigma(x)$ are of the same uniform depth. As usual, we write $t\sigma$ for the application of σ to a nominal term t, which just replaces all occurrences of variables x in t with $\sigma(x)$. Like in nominal algebra [17, 18, 19], application of substitutions thus explicitly does *not* avoid capture of names; these issues are instead dealt with at the level of the derivation system. We write $\mathsf{Term}_{\Sigma,m}(X)$ for the set of nominal terms of uniform depth m over X; equipped with the evident action of $\mathsf{Perm}(\mathbb{A})$ (in particular, $\pi \cdot \nu a.\, f(t_1, \dots, t_n) = \nu\pi(a).\, f(\pi \cdot t_1, \dots, \pi \cdot t_n)$), $\mathsf{Term}_{\Sigma,m}(X)$ is a nominal set.

▶ **Definition 5.3** (Equations and Theories)**.** A *depth-n Σ-equation*, written $X \vdash_n t = u$, consists of an orbit-finite nominal set X of *variables* and terms $t, u \in \mathsf{Term}_{\Sigma,n}(X)$. We often omit mention of n. A *graded (nominal) theory* $T = (\Sigma, E)$ consists of a graded signature Σ and a set E of Σ-equations, referred to as *axioms*.

Given a graded theory $T = (\Sigma, E)$, we define *derivable Σ-equations* inductively by the derivation rules in Figure 1. Note that the (ax) rule allows instantiation of axioms both under a permutation τ of the names and under a substitution σ of the variables. We refer to the condition imposed on σ in the premise of the rule as *derivable equivariance*. This condition ensures *equivariance of derivability* (i.e., if $X \vdash_m t = u$ is derivable, then so is $X \vdash_m \pi \cdot t = \pi \cdot u$ for every $\pi \in \mathsf{Perm}(\mathbb{A})$).

▶ Remark 5.4. Rule (ax) as phrased in Figure 1 has infinitely many premisses. However, it can be equivalently replaced with a finitary version where σ is defined only on the finitely many variables occurring in r and s (and on one representative of every orbit of Y, in

$$\text{(refl)}\ \frac{}{X \vdash_0 x = x} \qquad \text{(symm)}\ \frac{X \vdash_m u = t}{X \vdash_m t = u} \qquad \text{(trans)}\ \frac{X \vdash_m t = v \qquad X \vdash_m v = u}{X \vdash_m t = u}$$

$$\text{(cong)}\ \frac{X \vdash_m t_i = u_i \qquad (i = 1, \dots, p)}{X \vdash_{m+d(f)} \eta.f(t_1, \dots, t_p) = \eta.f(u_1, \dots, u_p)}\ (f/p \in \Sigma)$$

$$\text{(ax)}\ \frac{X \vdash_l \pi \cdot \sigma(y) = \sigma(\pi \cdot y) \qquad \forall \pi \in \mathsf{Perm}(\mathbb{A}), y \in Y}{X \vdash_{m+l} (\tau \cdot r)\sigma = (\tau \cdot s)\sigma}\ (Y \vdash_m r = s \in E, \tau \in \mathsf{Perm}(\mathbb{A}))$$

$$\text{(perm)}\ \frac{X \vdash_m t_i = (a\ b) \cdot u_i \qquad (i = 1, \dots, p)}{X \vdash_{m+d(f)} \nu a.f(t_1, \dots, t_p) = \nu b.f(u_1, \dots, u_p)}\ (f/p \in \Sigma_{\mathsf{b}}, a \neq b \in \mathbb{A}, a \# u_i)$$

Figure 1 Equational inference rules over a graded theory $T = (\Sigma, E)$.

the maybe pathological case that Y has orbits not mentioned in r, s) and then extended to all of Y ensuring derivable equivariance. This is made possible by imposing premisses ensuring well-definedness up-to-derivability of such an extension. The set of these premisses is orbit-finite, so that in a simultaneous inductive proof with derivable equivariance, we may restrict to finitely many representative premisses.

Like in standard universal algebra, the notion of graded nominal theory comes equipped with a notion of model for which we will establish soundness and completeness:

▶ **Definition 5.5** (Graded nominal algebras)**.** Given a graded signature Σ and a depth $n < \omega$, a *nominal* (Σ, n)*-algebra* A consists of a family $(A_i)_{0 \le i \le n}$ of nominal sets, called the *carriers*, and for each $f/p \in \Sigma$ and $m \le n - d(f)$, an equivariant function $f_{A,m}$ of type $A_m^p \to A_{m+d(f)}$ if $f \in \Sigma_0$, of type $\mathbb{A} \times A_m^p \to A_{m+d(f)}$ if $f \in \Sigma_{\mathsf{f}}$, and of type $[\mathbb{A}]A_m^p \to A_{m+d(f)}$ if $f \in \Sigma_{\mathsf{b}}$. A *homomorphism* $A \to B$ of (Σ, n)-algebras A, B is a family $(h_i)_{0 \le i \le n}$ of equivariant functions $h_i \colon A_i \to B_i$ satisfying the following equalities for $f/p \in \Sigma$, $a \in \mathbb{A}$, and $x = (x_1, \dots, x_p) \in A_m^p$:

$$\begin{aligned} h_{m+d(f)}(f_{A,m}(x)) &= f_{B,m}(h_m(x_1), \dots, h_m(x_p)) && \text{if } f \in \Sigma_0 \\ h_{m+d(f)}(f_{A,m}(a, x)) &= f_{B,m}(a, h_m(x_1), \dots, h_m(x_p)) && \text{if } f \in \Sigma_{\mathsf{f}} \\ h_{m+d(f)}(f_{A,m}(\langle a \rangle x)) &= f_{B,m}(\langle a \rangle (h_m(x_1), \dots, h_m(x_p))) && \text{if } f \in \Sigma_{\mathsf{b}}. \end{aligned}$$

We write $\mathsf{Alg}(\Sigma, n)$ for the category of nominal (Σ, n)-algebras and their homomorphisms. We also define (Σ, ω)-algebras and their homomorphisms; definitions are the same except that the range of indices is $i, m < \omega$ (rather than $\le n$)

Algebras actually satisfying the axioms of the theory are singled out as follows:

▶ **Definition 5.6** (Term evaluation, satisfaction, models)**.** Given a nominal (Σ, n)-algebra A (where $n \le \omega$) and a base depth $k \in \mathbb{N}_0, k \le n$, the *value* $[\![t]\!]_m^\iota \in A_{k+m}$ of a depth-m term t under an equivariant *valuation* $\iota \colon X \to A_k$ is defined by

$$\begin{aligned} [\![x]\!]_0^\iota = \iota(x) \qquad [\![f(t_1, \dots, t_p)]\!]_{m+d(f)}^\iota &= f_{A,k+m}([\![t_1]\!]_m^\iota, \dots, [\![t_p]\!]_m^\iota) \\ [\![a.f(t_1, \dots, t_p)]\!]_{m+d(f)}^\iota &= f_{A,k+m}(a, [\![t_1]\!]_m^\iota, \dots, [\![t_p]\!]_m^\iota) \\ [\![\nu a.f(t_1, \dots, t_p)]\!]_{m+d(f)}^\iota &= f_{A,k+m}(\langle a \rangle([\![t_1]\!]_m^\iota, \dots, [\![t_p]\!]_m^\iota)) \end{aligned}$$

We say that A *satisfies* a Σ-equation $X \vdash_m t = u$ if $[\![t]\!]^\iota_m = [\![u]\!]^\iota_m$ for every valuation $\iota\colon X \to A_k$ with $k + m \leq n$ (a vacuous condition if $m > n$). Finally, given a graded theory $T = (\Sigma, E)$, a (T, n)-*model* is a nominal (Σ, n)-algebra satisfying every axiom in E. We write $\mathsf{Alg}(T, n)$ for the full subcategory of (T, n)-models in $\mathsf{Alg}(\Sigma, n)$.

We obtain soundness w.r.t. this notion of model:

▶ **Theorem 5.7** (Soundness)**.** *The derivation system in Figure 1 is sound: If an equation is derivable in a graded theory T, then it is satisfied by every (T, n)-model.*

We prove completeness of the system by constructing free graded nominal algebras, which at the same time yields our construction of a graded monad on **Nom** from a graded nominal theory. We write $\sim$ for the equivalence relation on terms defined by derivable equality, which by equivariance of derivability is equivariant, and denote the equivalence class of a depth-n term t by $[t]_n$. Given a nominal set X and $n \leq \omega$, we define a (Σ, n)-algebra FX by carriers $(FX)_i = \mathsf{Term}_{\Sigma,i}(X)/{\sim}$ (for $i \leq n$ if $n < \omega$, or for $i < \omega$ if $n = \omega$), and for $f/p \in \Sigma$:

$$
\begin{aligned}
f_{FX,m}([t_1]_m, \dots, [t_p]_m) &= [f(t_1, \dots, t_p)]_{m+d(f)} && \text{if } f \in \Sigma_0,\\
f_{FX,m}(a, [t_1]_m, \dots, [t_p]_m) &= [a.f(t_1, \dots, t_p)]_{m+d(f)} && \text{if } f \in \Sigma_{\mathsf{f}},\\
f_{FX,m}(\langle a\rangle([t_1]_m, \dots, [t_p]_m)) &= [\nu a.f(t_1, \dots, t_p)]_{m+d(f)} && \text{if } f \in \Sigma_{\mathsf{b}}.
\end{aligned}
$$

This algebra is equipped with a *canonical valuation* $\eta\colon X \to (FX)_0$ given by $\eta(x) = [x]_0$. The rules of the system, specifically (cong) and (perm), guarantee that this structure is well-defined.

▶ **Theorem 5.8** (Free (T, n)-models)**.** *For a graded nominal theory T and $n \leq \omega$, the algebra FX is a free (T, n)-model over X w.r.t. the forgetful functor that sends a (T, n)-model A to the nominal set A_0, with η being the universal arrow.*

Since FX identifies terms precisely when they are derivably equal, we obtain

▶ **Corollary 5.9** (Completeness)**.** *The derivation system in Figure 1 is complete: If a depth-m equation is satisfied by every (T, n)-model of a graded nominal theory T where $m \leq n \leq \omega$, then it is derivable in T.*

From the adjunction of Theorem 5.8 for the case $n = \omega$, we obtain a graded monad by standard results [16, Section 3]. Specifically, we have a (strict) *action* $\star$ of the discrete monoidal category $(\mathbb{N}, +, 0)$ on $\mathsf{Alg}(T, \omega)$ given on objects by taking $n \star A$, for a (T, ω)-model A, to be the (T, ω)-model whose i-th carrier is A_{i+n} and whose interpretation $f_{n\star A,i}$ of $f \in \Sigma$ is $f_{A,i+n}$. We then obtain a graded monad $\mathbb{M}^T$ by sandwiching this action between the two adjoint functors as per Theorem 5.8. In particular, the functor parts M^T_n of $\mathbb{M}^T$ are given as $M^T_n X = (n \star FX)_0 = (FX)_n$; that is, we have $M^T_n X = \mathsf{Term}_{\Sigma,n}(X)/\sim$ as expected. The unit of $\mathbb{M}$ is just the canonical valuation η, and the multiplication is obtained from the counit of the adjunction; intuitively, η thus converts variables into terms, and the multiplication collapses layered terms.

▶ **Example 5.10** (Traces in nominal systems)**.** As a first example of a graded nominal theory, take Σ_0 to consist of depth-0 operations $0, +$, with 0 a constant and $+$ being binary and written in infix notation, and Σ_{f} to consist of a single unary operation pre of depth 1. We impose depth-0 equations ensuring that $0, +$ form a *nominal join-semilattice* (i.e. a nominal set carrying an equivariant join-semilattice structure), explicitly $X \vdash_0 x + 0 = x$, $X \vdash_0 x + x = x$, $X \vdash_0 x + y = y + x$, and $X \vdash_0 (x + y) + z = x + (y + z)$ for all orbit-finite X

(more precisely, representatives of all such X modulo isomorphism to ensure that the axioms form a set, a point we will elide henceforth) and $x, y, z \in X$. Additionally, we have depth-1 axioms

$$X \vdash_1 a.\mathsf{pre}(x + y) = a.\mathsf{pre}(x) + a.\mathsf{pre}(y) \quad \text{and} \quad \emptyset \vdash_1 a.\mathsf{pre}(0) = 0 \tag{5.1}$$

for all $a \in \mathbb{A}$, all orbit-finite X, and all $x, y \in X$. Notice here that postulating the equalities for *all* X, x, y ensures that they apply to all elements of all nominal sets (see also Remark 5.11); on the other hand, postulating the equalities for all rather than just some a is in fact inessential, since rule (ax) allows applying permutations to axioms. Similarly as in Example 4.3, the induced graded monad $\mathbb{M}$ has functor parts M_n given on objects by $M_n X = \mathcal{P}_\omega(\mathbb{A}^n \times X)$. Coalgebras for the functor $G = M_1 = \mathcal{P}_\omega(\mathbb{A} \times (-))$ are finitely branching *non-deterministic nominal transition systems*, that is, NOFAs (Section 2) in which all states are accepting. The graded semantics $(\mathbb{M}, id)$ for G captures precisely the trace semantics of such systems (i.e. the language semantics of the associated NOFA). We see an example with name-binding operations in Section 6.

▶ Remark 5.11. The necessity of combining operations with free or bound names instead of having names and name binding as separate syntactic constructs is owed to the grading. For instance, in Example 5.10, it is crucial that $a.\mathsf{pre}(-)$ has depth 1 as a whole, rather than being composed of two components, one of which would necessarily have depth 0.

In some previous systems of nominal algebra [18, 19], contexts of equations are presented in terms of variables and freshness conditions, which in our terms essentially amounts to restricting to contexts X that are *strong* nominal sets [40], i.e. for every $\pi \in \mathsf{Perm}(\mathbb{A})$ and $x \in X$, one has $\pi \cdot x = x$ if and only if $\pi(a) = a$ for all $a \in \mathsf{supp}(x)$. Roughly speaking, strong nominal sets thus do not have symmetries; more formally, every orbit in a strong nominal set is isomorphic to some nominal set of the form $\mathbb{A}^{\#n}$ for $n \in \mathbb{N}$, i.e. the set of n-tuples over $\mathbb{A}$ with pairwise distinct entries [33, Lemma 2.4.2]. For instance, the set $\{\{a, b\} \subseteq \mathbb{A} \mid a \neq b\}$ fails to be strong, due to the symmetry embodied in the equality $(a\ b) \cdot \{a, b\} = \{a, b\}$. Our system is thus more expressive in that it allows equations to be conditioned on symmetries in the context; e.g. we could impose an equation $f(x) = x$ conditioned on $(a\ b) \cdot x = x$. We do not currently claim to have real-life examples where this is needed; for instance, in Example 5.10 one could equivalently restrict all equations to strong nominal sets of variables.

Finally, we establish the announced criterion for the graded monad induced by a graded nominal theory to be depth-1 (Definition 4.6).

▶ **Definition 5.12.** A graded theory T is *depth-1* if all its operations and axioms are of depth at most 1.

▶ **Theorem 5.13.** *Graded monads induced by depth-*1 *graded nominal theories are depth-*1*.*

For instance, the graded monad of Example 5.10 is depth-1.

6 Case Study: Graded Monads for Global and Local Freshness

We now present our main examples, in which we cast the global and local freshness semantics of regular nominal transition systems (RNTS; Section 3) as graded semantics. We begin with global freshness, exploiting that the induced notion of state equivalence coincides with bar language semantics. The graded theory T^{bar} for bar language semantics extends the theory for traces of non-deterministic nominal transition systems (Example 5.10) by an additional depth-1 operation $\mathsf{abs} \in \Sigma_{\mathsf{b}}$, defining a signature Σ^{bar}, and additional depth-1 equations

$$X \vdash_1 \nu a.\,\mathsf{abs}(x+y) = (\nu a.\mathsf{abs}(x)) + (\nu a.\mathsf{abs}(y)) \quad \text{and} \quad \emptyset \vdash_1 \nu a.\,\mathsf{abs}(0) = 0 \tag{6.1}$$

for all orbit-finite X and $x, y \in X$. We write $at = a.\mathsf{pre}(t)$ and $|at = \nu a.\,\mathsf{abs}(t)$ for all terms t. A *pretrace* over a nominal set X is an element of $(w, x) \in \overline{\mathbb{A}}^* \times X$, with x thought of as a poststate reached after executing w. We can literally extend the notion of *α-equivalence* $\equiv_\alpha$ of bar strings to pretraces. We now define a graded monad $\mathbb{M}^{\mathsf{bar}} = ((M_n^{\mathsf{bar}}), \eta, (\mu^{nm}))$ of pretraces modulo α-equivalence by

$$M_n^{\mathsf{bar}} X = \mathcal{P}_\omega((\overline{\mathbb{A}}^n \times X)/\equiv_\alpha),$$

(hence $M_0^{\mathsf{bar}} X = \mathcal{P}_\omega X$) with $\eta(x) = [x]_\alpha$ and $\mu^{nm}(S) = \{[uv]_\alpha \mid [uV]_\alpha \in S, [v]_\alpha \in V\}$.

▶ **Theorem 6.1.** *The graded monad $\mathbb{M}^{\mathsf{bar}}$ is induced by the graded nominal theory T^{bar}, and in particular is depth-1 by Theorem 5.13.*

Recall that regular nominal transition systems are coalgebras for the functor H as per (3.2). To avoid discussion of infinitary theories, we restrict H to $H_\omega = \mathcal{P}_\omega(\mathbb{A}\times(-))\times\mathcal{P}_\omega([\mathbb{A}](-))$ (and hence restrict to finitely branching RNTS). We have an obvious isomorphism $\beta\colon H_\omega \to M_1$, giving rise to a graded semantics $(\mathbb{M}^{\mathsf{bar}}, \beta)$ for H. This graded semantics captures global freshness semantics:

▶ **Theorem 6.2.** *Two states in finitely branching RNTS are trace equivalent under global freshness semantics iff they are behaviourally equivalent w.r.t. the graded semantics $(\mathbb{M}^{\mathsf{bar}}, \beta)$.*

The *local freshness* semantics of an RNTS (Section 3) is computed from its bar language semantics, and imposes additional identifications. We define a graded theory T^{loc} for local freshness semantics by extending T^{bar} with depth-1 axioms

$$X \vdash_1 a.\mathsf{pre}(x) + \nu a.\mathsf{abs}(x) = \nu a.\mathsf{abs}(x) \tag{6.2}$$

for all orbit-finite X and $x \in X$, saying that every bound name can be instantiated by itself. We will see that this is already sufficient to describe local freshness. However, we can simplify some computations by introducing *name restrictions*: Similar to the computationally useful idea of closing RNNA under name dropping and thus closing the literal language under α-equivalence [37], we will introduce an algebraic operation that can arbitrarily restrict the support of a term. This construction will be essential for the description of equivalence games for local freshness semantics in Section 7. We define the graded theory T^{drop} by extending T^{loc} with a bound operation $\mathsf{res}/1$ at depth 0, which hides a name in its argument from the outside. Correspondingly, we add the *name restriction axioms* [34, Chapter 9]:

$$X \vdash_0 \nu a.\mathsf{res}(x) = x \quad \text{for } a \# x \quad \text{and} \quad X \vdash_0 \nu a.\mathsf{res}(\nu b.\mathsf{res}(x)) = \nu b.\mathsf{res}(\nu a.\mathsf{res}(x)). \tag{6.3}$$

We require this operation to be compatible with our join-semilattice structure, in that

$$X \vdash_0 \nu a.\mathsf{res}(x+y) = \nu a.\mathsf{res}(x) + \nu a.\mathsf{res}(y) \quad \text{and} \quad X \vdash_0 x = x + \nu a.\mathsf{res}(x), \tag{6.4}$$

with $X \vdash_0 \nu a.\mathsf{res}(0) = 0$ already implied by (6.3). For the depth-1 operations, we need the axioms

$$X \vdash_1 \nu a.\mathsf{res}(b.\mathsf{pre}(x)) = b.\mathsf{pre}(\nu a.\mathsf{res}(x)) \quad \text{and} \quad X \vdash_1 \nu a.\mathsf{res}(a.\mathsf{pre}(x)) = 0, \tag{6.5}$$

$$X \vdash_1 \nu a.\mathsf{res}(\nu b.\mathsf{abs}(x)) = \nu b.\mathsf{abs}(\nu a.\mathsf{res}(x)) \tag{6.6}$$

for $a \neq b$, where $X \vdash_1 \nu a.\mathsf{res}(\nu a.\mathsf{abs}(x)) = \nu a.\mathsf{abs}(x)$ is again implied by (6.3).

Let $\mathbb{M}^{\mathsf{loc}} = ((M^{\mathsf{loc}}), \eta, (\mu^{nm}))$ denote the graded monad induced by T^{loc} (we refrain from decorating η and μ with more indices) and similarly for $\mathbb{M}^{\mathsf{drop}}$ and T^{drop}. Again, we have by Theorem 5.13 that $\mathbb{M}^{\mathsf{loc}}$ and $\mathbb{M}^{\mathsf{drop}}$ are depth-1. From theory inclusion, we then have canonical natural transformations

$$\theta_n \colon M_n^{\mathsf{bar}} \to M_n^{\mathsf{loc}} \quad \text{and} \quad \kappa_n \colon M_n^{\mathsf{loc}} \to M_n^{\mathsf{drop}}$$

for every n, and obtain graded semantics $(\mathbb{M}^{\mathsf{loc}}, \theta_1 \cdot \beta)$ and $(\mathbb{M}^{\mathsf{drop}}, \kappa_1 \cdot \theta_1 \cdot \beta)$ for H. Both of these capture local freshness:

▶ **Theorem 6.3.** *Two states in finitely branching RNTS are trace equivalent under local freshness semantics iff they are behaviourally equivalent w.r.t. the graded semantics* $(\mathbb{M}^{\mathsf{loc}}, \theta_1 \cdot \beta)$ *iff they are behaviourally equivalent w.r.t. the graded semantics* $(\mathbb{M}^{\mathsf{drop}}, \kappa_1 \cdot \theta_1 \cdot \beta)$.

▶ Remark 6.4. Given a nominal set X, $x \in X$, and $N \subseteq \mathsf{supp}(x)$, write $x|_N = \mathsf{Fix}(N) \cdot x = \{\pi \cdot x \mid \pi \in \mathsf{Fix}(N)\}$ for the orbit of x under $\mathsf{Fix}(N)$ (similar to the states in the construction of name-dropping RNNA [37]). We can then describe elements of $M_0^{\mathsf{drop}} X$ as finite sets S of such orbits such that $x|_{N'} \in S$ whenever $x|_N \in S$ and $N' \subseteq N$.

7 Graded Behavioural Equivalence Games

For illustration of the benefits of having a depth-1 graded semantics, we discuss how graded behavioural equivalence games, previously considered for set-based coalgebras [11], transfer to the nominal setting. We then instantiate the generic game to trace semantics of RNTS (Section 6). The game has a categorical formulation that is analogous to the set-based case. The main novelty in the present work is the formulation of the game in nominal-algebraic terms using our system of graded nominal algebra from Section 5, presented next. To this end, we fix a depth-1 graded semantics $(\mathbb{M}, \beta)$ for a functor F on **Nom**, with $\mathbb{M}$ presented by a graded theory T, and an F-coalgebra (C, γ). Throughout, we exploit that in join semilattices, inequalities $s \leq t$ are expressible as equations $t = s + t$.

Predeterminization. Graded behavioural equivalence games are morally played on a *predetermization* of (C, γ), that is, on a coalgebra $M_0 C \to \overline{M}_1 M_0 C$ for a functor $\overline{M}_1$ on M_0-algebras induced from a depth-1 graded semantics. (The name of the construction is owed to the fact that in case $M_0 1 = 1$, it yields an actual determinization [11]). For our present purposes, we only need the construction of $\overline{M}_1$. Given an M_0-algebra A, the M_0-algebra $\overline{M}_1(A, a)$ may be described categorically via a coequalizer construction. Algebraically, $\overline{M}_1 A$ is obtained by freely extending A to an M_1-algebra and then forgetting A. That is, $\overline{M}_1 A$ consists of depth-1 terms over A, quotiented by equalities derivable in the given graded theory from depth-0 equalities that hold over A. It can be shown that $\overline{M}_1(M_n X, \mu^{0n}) \cong (M_{n+1} X, \mu^{0,n+1})$ [11]; we thus briefly write $\overline{M}_1 M_n = M_{n+1}$.

▶ **Example 7.1.**

1. *Global freshness semantics (graded monad* $\mathbb{M}^{\mathsf{bar}}$*):* We have $\overline{M}_1 A = A_\omega^{\mathbb{A}} \times [\mathbb{A}]A$, where the endofunctor $(-)_\omega^{\mathbb{A}}$ on M_0-algebras sends A to the set of functions $f \colon \mathbb{A} \to A$ with $f(a) = 0$ for almost all a.
2. *Local freshness semantics with name-dropping (graded monad* $\mathbb{M}^{\mathsf{drop}}$*):* An M_0-algebra A is a join semilattice $(A, +, 0)$ with a name restriction operation $[\mathbb{A}]A \to A, \langle a \rangle x \mapsto a \setminus x$. As such, we can describe $\overline{M}_1 A$ as the set of finitely supported functions $f \colon \mathbb{A} \to_{\mathsf{fs}} A$ such that $(a\ c) \cdot (a \setminus f(c)) \leq f(a)$ for all $a \in \mathbb{A}$ and $c \,\#\, (f, a)$.

In both cases, the given description implies immediately that $\overline{M}_1$ preserves monomorphisms, as needed for the correctness of the game (Theorem 7.3).

Equivalence games. Graded behavioural equivalence games come in a finite-depth and an infinite-depth variant [11]; we focus on finite depth. Generally, we blur the distinction between terms and elements of M_nC, i.e. equivalence classes of terms. The game is played by *Spoiler (S)* and *Duplicator (D)*, with D trying to prove and S trying to disprove behavioural equivalence. Configurations of the game are pairs $(s,t) \in M_0C \times M_0C$, read as equalities $C \vdash_0 s = t$ claimed by D. Given a set $Z \subseteq M_0C \times M_0C$ of depth-0 equalities, we write $Z \Rightarrow C \vdash_n s = t$ for terms $s,t \in \mathsf{Term}_{\Sigma,n}(C)$ if the depth-n equation $C \vdash_n s = t$ can be derived from the closure of Z under equivariance in the system of Section 5 (that is, equations in Z may be renamed but not substituted into). We view the map $\beta \cdot \gamma \colon C \to M_1C$ as a uniform depth-1 substitution σ. The game is parametrized by a number n of rounds:

▶ **Definition 7.2.** The *n-round* $(\mathbb{M},\beta)$*-behavioural equivalence game* $\mathsf{G}_n(\gamma)$ on (C,γ) is played in n rounds. A round starting in configuration (s,t) is played as follows:

1. Duplicator plays a finite set $Z \subseteq M_0C \times M_0C$ such that $Z \Rightarrow C \vdash_1 s\sigma = t\sigma$.
2. Spoiler picks an element $(s',t') \in Z$, which becomes the next configuration.

Any player who cannot move on their turn, loses. After n rounds have been played, ending in configuration (s,t), D wins if $1 \vdash_0 s\sigma_0 = t\sigma_0$ is derivable, where σ_0 is the depth-0 substitution mapping all variables to $\star \in 1$; we refer to this step as *calling the bluff*. Otherwise, S wins.

The game is intuitively understood as follows. We may regard $\beta \cdot \gamma \colon C \to M_1C$ as defining the behaviour of states $c \in C$ by algebraic equations $c = \sigma(c)$. The behaviour map $\gamma^{(n)} \colon C \to M_n1$ as per Definition 4.2 unfolds definitions of states n times and finally applies the substitution σ_0 mapping all variables to $\star$ as above; this defines a depth-n substitution σ_n such that $\gamma^{(n)}(c)$ is the equivalence class of $\sigma_n(c)$ in M_n1. We can thus understand the n-round game from configuration (s,t) as D trying to construct a nominal-algebraic proof of the depth-n equation $1 \vdash_n s\sigma_n = t\sigma_n$, with S challenging selected claims made by D.

Correctness of the game is formulated as follows.

▶ **Theorem 7.3** (Game correctness). *Let $(\mathbb{M},\beta)$ be a graded semantics for a functor F on* **Nom***, with $\mathbb{M}$ presented by a graded nominal theory, and let $\gamma \colon C \to FC$ be a F-coalgebra. Suppose that $\overline{M}_1$ preserves monomorphisms (i.e. injective M_0-homomorphisms). For $n \geq 0$, D wins the configuration (s,t) in $\mathsf{G}_n(\gamma)$ iff $(\gamma^{(n)})^*_0(s) = (\gamma^{(n)})^*_0(t)$. In particular, states $c,d \in C$ are β-behaviourally equivalent iff D wins (c,d) in $\mathsf{G}_n(\gamma)$ for every $n \geq 0$.*

▶ **Example 7.4.** For our main examples (Section 6), we give explicit descriptions of the game, which we can rearrange to let S move first. In both cases, we use normal forms of M_1C implicit in the equation $M_1 = \overline{M}_1M_0C$ and the respective description of $\overline{M}_1$ (Example 7.1), and the substitution σ representing the coalgebra structure.

1. *Global freshness semantics (graded monad $\mathbb{M}^{\mathsf{bar}}$):* Given a term $s \in M_0C$, i.e. a finite set of states, we can describe $s\sigma \in M_1C = \overline{M}_1M_0C = (M_0C)^{\mathbb{A}}_{\omega} \times [\mathbb{A}]M_0C$ as $s\sigma = ((s_a)_{a\in\mathbb{A}}, \langle c\rangle s_{\mathsf{l}c})$ where

   $$s_a = \{y \mid x \in s, x \xrightarrow{a} y\} \quad\text{and}\quad s_{\mathsf{l}c} = \{y \mid x \in s, x \xrightarrow{\mathsf{l}c} y\}) \quad \text{for } c \,\#\, s.$$

   In a configuration (s,t) with $c \,\#\, (s,t)$, S either picks an $a \in \mathbb{A}$ and $x \in s_a$ or an $x \in s_{\mathsf{l}c}$ (or analogously for t). Duplicator then responds with a subset $t' \subseteq t_a$ or $t' \subseteq t_{\mathsf{l}c}$ correspondingly, yielding the new configuration $(t' \cup \{x\}, t')$ corresponding to the claim $x \leq t'$ (and analogously for s). This also means that S cannot switch sides after the initial move.

2. *Local freshness semantics with name-dropping (graded monad $\mathbb{M}^{\mathsf{drop}}$):* Viewing terms $s \in M_0C$ as sets of name restrictions closed under further restriction as detailed in Remark 6.4, we can describe $s\sigma \in M_1C = \overline{M}_1M_0C$ as a function $\mathbb{A} \to M_0C, a \mapsto s_a$ where

$$s_a = \{y|_{N'} \mid x|_N \in s, x \xrightarrow{a} y, a \in N, N' \subseteq \mathsf{supp}(y) \cap N\}$$
$$\cup \; \{(a\ c) \cdot (y|_{N' \setminus \{a\}}) \mid x|_N \in s, x \xrightarrow{\mathsf{I}c} y, N' \subseteq \mathsf{supp}(y) \cap (N \cup \{c\})\}.$$

In a configuration (s, t), S then picks an $a \in \mathbb{A}$ and $x|_N \in s_a$ (or $y|_N \in t_a$), to which D responds with a subset $t' \subseteq t_a$ closed under further restriction, yielding the new configuration $(t' \cup \{x|_{N'} \mid N' \subseteq N\}, t')$ (or symmetrically), effectively corresponding to the claim $x|_N \leq t'$. Again, this means that S cannot switch sides after the initial move. As a rough summary, S picks a side where they play restricted successors, and D answers with restriction-closed sets of restricted successors under matching labels.

8 Conclusions and Future Work

We have introduced a universal-algebraic system for the presentation of graded monads on the category of nominal sets. In a subsequent case study, we have applied this system to model the global and local freshness semantics of regular non-deterministic transition systems (a slightly simplified form of *regular non-deterministic nominal automata (RNNA)* [37]) in the framework of *graded semantics* [30, 8]. Notably, the axiom distinguishing local from global freshness semantics just states that every bound name subsumes the corresponding free name. We have proved soundness and completeness of our system of *graded nominal algebra*, and we have shown that the graded monad induced by graded nominal theory is *depth*-1 if the graded theory itself is depth-1. In particular, this shows that the graded monads featuring in the local and global freshness semantics of regular non-deterministic nominal automata are depth-1.

In work on graded semantics over sets [30, 8, 11], partial orders [10], metric spaces [12], and relational structures [13], the somewhat technical-looking depth-1 property has played a key role in enabling characterization results for graded semantics in terms of modal logics and Spoiler-Duplicator games. We have illustrated this point by introducing a nominal variant of *graded equivalence games* [11], whose correctness depends on the depth-1 property, and instantiating the generic game to obtain game characterizations of trace equivalence in nominal transition systems under both local and global freshness semantics. Our system of graded nominal algebra plays a key role in graded equivalence games, which like in the set-based case can be viewed as playing out a graded algebraic proof. An important direction for future work is to extend further results in graded semantics to the nominal setting, notably results on characteristic modal logics [30, 8]. We expect that such results will, like the game characterization that we have already materialized in the present work, depend on the involved graded monads being depth-1. An additional point of future research is coverage of additional examples, such as fresh labelled transition systems [3].

References

1 Jirí Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius, and Alexandra Silva. A coalgebraic perspective on minimization and determinization. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures, FoSSaCS 2012*, volume 7213 of *LNCS*, pages 58–73. Springer, 2012. doi:10.1007/978-3-642-28729-9_4.

2 Jiří Adámek, Horst Herrlich, and George E. Strecker. *Abstract and concrete categories: The joy of cats*. John Wiley & Sons Inc., 1990. Republished in: Reprints in Theory and Applications of Categories, No. 17 (2006) pp. 1–507. URL: `http://tac.mta.ca/tac/reprints/articles/17/tr17abs.html`.

3 Mohamed H. Bandukara and Nikos Tzevelekos. A logic for fresh labelled transition systems. *CoRR*, abs/2506.14538, 2025. `doi:10.48550/arXiv.2506.14538`.

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

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

6 Filippo Bonchi, Daniela Petrisan, Damien Pous, and Jurriaan Rot. Coinduction up-to in a fibrational setting. In Thomas A. Henzinger and Dale Miller, editors, *Computer Science Logic and Logic in Computer Science, CSL-LICS 2014*, pages 20:1–20:9. ACM, 2014. `doi:10.1145/2603088.2603149`.

7 Ranald Alexander Clouston and Andrew M. Pitts. Nominal equational logic. In Luca Cardelli, Marcelo Fiore, and Glynn Winskel, editors, *Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin*, volume 172 of *ENTCS*, pages 223–257. Elsevier, 2007. `doi:10.1016/J.ENTCS.2007.02.009`.

8 Ulrich Dorsch, Stefan Milius, and Lutz Schröder. Graded monads and graded logics for the linear time - branching time spectrum. In *Concurrency Theory, CONCUR 2019*, volume 140 of *LIPIcs*, pages 36:1–36:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.CONCUR.2019.36`.

9 Chase Ford. *Presentations of Graded Coalgebraic Semantics*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2023. URL: `https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/23568`.

10 Chase Ford, Stefan Milius, and Lutz Schröder. Behavioural preorders via graded monads. In *Logic in Computer Science, LICS 2021*, pages 1–13. IEEE, 2021. `doi:10.1109/LICS52264.2021.9470517`.

11 Chase Ford, Stefan Milius, Lutz Schröder, Harsh Beohar, and Barbara König. Graded monads and behavioural equivalence games. In Christel Baier and Dana Fisman, editors, *Logic in Computer Science, LICS 2027*, pages 61:1–61:13. ACM, 2022. Full version available on arXiv under `https://arxiv.org/abs/2203.15467`. `doi:10.1145/3531130.3533374`.

12 Jonas Forster, Lutz Schröder, Paul Wild, Harsh Beohar, Sebastian Gurke, Barbara König, and Karla Messing. Quantitative graded semantics and spectra of behavioural metrics. In Jörg Endrullis and Sylvain Schmitz, editors, *Computer Science Logic, CSL 2025*, volume 326 of *LIPIcs*, pages 33:1–33:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/LIPIcs.CSL.2025.33`.

13 Jonas Forster, Lutz Schröder, and Paul Wild. Conformance games for graded semantics. In *Logic in Computer Science, LICS 2025*, pages 555–567. IEEE, 2025. `doi:10.1109/LICS65433.2025.00048`.

14 Florian Frank, Daniel Hausmann, Stefan Milius, Lutz Schröder, and Henning Urbat. Alternating nominal automata with name allocation. In *Logic in Computer Science, LICS 2025*, pages 57–70. IEEE, 2025. `doi:10.1109/LICS65433.2025.00012`.

15 Florian Frank, Stefan Milius, and Henning Urbat. Coalgebraic semantics for nominal automata. In Helle Hvid Hansen and Fabio Zanasi, editors, *Coalgebraic Methods in Computer Science, CMCS 2022*, volume 13225 of *LNCS*, pages 45–66. Springer, 2022. `doi:10.1007/978-3-031-10736-8_3`.

16 Soichiro Fujii, Shin-ya Katsumata, and Paul-André Melliès. Towards a formal theory of graded monads. In Bart Jacobs and Christof Löding, editors, *Foundations of Software Science and Computation Structures, FOSSACS*, volume 9634 of *LNCS*, pages 513–530. Springer, 2016. `doi:10.1007/978-3-662-49630-5_30`.

17 Murdoch Gabbay and Aad Mathijssen. A formal calculus for informal equality with binding. In Daniel Leivant and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information and Computation, WoLLIC 2007*, volume 4576 of *LNCS*, pages 162–176. Springer, 2007. doi:10.1007/978-3-540-73445-1_12.

18 Murdoch James Gabbay. Nominal algebra and the HSP theorem. *J. Log. Comput.*, 19(2):341–367, 2009. doi:10.1093/LOGCOM/EXN055.

19 Murdoch James Gabbay and Aad Mathijssen. Nominal (universal) algebra: Equational logic with names and binding. *J. Log. Comput.*, 19(6):1455–1508, 2009. doi:10.1093/LOGCOM/EXP033.

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

21 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Log. Methods Comput. Sci.*, 3(4), 2007. doi:10.2168/LMCS-3(4:11)2007.

22 Matthew Hennessy. A fully abstract denotational semantics for the pi-calculus. *Theor. Comput. Sci.*, 278:53–89, 2002. doi:10.1016/S0304-3975(00)00331-5.

23 Falk Howar, Bengt Jonsson, and Frits Vaandrager. Combining black-box and white-box techniques for learning register automata. In *Computing and Software Science – State of the Art and Perspectives*, volume 10000 of *LNCS*, pages 563–588. Springer, 2019. doi:10.1007/978-3-319-91908-9_26.

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

25 Michael Kaminski and Daniel Zeitlin. Finite-memory automata with non-deterministic reassignment. *Int. J. Found. Comput. Sci.*, 21(5):741–760, 2010. doi:10.1142/S0129054110007532.

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

27 Alexander Kurz, Stefan Milius, Dirk Pattinson, and Lutz Schröder. Simplified coalgebraic trace equivalence. In Rocco De Nicola and Rolf Hennicker, editors, *Software, Services, and Systems*, volume 8950 of *LNCS*, pages 75–90. Springer, 2015. doi:10.1007/978-3-319-15545-6_8.

28 Alexander Kurz and Daniela Petrisan. On universal algebra over nominal sets. *Math. Struct. Comput. Sci.*, 20(2):285–318, 2010. doi:10.1017/S0960129509990399.

29 Alexander Kurz, Daniela Petrisan, and Jirí Velebil. Algebraic theories over nominal sets. *CoRR*, abs/1006.3027, 2010. doi:10.48550/arXiv.1006.3027.

30 Stefan Milius, Dirk Pattinson, and Lutz Schröder. Generic trace semantics and graded monads. In *Algebra and Coalgebra in Computer Science, CALCO 2015*, volume 35 of *LIPIcs*, pages 253–269. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CALCO.2015.253.

31 Frank Neven, Thomas Schwentick, and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Log.*, 5(3):403–435, 2004. doi:10.1145/1013560.1013562.

32 Joachim Parrow, Johannes Borgström, Lars-Henrik Eriksson, Ramunas Forsberg Gutkovas, and Tjark Weber. Modal logics for nominal transition systems. *Log. Methods Comput. Sci.*, 17(1):6:1–6:49, 2021. URL: https://lmcs.episciences.org/7137.

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

34 Andrew Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press, 2013. doi:10.1017/CBO9781139084673.

35 Simon Prucker and Lutz Schröder. Nominal tree automata with name allocation. In Rupak Majumdar and Alexandra Silva, editors, *Concurrency Theory, CONCUR 2024*, volume 311 of *LIPIcs*, pages 35:1–35:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CONCUR.2024.35.

36 Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.
37 Lutz Schröder, Dexter Kozen, Stefan Milius, and Thorsten Wißmann. Nominal automata with name binding. In *Foundations of Software Science and Computation Structures, FOSSACS 2017*, volume 10203 of *LNCS*, pages 124–142, 2017. doi:10.1007/978-3-662-54458-7_8.
38 Hannes Schulze, Lutz Schröder, and Üsame Cengiz. Graded monads in the semantics of nominal automata, 2026. doi:10.48550/arXiv.2510.24353.
39 A. Smirnov. Graded monads and rings of polynomials. *J. Math. Sci.*, 151:3032–3051, 2008. doi:10.1007/s10958-008-9013-7.
40 Nikos Tzevelekos. *Nominal Game Semantics.* PhD thesis, University of Oxford, 2008.
41 Henning Urbat, Daniel Hausmann, Stefan Milius, and Lutz Schröder. Nominal büchi automata with name allocation. In Serge Haddad and Daniele Varacca, editors, *Concurrency Theory, CONCUR 2021*, volume 203 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CONCUR.2021.4.
42 Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In Jos C. M. Baeten and Jan Willem Klop, editors, *Theories of Concurrency, CONCUR 1990*, volume 458 of *LNCS*, pages 278–297. Springer, 1990. doi:10.1007/BFB0039066.